DTU Physics
Department of Physics

Machine learning methods for geometry optimization of atomic structures

PhD Thesis

Estefanía Garijo del Río, April 2021

Machine learning methods for geometry optimization of atomic structures

PhD Thesis
April, 2021

By
Estefanía Garijo del Río

Published by: DTU, Department of Physics, Fysikvej, Building 309, 2800 Kgs. Lyngby Denmark
www.fysik.dtu.dk

Abstract

In this thesis, a series of novel methods for the identification of the optimal geometry of atomic structures are introduced. The geometry of a material or molecule, that is, the arrangement of its atoms in space, determines all its properties. Thus, the determination of the geometry of an atomic system is the first step in any computational study and the development of computationally efficient methods is important in order to accelerate research in materials science.

Even though methods for the identification of the structures with minimum energy (both locally and globally) and transition states are abundant in literature, these methods often rely on quantum chemistry methods, such as density functional theory (DFT), which can be computationally very expensive. The methods presented in this thesis address this issue by using machine learning methods to model the potential energy surface. The machine learning model can then be used to guide the search of the optimal geometry and, thus, reduce the computational time.

An important subfield of geometry optimization of atomic structures is the identification of a local minimum of the potential energy surface, that is, a structure with no internal forces on the atoms. For this problem, we introduce two new minimization methods, which consistently achieve a reduction of the number of DFT calculations. For one of these methods, we show that this reduction can be of up to a factor two for adsorption systems. We further show that the reuse of the trajectories from former local optimizations and transition state search methods can further speed up the calculations.

For many applications, however, it is not enough to use a local optimization method, since the approximate structure of the system studied is not known. In this thesis we present two different methods to tackle this problem. We have proposed a method that uses a message-passing neural network to determine the optimal prototype for a material. This can be used in the context of computational screening. Furthermore, we have extended the Gaussian process regression formalism used in local optimization so that it can be used for the global optimization problem. In this way, we have created a novel global optimization method that can identify the global minimum in a fraction of the DFT evaluations needed by other methods, and used it to identify the optimal structure of Ta_6O_{15} clusters and the oxidized structure of ZrN.

Resumé

I denne afhandling introduceres en række af nye metoder til identificering af den optimale geometri for atomare strukturer. En materiales eller et molekyles geometri, dvs. konfigurationen af atomerne i rummet, bestemmer alle dets egenskaber. Det første skridt i ethvert numerisk studie er således at finde den geometriske struktur for det atomare system og udviklingen af beregningsmæssigt effektive metoder er vigtigt for at accelerere forskning i materialevidenskab.

Selvom der i litteraturen findes et væld af metoder til at finde strukturer med minimal energi (både lokalt og globalt), så afhænger disse ofte af kvantekemiske metoder, såsom Density Functional Theory (DFT), som kan være beregningsmæssigt meget dyre. Metoderne som præsenteres i denne afhandling omgår denne forhindring ved at bruge maskinlæringsmetoder til at modellere den potentielle energi. Modellen kan derefter bruges til, at vejlede søgningen efter den optimale geometri og derved reducere beregningstiden.

Et vigtigt underfelt i optimisering af atomare strukturer er idenficeringen af et lokalt minimum i den potentielle energi, det vil sige, en struktur uden interne kræfter på nogle af atomerne. Til dette problem introducerer vi to nye minimeringsmetoder, som konsekvent opnår en reduktion i antallet af DFT-udregninger. For den ene af disse metoder viser vi, at denne reduktion kan være op til en faktor 2 for adsorptionssystemer. Vi finder derudover at genbrug af data fra tidligere lokale optimeringer og metoder til at finde overgangstilstande kan reducere beregningstiden yderligere.

Til mange anvendelser er det dog ikke nok at bruge lokale optimeringsmetoder, da ingen god tilnærmelse til den fysiske struktur er kendt. I denne afhandling præsenterer vi to forskellige metoder til at håndtere dette problem. Vi har brugt et *message-passing* neuralt netværk til at bestemme den optimale strukturprototype for et givent 3D system. Dette kan bruges i screening studier af materialer. Derudover har vi udvidet Gaussisk process formalismen brugt i lokalt optimering, således at den kan bruges til global optimering. På denne måde har vi lavet en ny global optimeringsmetode som kan identificere det globale minimum med en brøkdel af de DFT evalueringer som andre metoder har brug for, og vi har brugt denne metode til at identificere den optimale struktur for Ta_6O_{15} klynger og den oxiderede variant af ZrN.

Preface

This thesis is submitted in candidacy for the Ph.D. degree in physics from the Technical University of Denmark. The work was carried out at the section for Computational Atomic-scale Materials Design (CAMD) at the Department of Physics, in the period from August 2017 to April 2021, and supervised by professors Karsten W. Jacobsen and Kristian S. Thygesen. This Ph.D. project was funded by the VILLUM Fonden research grant (9455), through the VILLUM Center for Science of Sustainable Fuels and Chemicals.

Kogens Lyngby, 21st April 2021

Estefanía Garijo del Río

Caminante, son tus huellas
el camino y nada más;
caminante, no hay camino,
se hace camino al andar.
Al andar se hace camino,
y al volver la vista atrás
se ve la senda que nunca
se ha de volver a pisar.
Caminante no hay camino
sino estelas en la mar.

Antonio Machado, Campos de Castilla (1912).

Translation:

Wanderer, your footsteps are
the road and nothing else;
wanderer, there is no road,
the road is made as one walks.
As one walks the road is made,
and upon taking a glance back
it is seen the path that never
shall be trod again.
Wanderer, there is no road
but wakes on the sea.

Antonio Machado, Fields of Castile (1912).

Acknowledgements

The completion of this thesis has involved the support of many people. There is a lot of people I would like to thank for their help in such a large project, but the details would fill another equally long thesis. Thus, here is a summary of the main highlights.

First and foremost, I would like to thank Karsten, my supervisor, for his support and advice through this three plus years long journey. The path was not straightforward and things very often did not go as we expected, but Karsten showed plenty of energy and enthusiasm in every turn. I would also like to thank him on the personal side for his understanding and support during all my struggles.

I am grateful for the help and patience of Jens Jørgen and Ole. I have had asked for many weird stuff during the Ph.D. and you always had some advice or fix. I have also lost the count of how many times I have deleted all my files or screwed up with my GPAW version, and you have been there every time to help me. This thesis would not have been nearly as good without your patience.

Marianne and Bettina also deserve a special mention. Thank you with your help and patience with the paperwork, both DTU related, but also advice with the Danish system. You were always there to help me with a smile, thank you.

I am grateful to Hanne and Joan for the emotional guidance during this process. My project and my life would often mismatch and they came up with a route to handle them both.

I have been lucky enough to collaborate with some amazing Ph.D. students and post-docs, who have generously shared their time and knowledge with me. Special thanks to Peter, José, Sami and Andreas, I have learnt a lot from you all. I am also grateful to professor Thomas Bligaard for many interesting discussions.

It would be impossible not to mention the members of the the CAMD team (in no specific order): Per, Peter, Nicky, Mohnish, Korina, Sten, Daniele, Morten, Thorsten, Mikkel, Ask, Douglas, Anders, Thorbjørn, Asbjørn, Luca, Hadeel, Simone, Fabian, Mark, Stefano, Mads, Sami, Sajid, Alireza, Nikolaj, Matthew, Urko, Jiban, Ask, Cuauhtémoc, Joachim and Casper[1]. Thank you for the discussions about all possible topics: physics, spaghetti, the physics of spaghetti, Italian coffee, the number of flamingos in Australia, white tigers, tick pits, supercomputers made of very small birds, …, you name it. Thank you for the coffee breaks, the cookie breaks, pizza breaks and Friday bars (and for turning my office into a bar, I guess …). Also thank you for proofreading my thesis collectively.

Also, thank you to the exceptionally welcoming team at UCLouvain (in Belgium) during my external stay, specially to Vishank, Sergi, Aurélie, George and Martha; and to other DTU students/friends: Kåre, Diana and Mads; for good discussions over academic and not-so-academic topics.

I would like to thank to all my friends (to all of them, even if they are not mentioned here specifically) for the massive support I have received during these years. Without

[1] I hope I did not miss anyone!

their help I would not have been able to complete this thesis. Marina Peñalver, Gonzalo Fernández de Santaella, Sandra Malpica, Pau Hernández, Carlos Salort, Korina Kuhar, Mariana Sousa and Matías Fenoglio deserve a special mention. Thank you for being there for me during the most difficult times. I am also thankful to Asbjørn, since he has been the main source of support during the last months of this thesis.

Thank you to my mother for encouraging me to pursue my PhD and thank you for the long hours she spent listening to me. Thank you to my grandmother for always believing in me, even if she was not very sure about what a PhD was. I wished both of you were here today to see this work completed. Thank you to the Pineapple-Squad (also known as the *del Río* family), for the incredible amount of help and support during the most difficult days of this process, specially to my uncle Javi and my cousin Andrea. I believe we make an amazing team.

And last, but not least, thank you to my sister Celia. One could not wish for a more generous and supportive sister than you. Thank you for accompanying me through this journey.

Contents

List of Symbols

$\mathcal{E}$	Total energy of an atomic system
N	Number of atoms in an atomic structure
N_e	Number of electrons in an atomic system
E	Potential energy/potential energy surface
ϵ	Total energy of the electronic structure problem
Ψ	Total wave function of an atomic system
Θ	Wave function of the nuclei
Φ	Many body electronic wave function
$\mathbf{R}_i$	Position of the $i-$th nuclei/atom
$\mathbf{R}$	Set of positions of all atoms
$\mathbf{r}_i$	Position of the $i-$th electron
$\mathbf{r}$	Set of positions of all electrons
$\mathbf{f}_i$	Force on the $i-$th atom
$\mathbf{f}$	List of the forces on all the atoms
n_e	electronic density
ε_i	Kohn-Sham energies
φ_i	Kohn-Sham orbitals
n	Number of items in the training set
$\mathbf{x}$	Input of a machine learning method/ Cartesian coordinate [2]
$\mathbf{x}_i$	$i-th$ input to a machine learning method
y_i	Targets to a machine learning method
$f(\mathbf{x})$	underlying function to a machine learning method
$\mathbf{X}$	Design matrix (in Cartesian coordinates)
$\mathcal{N}$	Gaussian probability distribution function
$\mathbf{h}_k$	k-th layer of hidden units of a neural network
$\mathbf{W}_k$	Weights of the k-th layer of a neural network
$\mathbf{b}_k$	Biases of the k-th layer of a neural network
$g(\cdot)$	Activation function of a neural network
$k(\cdot,\cdot)$	Kernel function
$m(\cdot)$	Prior function of a kernel method
C	Gram matrix of a kernel method
μ	Mean of the posterior probability in a Gaussian process
σ^2	Variance of the posterior probability in a Gaussian process
ρ	Modified Oganov fingerprint
$\mathbf{p}$	Direction of search in local optimization
H_k	Hessian matrix at configuration $\mathbf{x}_k$
B_k	Quasi-Newton approximation of the Hessian matrix at the k-th iteration
α_k	Step size of the k-th iteration of a local optimizer

ϱ	Descriptor to an atomic structure
$\boldsymbol{\rho}$	Design matrix in the descriptor space of ϱ
$\nabla_{\mathbf{x}}$	Nabla operator with respect to Cartesian coordinates
σ_n^E	Regularization for the energy in GPR
$\sigma_n^{\mathbf{f}}$	Regularization for the forces in GPR
Σ_n	Regularization matrix for a GPR with energies and forces
$K(\cdot,\cdot)$	Kernel with gradient information (it is a matrix)
$\mathbb{C}$	Gram matrix of a GPR with gradient information
$\mathbb{V}$	Variance for of a GPR with gradient information
ℓ	Scale of a stationary kernel
k_0	Prefactor of a kernel

List of Abbreviations

PES	Potential energy surface
DFT	Density functional theory
EMT	Effective medium theory
XC	Exchange-correlation
PBE	Perdew–Burke-Ernzerhof functional
BFGS	Broyden, Fletcher, Goldfarb and Shano approximation
LCB	Lower confidence bound acquisition function
MEP	Minimum energy path
NEB	Nudged elastic band
GP	Gaussian process
GPR	Gaussian process regression
ML	Machine Learning
NN	Neural network
MAE	Mean absolute error
RMSE	Root mean square error

[2] All algorithms in this thesis could be ultimately considered to depend on the Cartesian coordinates as an input, since they are used in the building of descriptors. It also results in simpler notation

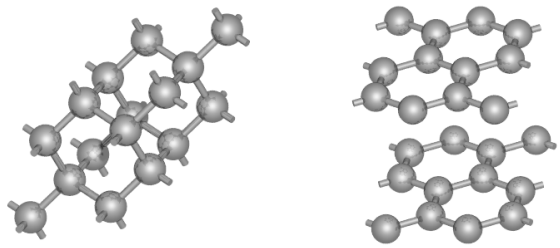
1 Introduction

Matter is made up of atoms. Every object we see, every material and every molecule is built with the same building blocks, the 108 atomic species of the periodic table. And yet, the knowledge of the building blocks of something, i.e., its chemical compositions, does not necessarily predict its properties. In order to characterize a material or molecule, both the chemical composition and the arrangement of its atoms in space (i.e., the geometry of its atomic structure) are needed [1].

Take diamond as an example. Diamond is a crystal made of carbon atoms: each carbon atom is bonded covalently to four other carbon atoms forming a tetrahedron. This kind of bonds are very stable, making diamond one of the strongest materials known. In addition, since all electrons in diamond are "invested" on forming the covalent bonds, they are not available for conduction. As a result, diamond is an insulator to current and transparent (i.e. it has a large band gap).

Contrary to the popular commercial, diamonds do not last for ever. Diamonds are formed under exceptional pressures and temperatures at great depths under the Earth's crust, but once they emerge to the surface, they slowly start turning into graphite in a process that takes billions of years [2]. Graphite is the stable form of carbon at room temperature, and its properties could not be more different from those of diamond. It is one of the softest materials in nature, it is exfoliated very easily, making it a perfect component for pencils. It has an intense gray metallic colour and it is, indeed, conducting.

The difference between the properties of graphite and diamond originates from the difference between the geometry of the atomic structure. Instead of the tetrahedrally bonded crystal, graphite is made up of layers of carbon atoms whose bonds form hexagons. The carbon atoms are bonded to other atoms in the same layer by covalent bonds and the layers are kept together by much weaker van der Waals forces. This



(a) Diamond (b) Graphite

Figure 1.1: Atomic structures of diamond (the bonds of each atom point towards the corner of a tetrahedron) and graphite (the bonds form layers of hexagons).

explains why it makes such good pencils: the van der Waals bonds are easy to break, transferring the atoms in the outermost layers form the pencil to the paper [2].

Diamond and graphite are not the only examples for which the atomic structure determines the properties of the substance. Carbon has other allotropes with very surprising properties [3–5]. Since they all have the exact same composition, it is their structure that determines their properties.

The behaviour of matter at the atomic scale, including the geometry of atomic systems, is determined by the laws of quantum mechanics. Over the last decades, new computational methods have been developed that allow to simulate matter at quantum scale both efficiently and accurately. These methods have enabled researchers expand their understanding of matter. The recent increase in computational power has led to the revolution of high throughput computational materials and drugs design: it is now possible to simulate thousands of new materials in order to find those whose properties are relevant for technological applications [6, 7], opening new pathways for materials research.

The first step in such computational studies is, as explained above, to determine the geometry of the atomic structure, since all the other properties will follow from that. This can be achieved by moving the atoms around until the energy of the atomic configuration is minimized. The energy and the forces of an atomic system can be readily computed using quantum chemistry methods such as DFT [8, 9], and be used to guide the search. Due to the complexity of the quantum description of matter, however, each quantum chemistry calculation may be computationally expensive. In this way, the number of energy and force calculations required to determine the geometry of an atomic system often becomes a computational bottleneck that limits the scope of high-throughput studies.

In this thesis, we show how reusing the data from previous calculations along the way can help to guide the search for optimal atomic structures. We achieve this by using machine learning methods, that is, statistical models that use the energy and force of all the known structures to predict the energy and forces of those yet unknown. These models are then used as surrogate potentials, in the sense that they can be used instead of the original quantum mechanical potential for structure search. We show it is possible to build models on the fly, as the geometry optimization progresses, that increase their accuracy as more model configurations are added. In this way, we show how the identification of optimal geometries can be achieved with fewer energy calculations as compared with traditional methods, reducing the computational costs of the computational simulation of materials and molecules.

Figure 1.2 sketches how the algorithms presented in this thesis work. The quantum mechanical potential is depicted with a solid back line and shows a minimum at about $z \sim 1.5$ Å. This is the target of the method and the current atomic configuration is marked by a blue dot. The information of all the configurations explored (marked with black dots) is used to build a machine learning model (blue line) that matches the energies and forces of all the known configurations. The resulting machine learning model will have a minimum, which may have a predicted an energy that is lower than any of

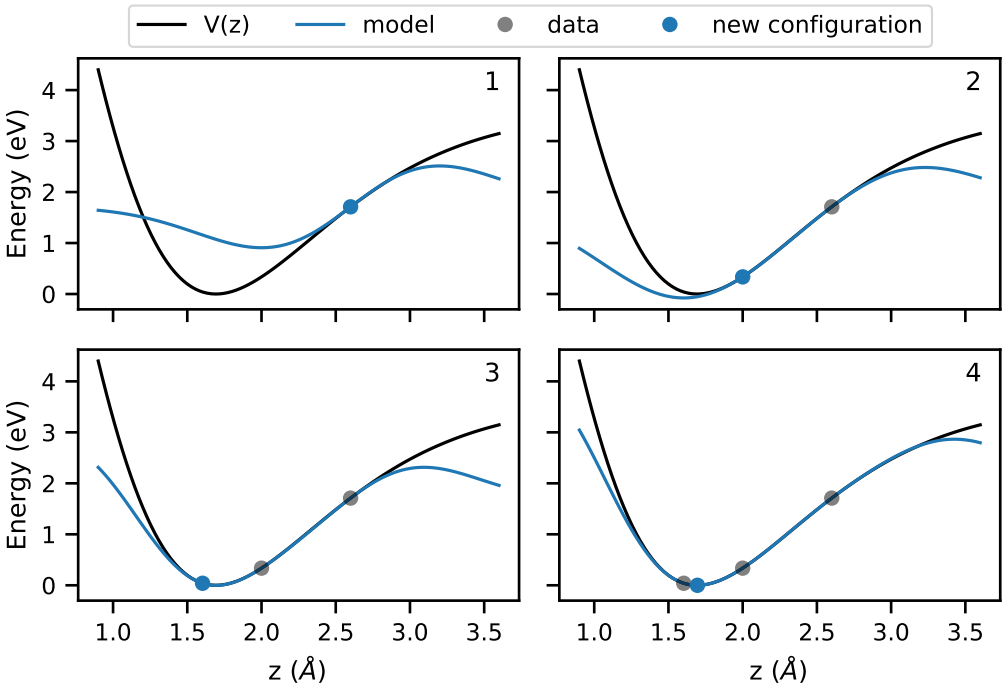


Figure 1.2: Example of the use of a data-driven surrogate model to identify a minimum configuration. The optimization begins with a single configuration, which is used to build a model (blue) of the potential $V(z)$ (black) that has the right energy and force, as illustrated in panel 1. The minimization of the model potential provides a new configuration that is lower in energy (panel 2), and can be used to improve the accuracy of the model. The process can be then repeated (panels 3 and 4) until the minimum of the potential $V(z)$ is found. The system studied is a gold atom constrained in 1 dimension, to vary its distance z to a fcc (100) gold surface. The potential $V(z)$ of the system is described with effective medium theory [10].

the energies in the training set. The methods in this thesis then proceed to find the minimum of the machine learning model and to compute its energy and force. It might be that the new configuration is not a good estimate of the minimum, but the information about its energy and force will certainly improve the machine learning model. The new machine learning model will have a new minimum, that can be sampled again and whose energies and forces will in turn further improve the model. This procedure eventually converges to the optimal configuration, as the machine learning model converges towards the underlying quantum mechanical potential.

1.1 Outline

This thesis is organized as follows. In Chapter 2, the physics of the atomic structure problem is described, from the perspective of quantum mechanics. The concept of

potential energy surface, which will be the target of all the models presented, is introduced, and its numerical computation is discussed. In Chapter 3, the machine learning techniques that have been used to create computationally efficient models of the potential energy surface are introduced. After a general discussion, the two machine learning methods that are used in the papers contained in this thesis, neural networks and Gaussian process regression, are described. This chapter also contains a discussion about the use of descriptors of atomic structures as inputs to machine learning algorithms.

Chapter 4 discusses the numerical techniques that are most commonly used in the optimization of potential energy surfaces. After a description of the most common methods for finding local minima, global optimization and transition state search techniques are briefly reviewed. The technical details and choices of the Gaussian process regression models that have been used in this thesis are discussed in Chapter 5. The benefits of including force information in the description of the PES are examined, before discussing the different choices of kernels and priors. The chapter finishes with a discussion on the optimization of the hyperparameters and the computational cost.

The results of the research presented in this thesis are given in Chapter 6, in the form of a summary of the papers. An interested reader may find the papers in an appendix at the end of this thesis. Chapter 7 contains the conclusions.

2 The description of matter at atomic level

In order to study the geometry of the atomic structure of a material or molecule and the properties associated with it, it is necessary to describe the system with quantum mechanics. In this chapter, the basic theory governing the dynamics of atoms and electrons is reviewed. The concept of potential energy surface (PES), which is central to this thesis, and its relationship with the geometry of the atomic structures are discussed. The chapter finishes with a review of density functional theory (DFT) as a way to compute potential energy surfaces.

2.1 The structure of matter

At a fundamental level, matter is made up of atoms and can be described at a quantum level by the interaction of its constituents: the nuclei and the electrons. These are charged bodies, and interact with each other via the electrostatic repulsion of electron pairs:

$$\hat{V}_{ee}(\mathbf{r}) = \sum_{\substack{i=1 \\ j>1}}^{N_e} \frac{1}{\|\mathbf{r}_i - \mathbf{r}_j\|}, \tag{2.1}$$

electrostatic repulsion of atomic nuclei pairs:

$$\hat{V}_{aa}(\mathbf{R}) = \sum_{\substack{i=1 \\ j>1}}^{N} \frac{Z_i Z_j}{\|\mathbf{R}_i - \mathbf{R}_j\|} \tag{2.2}$$

and the electrostatic attraction between electrons and nuclei:

$$\hat{V}_{ea}(\mathbf{R}, \mathbf{r}) = -\sum_{i,j=1}^{N,N_e} \frac{Z_i}{\|\mathbf{R}_i - \mathbf{r_j}\|}. \tag{2.3}$$

Here, N is the number of atoms and N_e the number of electrons, Z_i and $\mathbf{R}_i$ the atomic number and the position of the $i-$th atomic nuclei,respectively, and $\mathbf{r}_i$ the position of the $i-$th electron. The symbols **R** and **r** denote the set of the positions of all the nuclei $\mathbf{R} = (\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_N)$ and the electrons, $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{N_e})$. These expressions, as well as the remaining in this chapter, are given in atomic units.

The state of the system is described by its wave function Ψ, which is give by the solution to the time-independent Schrodinger equation:

$$\hat{H}(\mathbf{R}, \mathbf{r})\Psi(\mathbf{R}, \mathbf{r}) = \mathcal{E}\Psi(\mathbf{R}, \mathbf{r}) \tag{2.4}$$

where $\hat{H}$ is the Hamiltonian operator of the system and $\mathcal{E}$ its energy. For atomic systems, the Hamiltonian,

$$\hat{H} = \hat{T}_a + \hat{T}_e + \hat{V}_{ee} + \hat{V}_{ae} + \hat{V}_{aa}, \tag{2.5}$$

is given by the sum of the potential energy operators of the electrostatic interaction $\hat{V}_{ee}$, $\hat{V}_{ae}$ and $\hat{V}_{aa}$, and the kinetic energy operator of the nuclei:

$$\hat{T}_a = -\sum_{i}^{N} \frac{1}{2M_i} \nabla^2_{\mathbf{R}_i} \tag{2.6}$$

and the electrons:

$$\hat{T}_i = -\sum_{i=1}^{N_\mathrm{e}} \frac{1}{2} \nabla^2_{\mathbf{r}_i}. \tag{2.7}$$

M_i stands for the mass of the $i-$th nuclei.

The partial differential equation (2.4) is computationally challenging to solve numerically even for systems with few nuclei and electrons. In order to be able to simulate systems of interest, such as materials, molecules or surface problems, one usually makes some well-established approximations.

2.1.1 The Born-Oppenheimer approximation

The nuclei of the atoms involved in the problem are several orders of magnitude heavier than the electrons: the mass of the lightest possible nuclei, the proton, is roughly 2000 times heavier than an electron. Thus, the motion of the nuclei is much slower than the motion of the electrons. As a consequence to this difference in time scales, one can approximate the solution of the problem as if the electrons moved in a potential caused by stationary nuclei, adapting immediately to the changes of position of the nuclei, and the nuclei moved in a potential caused by a cloud of electrons. This is called the Born-Oppenheimer approximation.

Under this approximation, we factorize the wave function Ψ as:

$$\Psi(\mathbf{R}, \mathbf{r}) = \Theta(\mathbf{R})\Phi(\mathbf{r}|\mathbf{R}), \tag{2.8}$$

where $\Theta(\mathbf{R})$ is the wave function of the nuclei which depends solely on the position of the nuclei $\mathbf{R}$ and $\Phi(\mathbf{r}|\mathbf{R})$ is the wave function of the electrons subject to a fixed nuclei. Under these conditions, equation (2.4) can be decoupled into two partial differential equations, one that describes the motion of the electrons under fixed nuclei:

$$\hat{H}_e(\mathbf{r}|\mathbf{R})\Phi(\mathbf{r}|\mathbf{R}) = \left[\hat{T}_e + \hat{V}_{ee} + \hat{V}_{ea}\right] \Phi(\mathbf{r}|\mathbf{R}) = \varepsilon(\mathbf{R})\ \Phi(\mathbf{r}|\mathbf{R}) \tag{2.9}$$

and another one for the motion of the nuclei:

$$\left[\hat{T}_a + \hat{V}_{aa} + \varepsilon(\mathbf{R})\right] \Theta(\mathbf{R}) = \mathcal{E}\ \Theta(\mathbf{R}). \tag{2.10}$$

2.2 The atoms: The Potential Energy Surface

The Born-Oppenheimer approximation allows the description of the geometry of the atoms in materials and molecules provided the solutions to the electronic structure problem are known. The Hamiltonian operator for the atomic motion in equation (2.10) can be rewritten to introduce the concept of *potential energy surface* (PES) of the system, by realizing that the function of the atomic positions $E(\mathbf{R})$:

$$E(\mathbf{R}) = \hat{V}_{aa} + \epsilon(\mathbf{R}), \tag{2.11}$$

describes the effective potential in which the atoms move.

The notion of potential energy surface provides with a framework for working with molecular geometries, by defining a $3N$ dimensional manifold whose critical (i.e. zero gradient) points mark some of the atomic configurations of interest for applications in chemistry and materials science [11, 12]. A visualization of the potential energy surface and its critical points can be found in Figure 2.1.

Probably, the most interesting point of the PES is its *global minimum*: the configuration with the lowest potential energy will be the one in which the atoms would be in equilibrium at 0K temperature (neglecting ground state vibrations). For systems with many atoms, this configuration is usually difficult to find, which is the reason why many applications find a local minimum instead [13].

A *local minimum* of the potential energy is a configuration whose energy is lower or equal to the energies of neighboring configurations. Note the global minimum is a local minimum as well, and consequently, research studies often try to identify the global minimum by determining several local minima and comparing their energies [14]. At temperatures other than 0, it might be possible to find the system in different local minima depending on the conditions of the experiment: for example, the reactants, intermediates and products of a chemical reaction each correspond to a minimum of the potential energy surface. In addition, even when the geometry of the system is approximately known from experiment, it is still interesting to find the closest local minimum of the potential energy surface, since it has zero force on the atoms (see Section 2.2.1) and consequently makes the best estimation of the equilibrium configuration within that level of theory. A longer discussion on this regard can be found in Chapter 4.

Saddle points are also of interest. The reaction path between two minima is the path connecting them with the minimum energy among all possible paths, and it is called the *minimum energy path*. The maximum along this path is the geometry corresponding to the *transition state* of the reaction and it is necessarily a first order saddle point (this is, its Hessian matrix has one and only one negative eigenvalue).

2.2.1 Hellmann-Feynman Theorem

The force on the $i-$th atom $\mathbf{f}_i$:

$$\mathbf{f}_i = -\nabla_{\mathbf{R}_i} E(\mathbf{R}) \tag{2.12}$$

can be readily obtained from the electronic structure calculation by virtue of the Hellmann-Feynman theorem:

$$\nabla_{\mathbf{R}} \varepsilon(\mathbf{R}) = \langle \Phi(\mathbf{R}) | \left(\nabla_{\mathbf{R}} \hat{H}_e(\mathbf{R}) \right) | \Phi(\mathbf{R}) \rangle \tag{2.13}$$

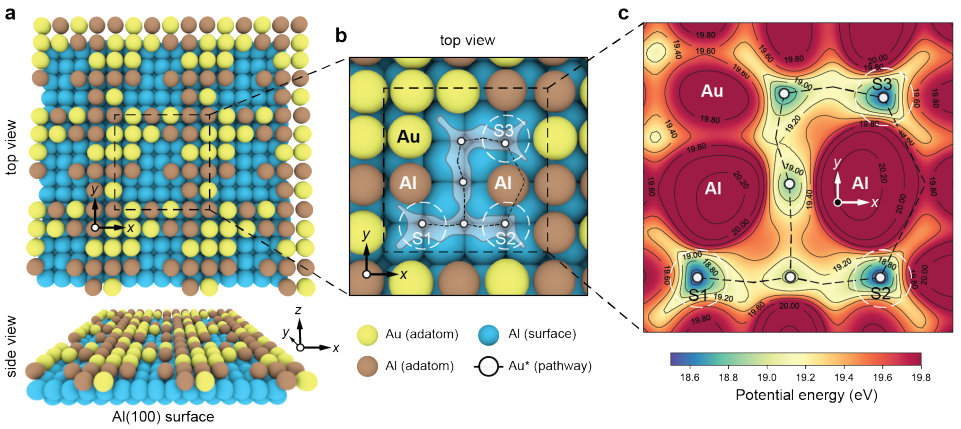


Figure 2.1: Geometry and potential energy surface of a moving gold on a aluminum (100) surface covered by other aluminum and god adatoms. Panel (**a**) shows the atomic structure of the slab represented in the other panels, with the aluminum atoms of the slab represented in blue and the gold and aluminum adatoms in yellow and brown, respectively. Panel (**b**) shows a closer look to the unit cell of the system, where the moving gold atom has been removed for better visualization and the critical points and their basins are marked in white. Panel (**c**) shows the 2-D projection of the potential energy surface as a function of the position of the moving gold atom. The colors in the contour plot show the potential energy surface of the system when the gold atom is constrained to that (x, y) position and it is relaxed on the z axis. The points S1 and S2 are local minima of the potential energy surface and the dashed line connecting them is the *minimum energy path*. The white dot along this path marks the saddle point representing the transition state. Source: Paper III: *An artificial intelligence-driven approach for the1exploration of potential energy surfaces.*

where the wave functions Φ represent the eigenvalues of the electronic structure Hamiltonian $\hat{H}_e$.

Note that the differential operator in $\nabla_{\mathbf{R}}\ \hat{H}_e(\mathbf{R})$ affects only to the Hamiltonian operator, and not the the wave function Φ. This means that once the electronic wave function Φ and the potential energy surface $E(\mathbf{R})$ have been obtained from the electronic structure calculation, the forces can be obtained with little computational effort. For this reason, the usual practice in the determination of critical points of the potential energy surface is to use gradient-based methods.

It is easy to see that a similar expression does not hold for higher derivatives. Even though the knowledge of the Hessian of the potential energy surface would be an advantage for many problems, its computation usually involves a finite differences approach, which can become very computationally expensive for systems with many atoms in the unit cell and many degrees of freedom. .

2.3 The electronic structure problem

We now turn to the problem of the electronic structure in equation (2.9). There are a number of approximations and methods in literature to solve this problem [15]. In the work presented in this thesis, the electronic structure problem has been solved using *density functional theory* (DFT) as implemented implemented in ASE [16, 17] and GPAW [18] , with the exception of paper III, where the Vienna Ab-initio Simulation Package (VASP) [19, 20] has been used.

It is worth noting that the methods introduced in the rest of this work depend only loosely on the choice of the approach to the solution of the electronic structure problem. As long as the Hellmann-Feynman theorem (2.13) provides a computationally cheap way of obtaining the forces on the atoms, any electronic structure method could be used.

In the rest of this chapter, the fundamentals of density functional theory are sketched and some numerical challenges are discussed. It does not intend to be a thorough discussion of the topic, but rather an introduction to the properties and challenges that will be discussed later on in this thesis. For further details, the reader is suggested to consult any of the many and excellent books and articles on the subject, as for example [15].

In order to facilitate the reading, the dependence on the degrees of freedom of the motion of nuclei **R** has been dropped from the notation in this section.

2.3.1 Density Functional Theory

The solution to the electronic structure problem in equation (2.9) is expressed in terms of the many body (anti-symmetric) wave function of all electrons $\Phi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{N_e})$; an object that becomes increasingly challenging from the point of view of computations as the number of electrons in the system N_e increases. In their foundational paper from 1964, Hohenberg and Kohn [8] showed that the solution to the electronic structure problem can be fully characterized by the ground-state electronic density $n_e(\mathbf{r})$ (a function of three variables) and proved that a variational principle exists for the ground state energy in terms of trial densities.

Based on this, Kohn and Sham proposed a practical way to efficiently solve the electronic structure problem in 1965 [9]. One can introduce an auxiliary system of non-interactig electrons under an external single particle potential $v_s(\mathbf{r})$ that needs to be determined so that the solution to non-iteracting problem (i.e. its ground state density $n_e(\mathbf{r})$ and energy ε) matches the solution of the original problem. The differential equation for the non-interacting system can then be decoupled into a system of single particle differential equations, called Kohn-Sham equations:/

$$\left[-\frac{1}{2}\nabla^2 + v_s(\mathbf{r})\right] \varphi_i(\mathbf{r}) = \epsilon_i \, \varphi_i(\mathbf{r}), \tag{2.14}$$

where φ_i are the single particle wave functions (or Kohn-Sham orbitals) and ϵ_i are the single particle energies (or Kohn-Sham energies).

It is easy to show that the single-electron reference potential $v_s(\mathbf{r})$ introduced in equation (2.14) can be written as:

$$v_s(\mathbf{r}) = v_{ext}(\mathbf{r}) + \int \frac{n_e(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|} d\mathbf{r}' + \frac{\delta E_{xc}[n_e]}{\delta n_e(\mathbf{r})}, \tag{2.15}$$

where v_{ext} is the single-electron external potential each electron feels (i.e. the potential created by the nuclei in equation (2.3)), the second term is the potential caused by the average electronic distribution represented by the density $n_e(\mathbf{r})$, and $E_{xc}(\mathbf{r})$ is the *exchange-correlation* functional, which is discussed in Section 2.3.2.

Thus, given an expression for the exchange-correlation functional E_{xc}, the Kohn-Sham equations (2.14) can be solved numerically to find the total energy ε and the density n_e of the interacting system, as functions of the Kohn-Sham energies:

$$\varepsilon = \sum_{i=1}^{N_e} \epsilon_i, \tag{2.16}$$

and orbitals

$$n_e(\mathbf{r}) = \sum_{i=1}^{N_e} |\varphi_i(\mathbf{r})|^2. \tag{2.17}$$

These equations provide a framework for computing the total energy of the electronic structure problem for a particular configuration of the nuclei **R** that can then be used, as it has been mentioned before, to obtain the potential energy surface and its derivatives, the main object of interest in this thesis.

The quality of the description of the potential energy surface provided by the DFT depends on two crucial factors: the level of approximation in the exchange-correlation potential and the details of the numerical implementation of the solution to the Kohn-Sham equations. In the following, these two factors are briefly discussed, highlighting the main choices for the PES models presented in the rest of this thesis along with the arising potential challenges.

2.3.2 Exchange-correlation

Despite of the fact that the density functional theory framework shows that a universal exchange-correlation functional exists so that the density solution to the Kohn-Sham matches exactly the density of the electronic structure problem (equation (2.9)), the exact exchange-correlation potential is not known. Practical calculations of the electronic structure of atomic systems use approximations to model the exchange correlation functional in order to approximate the value of physical quantities.

There is a vast number of approximations or exchange correlation functionals in literature (see, for example, the review presented [15], chapter 5), and it is still an ongoing research topic. Different approximations require different amounts of computational resources and are known to estimate different physical quantities at different levels of accuracy.

The simplest level of approximation is the local density approximation (or LDA) functional, introduced by Kohn and Sham in their original paper [9]:

$$E_{xc}\left[n_e\right] = \int n_e(\mathbf{r})\epsilon_{xc}[n_e(\mathbf{r})]\ d\mathbf{r}, \tag{2.18}$$

where the exchange-correlation energy density $\epsilon_{xc}[n_e]$ is obtained from the homogeneous electron gas.

The accuracy of LDA functional can be improved by including the gradient of the density in the exchange-correlation expression, leading to a family of functionals known as the generalized gradient approximation, or GGA.

The GGA functionals are known to improve the description of binding energies and bond distances, as compared to the LDA. Thus, in this thesis, the LDA has been used for method development and different GGA flavors have been used to improve the description of the physical quantities when necessary. We have used the PBE (Perdew-Burke-Ehorenhoff) functional [21], probably the best-stablished GGA functional, together with two revised versions: PBEsol [22] for the description of solids and RPBE [23] for the description of molecules on surfaces.

2.3.3 Numerical solution to the Kohn-Sham equations

The differential equation for the Kohn-Sham orbitals and energies (2.14) is a non- linear differential equation: it depends on the reference potential v_s which in turn depends on the density through expression (2.15) and the density itself depends on the Kohn-Sham orbitals, as shown in equation (2.17).

This means that the Kohn-Sham equations need to be solved numerically. Even when an already existing implementation of a numerical method is used, there are usually a number of choices that are left to the user, who in turn needs to make a compromise between the accuracy of the result and its computational cost when making the choice. In this section, the numerical methods that have been used in this thesis are sketched, with a particular focus on those aspects that have presented a challenge in the course of finding the critical points of the potential energy surface.

These equations usually are solved in a self consistent way. A diagram illustrating the self consistent field (SCF) iteration can be found in Figure 2.2. Starting with an initial estimate for the Kohn-Sham orbitals, these determine an estimate of the effective potential v_s. Disregarding the dependence of the effective potential on the orbitals, the Kohn-Sham energies and orbitals can then be found by solving the linear eigenvalue problem in equation (2.14). The new estimate of the orbitals result in a new potential, closing the loop. The iteration continues until the estimates of the change between two steps of the quantities of physical interest, such as the total electronic energy ε, the forces on the atoms **f** and the electronic density n_e, falls under a given threshold.

The first step in solving the linear eigenvalue problem is discretization: this is, in order to solve the differential equation, solutions must be represented in some way. As an example, the work presented in this thesis, plane wave basis and linear combination of

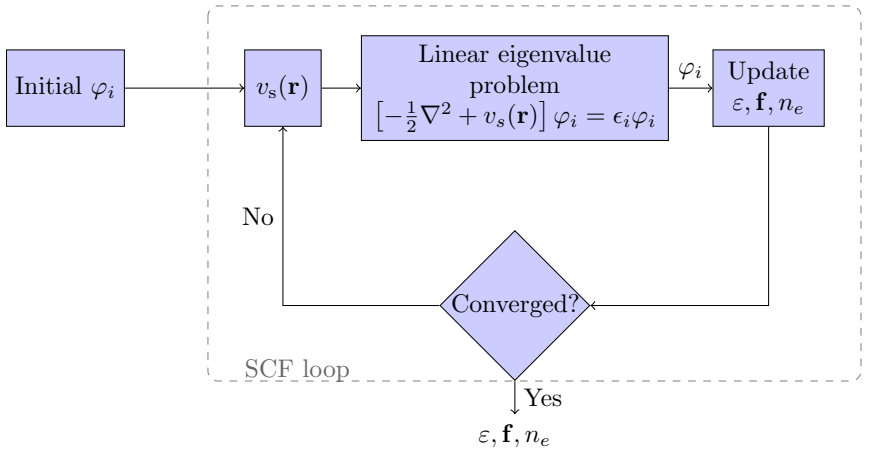


Figure 2.2: Diagram illustrating the self consistent field iteration, the procedure used to solve the Kohn-Sham equations. The method takes an initial estimate of the Kohn-Sham orbitals as an input and returns the electronic energy ε, the forces on the atoms **f** and the electronic density n_e as an output.

orbitals [24] have been used. The discretization of the wave function into a basis set involves an approximation, and for this reason, it must be treated carefully to ensure the results of the electronic structure calculation are converged enough to represent the target property with enough accuracy. This can be achieved by tuning the parameters most basis sets involve to reach a good compromise between accuracy and computational speed. An example of a potential source of error is the egg-box effect [25], in which the fact that the wave functions are represented by a fixed grid can lead to the break of the translation symmetry when the nuclei are translated with respect to the grid.

Since the Kohn-Sham orbitals need to be orthogonal functions to each other, the wave functions in the core regions, i.e., close to the nuclei, are usually highly oscillatory, and thus, difficult to discretize (their representation would involve large basis sets).

A common way to deal with this problem is known as the *projector augmented-wave method* (PAW) [18, 26], where the implementation avoids working with core electrons and work only with a modified representation of the valence wave functions that is is smooth everywhere. The modified wave functions only differ from original ones in an augmentation sphere around each nuclei but match exactly outside of the spheres. This method allows one to solve the Kohn-Sham problem with smaller basis set sizes, improving both accuracy and computational speed. For algorithms navigating PES it is important to take into consideration that the PAW method breaks if the spheres overlap too much, therefore, it should be avoided placing the atoms too close. Some implementations, like GPAW, raise an error message in these situations.

Solving the linear eigenvalue problem for the single particle Hamiltonian in the Kohn-

Sham equation (also called diagonalization of the Hamiltonian) is typically the most computationally expensive part of any density functional theory code. Even though some methods with lower scalings exist in literature [27, 28], density functional theory codes typically exhibit cubic scaling of the computational time with the number of orbitals to be computed $O(N_e^3)$ [13], which comes as a consequence of the code numerically enforcing the orthogonality of the Kohn-Sham orbitals. Since the number of orbitals required in a problem is usually proportional to the number of atoms, the computation time in density functional theory usually scales with the cube of the number of atoms in the unit cell.

Numerical artifacts stemming from the self consistent field iteration or the choice of the basis set may produce spurious non-physical behaviour in the potential energy surface. It is important to choose the parameters in an electronic structure calculation in such a way these artifacts are minimized when they are important to the property that is being modelled.

2.4 Afterword

The approximations and methods introduced in this chapter have enabled researchers to successfully model, understand and make predictions for a plethora of problems involving atomic level interactions over the past decades. Despite of this success, the computational resources necessary to describe many of the systems of interest remain vast. For instance, the description of the adsorption of a molecule on a surface typically involves dozens of atoms, and even with a good insight, finding the closest minimum might require dozens of potential energy calculations involving tens of self consistent field steps each and hundreds of electrons.

Hence, such calculations are usually run at super-computing facilities and even then researchers often have to impose limitations to the number of atoms considered, the number of configurations explored or the level of accuracy used in the exchange correlation description. In the following chapters of this thesis, we discuss and introduce ways of further reducing the computational cost of the description of atomic systems.

3 Machine Learning for materials, molecules and surfaces

The computational tools presented in the previous chapter, combined with the rapid growth of computer power over the last decades, has allowed the computation of the properties of millions of materials and molecules worldwide. These calculations are gathered in large databases, such as OQMD [29, 30], AFLOWLIB [31], Materials project [32], NOMAD [33] or CMR [34] and C2DB [35, 36]. Often, these calculations are part of large scale computational screening studies, where the properties of thousands of materials are computed in order to find candidates for their use in specific applications. As a result, new materials are discovered every year, and they are added to open materials repositories.

Since the number of possible materials has been estimated to be around 2 trillion (a couple of orders of magnitude give or take) [1] and speculating that the number of possible molecules could yield a similar count, it is clear that new strategies for materials and molecules discovery would be desirable. The combinatorial nature of the structure determination (or potential energy optimization) for systems with a large number of atoms in the unit cell poses a problem of similar nature.

The availability of large scale data together with the current level of computational resources present new opportunities in the field of materials and drug design [37]. Nowadays, we are seeing an increase in effort to utilise computers to recognize statistical patterns in materials databases, that may increase our understanding or allow as to make faster predictions. In this line, in recent years, many successful examples of machine learning for materials, molecules and surfaces have been published (see next section for some examples of references).

In this chapter, the theory behind the machine learning methods in this thesis is discussed. First, a general discussion of machine learning and supervised learning in particular as a set of methods is introduced. This is a very broad topic, and the interested reader is referred to classical books like by Bishop (2006) [38] or Hastie, Tibshirani and Friedman (2009) [39]. The structure of the section has been inspired by the presentation in Goodfellow *et al.*, chapter 5 [40]. Then, the two machine learning methods used in this thesis: neural networks and kernel methods; are reviewed. Finally, the representation of materials and molecules as an input to the machine learning method is introduced and its importance is discussed.

3.1 Machine Learning

One of the most widely spread definitions of machine learning was given Thomas Mitchell in 1997: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.” [41]

The task T is typically any task for which there is no clear way to solve it by a detailed program written by a human, or there is an advantage in not doing so, and can benefit from the amount and type of existing data. Recently published tasks in the field of data-driven discovery of materials and molecules include the prediction of the electronic density $n(\mathbf{r})$ given the structure of a molecule [42], predicting the heat of formation of molecules and materials [43–45] given the positions of their atoms and their stoichiometries, the development of new exchange-correlation functionals [46–49], the screening of materials and molecules of solar cell applications given their stoichiometries [50, 51], the generation of computationally fast force-fields to speed up molecular dynamics simulations [52, 53], among others.

The experience E in the context of materials design is the set of materials whose properties are already available, also referred to as the training set. This dataset could comprise a subset of the materials in some experimental [54] or computational databases (as mentioned above) [29–36] or could be generated on-the-fly, as the task progresses, by the requirements of the algorithm itself (known as *active learning*)).

Depending on the kind of information for each piece of data the computer program is allowed to use, machine learning algorithms are usually classified into two groups:

> *unsupervised learning*, where each data point is represented by a set of features and the method's task is to learn patterns present in the structure of the dataset, and
>
> *supervised learning*, where data points are represented by features and targets and the task of the method is to learn the the mapping from features to targets.

Most methods in this thesis belong to the second type.

The performance measure P depends on the task at hand, but most supervised learning methods measure the accuracy of the model in some way. Common accuracy measures of accuracy for supervised methods f mapping n features $\{\mathbf{x}_i\}_{i=1}^n$ into n targets $\{y_i\}_{i=1}^n$ include the mean absolute error (MAE):

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - f(\mathbf{x}_i)|}{n} \tag{3.1}$$

and the root mean square error (RMSE):

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2}{n}}. \tag{3.2}$$

In real world applications, we are interested on the performance of the program on unseen data points, as opposed to on the experience presented. In other words, we are interested on the ability of the model to generalize from experience, and the performance measure P should capture such interest. A common practice is to evaluate the performance of the method on previously unseen data, known as the *test set*.

Successful machine learning models will optimize their capacity (that is, their complexity in the sense of the amount of solutions they are able to represent) to obtain good

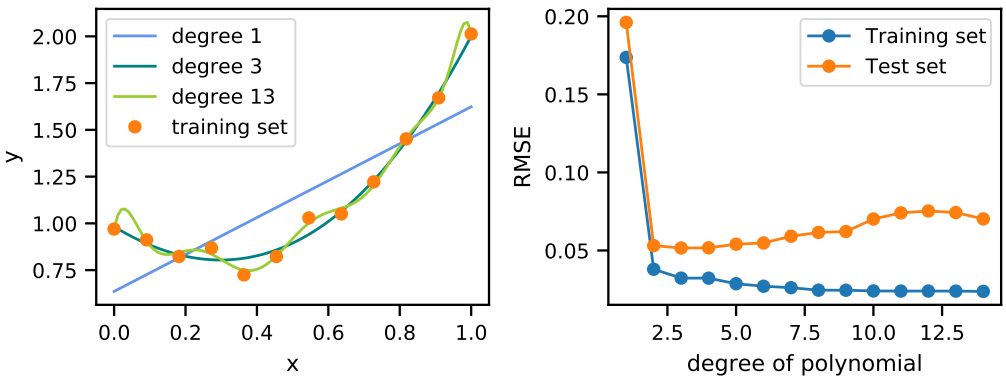


Figure 3.1: Illustration of the capacity and the generalization error of machine learning models for the example of a polynomial fit. The training and test data sets have been generated assuming the underlying function to be $f(x) = (x-1)^3 + 4(x-1/2)^2 + 1 + \xi$, with ξ representing white noise with standard deviation 0.05. Low degree polynomials underfit the data (see the linear model in the left panel) while choosing a high enough degree of the polynomial overfits the data, resulting in a model that goes through all the points, but does not generalize correctly (see the 13^{th} degree polynomial in the left panel). A model whose complexity (the degree) is in between generalizes best from the data. This is shown in the right panel: As the complexity of the model increases, the error in the training set falls, but the error in the test set shoes a U shape. Models with low capacities cannot capture the underlying complexity of the data, so they are biased. Models with high capacity can represent the noise in the data as well as the actual trends, resulting in models that do not generalize properly (overfiiting). The training set and the test set in the right panel have been randomly generated and have 20 and 200 points, respectively.

performances on both the training and the test sets. A general trend in machine learning is that as the capacity of the model increases, the error in the training set decreases, while the error in the test set exhibits a U shape: low capacities cannot represent the patterns in the data enough to predict the test set (*underfitting*) while high capacities learn the noise in the training set together with the underlying patterns, increasing the error in the test set (*overfitting*). The goal when training a machine learning model is to find a balance between underfitting and overfitting. This problem is illustrated over the problem of fitting a third order polynomial with noisy data in Figure 3.1.

Many models can be dialed to show preference for the certain solutions, only choosing the least preferred ones if there is strong evidence in the training set favoring them. This can be used to reduce the test error while still having a good performance in the training set. This action is called *regularization* [40]. A common example is L^2 norm regularization, where a penalty is imposed for solutions with large L^2 norms of the parameters. For example, in the case of multi-variable linear regression, this prevents having large weights, producing models that generalize better.

More generally, many machine learning methods have additional parameters that are

not determined by the learning procedure themselves (any value of them will produce a learning method) but that determine the behaviour of the algorithm. These are termed *hyperparameters* and examples include the degree of the polynomial in a polynomial regression (see Figure 3.1) or the strength of the penalty in L^2 regularization. The hyperparameters of an algorithm can be tuned to obtain better generalization results, to ensure the model produces better predictions.

Usual practice in the machine learning field proceeds as follows: The parameters of the machine learning method are optimized to reduce the error in the training set, and then the machine is tested on yet unseen data. The data set that is not used in the parameter training but is still used for decision making is termed *validation set*, and it is often chosen as a subset of the training set [39].

As the last part of the machine learning definition states, in machine learning applications the error in the training and the test set should reduce as the training set size increases. This is usually visualized by representing the error in test set as a function of the training set size, which is called the *learning curve*.

For many machine learning problems with noise-free data, the error decays as a power law of the size of the training set in the limit of large training set sizes [55]. As noted by Huang and von Lilienfeld [56, 57], such power law becomes a linear relationship in logarithmic scale:

$$\log(\text{Error}) = a - b\log(n), \tag{3.3}$$

where n is the size of the training set. The parameters a and b encode the ability of the algorithm to extract patterns from data for moderate training set sizes.

The desirable properties of the model include a small off-set and a large negative slope as well as to continue with the linear trend in a logarithmic scale instead of saturating to some error as the number of instances in the training set increases. The first properties, i.e. small a and large b, mean the model will need less instances to achieve the accuracy necessary for the target application. The smaller the requirement on the size of the training set the better, since this reduces the computational cost of the training process, but more importantly it is very often very difficult or impossible to increase the size of the training set at will (for example, because the training instances are DFT calculations involving many atoms, which can be computationally very expensive). Better off-sets and slopes can be achieved by incorporating prior information into the model [56, 58]: enforcing the symmetries instead of having the model learning them [59, 60] or enforcing a resemblance between the functional form of the model and the target function: such as using descriptors whose behaviour is similar to the Hamiltonian by including the appropriate interaction ordering information (that is, explicitly including information about 2-body, 3-body, etc interactions) [56, 58]. Enforcing symmetries and target similarity can be achieved by enforcing these properties in the method itself or by choosing a representation of the elements in the training set that already incorporate them (see Section 3.4).

Saturating learning curves can become an issue since they may prevent the model from achieving the required accuracy for the target application. The cause might be

the presence of noise in the training data: in this case, the error will saturate at the value comparable to the noise in the training set. However, saturating learning curves can be found even in noise-free contexts: in this case they are usually due to underfitting: the model is not flexible enough to fit all the data and it reaches a compromise. A particularly concerning case is the one of degenerate descriptors (or even models!): if two physically different inputs are mapped into the same value at any point of the process, the machine learning method will not be able to distinguish them at all. It is easy to see this will lead to a saturating learning curve. This issue is further discussed in Section 3.4.

The concepts introduced in this section are illustrated in the rest of the chapter with two examples of machine learning methods: neural networks and kernel methods (in particular, Gaussian processes).

3.2 Neural networks

Neural networks are a class of machine learning algorithms that have been gaining popularity in recent years. They are very flexible methods, which have proven successful in difficult tasks involving the approximation of complex non-linear functions of complex, multidimensional and often structured inputs [40].

Originally inspired on the way the human brain works, artificial neural networks are composed of many small and interconnected processing units called neurons or units [40]. These neurons are often arranged in several *layers*, allowing for the models to be very flexible. In most supervised learning applications, the inputs to the network are regarded as a units to be filled with data and the output of the network is the output of the last layer of units. The units in the layers in between the input and the output are called *hidden units*. As an example of how a typical unit looks, hidden layers in feed-forward neural networks (a common neural network architecture for supervised learning) are a parametrized non-linear function of the units in the previous layer of the form:

$$\mathbf{h}_k = g\left(\mathbf{W}_k^T \mathbf{h}_{k-1} + \mathbf{b}_k\right) \tag{3.4}$$

where $\mathbf{h}_k$ and $\mathbf{h}_{k-1}$ are the outputs of layers the k-th and the $k-1$-th layers, $\mathbf{W}$ and $\mathbf{b}$ are the weights matrix and the bias vector, and $g(\cdot)$ is a fixed non-linear function (in the sense that it does not depend on any parameters) non-linear function called the *activation function*. Typical examples of activation functions include the rectified linear unit $g(z) = \max(0, z)$, the sigmoid function or the hyperbolic tangent.

Thus, the representation of complicated non-linear functions is achieved as a convolution of many layers of linear transformations plus activation functions. The parameters to the linear transformations $\mathbf{W}_k$ and $\mathbf{b}_k$ are then optimized to fit the data. The neural network is given some input examples that are then used to produce an estimate of the loss function (that is, the error). The gradient of the loss function is estimated using the chain rule over all the convolutions with a method known as *back propagation*. The lost function is then minimized with a gradient based optimizer to find the optimal parameters. It is usual to use a stochastic gradient-based optimizer to increase the chances of reaching the global optimum of the loss function by using the noise to avoid getting trapped in a local minimum.

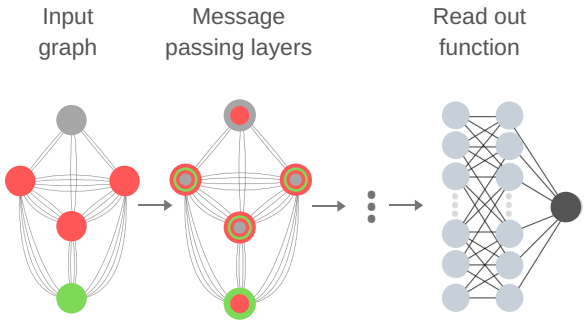


Figure 3.2: Visualization of a message passing neural network for predicting the heat of formation of materials. The input to the neural network in this case is a graph, whose nodes (and possibly edges) are encoded by a vector. The state of the nodes and the vertices is updated by passing messages from its neighbors a number of times. Finally, the output of the neural network is given by a non-linear read out function, containing a hidden fully connected layer.

In recent years, neural networks have proven to be successful machine learning methods for the modelling of structured data, such as images or time series [40]. In Paper I: *Materials property prediction using symmetry-labeled graphs as atomic-position independent descriptors*, we have taken advantage of the flexibility of these models to use graphs as descriptors of the instances in the training set (see Section 3.4 for more information on this representation). For this purpose, we have used a graph-convolutional or *message passing neural network*, that have proven themselves to be very successful in the prediction of the properties of materials and molecules given their graphs [42, 43, 61–63].

Figure 3.2 illustrates the architecture of these kind of artificial neural networks. Message passing neural networks are composed of two distinct parts: The message passing layers and the read out function. Let us consider an input graph for which the state of its nodes is represented by vectors $\mathbf{h}_v^0 \in \mathbb{R}^d$, and the state of its edges by vectors $\mathbf{e}_{v,w}^0$, where v and w run over the nodes of the graph. At each message-passing layer k a message $\mathbf{m}_v^{k+1}$ is received at every node v from the nodes it is connected to, $w \in N(v)$:

$$m_v^{k+1} = \sum_{w \in N(v)} M_k \left(\mathbf{h}_v^k, \mathbf{h}_w^k, \mathbf{e}_{vw}^k \right) . \tag{3.5}$$

This message is computed as the message function $M_k(\cdot)$, which depends on the current state of the receiving node $\mathbf{h}_v^k$, its neighbors $\mathbf{h}_w^k$, and the edges connecting them $e_{v,w}^k$.

Then, the receiving node v updates is state using the state transition function $S_k(\cdot)$,

which is a non-linear function of its current state and the message received:

$$\mathbf{h}_v^{k+1} = S_k\left(\mathbf{h}_v^k, \mathbf{m}_v^{k+1}\right). \tag{3.6}$$

It is also possible to update the state of the edges accordingly, by using the edge update function $E_k(\cdot)$ and the states of the edge and the nodes it connects:

$$\mathbf{e}_{vw}^{k+1} = E_k(\mathbf{h}_v^k, \mathbf{h}_w^k, \mathbf{e}_{vw}^k). \tag{3.7}$$

The output of the layer is, thus, a new graph, where the states of the nodes and edges have been updated. It is usual to concatenate several message passing layers, so that the state of each node "learns" the properties of its surroundings. Finally, the updated state of the graph is used to predict the output of the neural network y, modelled as some read out function $R(\cdot)$ of the stats of all the nodes $\{h_v\}$ in the graph G:

$$y = R\left(\{\mathbf{h}_v \in G\}\right). \tag{3.8}$$

In general, the underlying functions M_k, S_k, E_k and R will be non-linear functions, and can be parametrized as one or several convoluted units in the form of 3.4. In this way, message passing neural networks are able to learn a vectorial representation of the information displayed as a graph that best predicts the target in a flexible "all-in-one" way.

The flexibility of neural network models results in a universal approximation framework: provided enough hidden layers are used, a neural network is able to approximate any non-linear function [39, 40]. This flexibility, however, comes at a cost. The complexity of most neural network models makes them difficult to interpret, since inputs are processed multiple times by different non-linear units. Often, the result of the fit is some sort of black box, from which good predictions can be extracted but no further conclusions can be inferred.

This flexibility also means that the size of the training set size that is needed to produce small generalization errors is large compared to other methods. As a consequence, neural networks also often turn out to be computationally demanding to train (despite it being fast to evaluate).

3.3 Kernel methods

The central characteristic of these methods is the use of a function of two points in the input space called the *kernel*: $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$, $k(\mathbf{x}, \mathbf{x}') : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. This function encodes a non-linear transformation of the arguments and an inner product of the result in the transformed space, even though the functional form of the transformation might not be explicitly available. This characteristic allows to use complicated non-linear transformations of the arguments. By regarding the kernel as a similarity measure between two points in input space, these methods provide a flexible non-linear way of interpolation.

Kernel methods can be used to generalize linear regression of non-linearly transformed arguments, in a method called the *kernel ridge regression*. The basic equation for kernel ridge regression reads [38]:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i). \tag{3.9}$$

Note that since kernel ridge regression has many parameters α_i as points in the training set $\mathbf{x}_i$, it is possible for the prediction $f(\mathbf{x})$ to go through all the points in the absence of regularization.

The coefficients α_i in equation (3.9) can be expressed in matrix form as follows. Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ denote the design matrix and let us define $k(\mathbf{x}, \mathbf{X}) = (k(\mathbf{x}, \mathbf{x}_1), k(\mathbf{x}, \mathbf{x}_2), \ldots, k(\mathbf{x}, \mathbf{x}_n))$ and $k(\mathbf{X}, \mathbf{x}) = (k(\mathbf{x}, \mathbf{X}))^T$. Let $k(\mathbf{X}, \mathbf{X})$ denote the Gram matrix:

$$k(\mathbf{X}, \mathbf{X}) = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \ldots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \ldots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \ldots & k(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix}. \tag{3.10}$$

Since the kernel function is in fact the result of an inner product in a transformed space, the Gram matrix is the matrix collecting the inner products of a basis, in this case, of the points in the training set used as vector basis, hence the name of this matrix. Consequently, as it follows from the basic properties of inner products, the Gram matrix $k(\mathbf{X}, \mathbf{X})$ must be symmetric and positive (semi-)definite[1].

The prediction of the kernel ridge regression method can them be written as:

$$f(\mathbf{x}) = k(\mathbf{x}, \mathbf{X})\, C^{-1}\mathbf{y} \tag{3.11}$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, C is the regularized Gram matrix, $C = k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbb{I}$, σ_n is the regularization and $\mathbb{I}$ is the identity matrix.

The prediction in equation (3.11) can be obtained from a different point of view: in the Bayesian framework, this equation can be interpreted as the average prediction of a *Gaussian process regression* (GPR).

In Gaussian process regression, the model for the data is a stochastic process with label $\mathbf{x}$, which assumes that all the variables $f(\mathbf{x})$ are given by a multi-dimensional Gaussian distribution. Hence, the model is a probability distribution over functions of $\mathbf{x}$ and it is fully specified by a mean function of $\mathbf{x}$ and a covariance function of $\mathbf{x}$ and $\mathbf{x}'$.

In this Bayesian framework, a prior probability distribution over function space before any data is presented is assumed. If the prior distribution is assumed to be Gaussian, it can be fully specified by its mean function, $m(\mathbf{x})$, also called the *prior function* and its covariance function, $k(\mathbf{x}, \mathbf{x}')$, also called the *kernel*:

$$p(f) = \mathcal{N}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right). \tag{3.12}$$

[1] These two characteristics ($k(\mathbf{X}, \mathbf{X})$ being symmetric and positive (semi-)definite), become very handy when debugging the implementation of kernels that include derivatives.

As for the Bayesian likelihood $p(\mathbf{y}|f(\mathbf{X}))$, that is, the probability of the targets **y** if the model f is assumed to be true, the usual assumption is taken: observations are independent and each observation is distributed as a Gaussian distribution $p(y_i|f(\mathbf{x})) = \mathcal{N}(f(\mathbf{x}_i), \sigma_n^2)$ around the prediction with variance σ_n^2, which models the noise in the data.

The *posterior probability* $p(f|\mathbf{y}, \mathbf{X})$, that is, the probability of the model's output given the training data, can be obtained from Bayes theorem:

$$p(f|\mathbf{y}; \mathbf{X}) \propto p(\mathbf{y}|f(\mathbf{X}))p(f). \tag{3.13}$$

It is easy to see that the posterior probability will be a Gaussian process itself (i.e. a set of Gaussian distributed random variables labelled by the continuous variable **x**), hence the name of the machine learning method. It can be shown that the mean of the Gaussian process is the function [38, 64]:

$$\mu(\mathbf{x}) = m(\mathbf{x}) + k(\mathbf{x}, \mathbf{X})\, C^{-1}\, (\mathbf{y} - m(\mathbf{X})) \tag{3.14}$$

and the variance at a given point **x** is:

$$\sigma^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, \mathbf{X})\, C^{-1}\, k(\mathbf{X}, \mathbf{x}). \tag{3.15}$$

Thus, the predictions of the Gaussian process regression are determined exclusively by the data **X** and **y**, the prior expectations of the user on the value $m(\mathbf{x})$ and the correlation model between points $k(\mathbf{x}, \mathbf{x}')$, as well as the noise value σ_n. An example of Gaussian process regression can be found in Figure 3.3.

It is interesting to note that the equation for the average prediction of the Gaussian process (3.14) recovers the expression for the prediction of kernel ridge regression (3.11) when $m(\mathbf{x}) = 0$, exposing the connection between the two methods. The Bayesian framework offers, in addition to the prediction provided by the kernel ridge regression, a measure of the uncertainty on the prediction, since the variance $\sigma^2(\mathbf{x})$ can be used to obtain a range of likely values of the prediction. We note that the uncertainty depends only on the distribution of the data and the prior assumption on the correlation function between points, being a measure of how correlated the test point **x** to the training set according to the kernel function k.

The Bayesian approach in Gaussian process regression also provides a way to build better models. It is usual to encode the prior assumptions in a flexible way by considering a family of prior and kernel functions that depend on a set of hyperparameters, rather than by fixed functions. The hyperparameters of the Gaussian process $\boldsymbol{\theta}$, which may include the regularization σ_n in addition to the hyperparameters in the prior and the kernel, can be determined by maximizing the marginal likelihood [64]:

$$\log p(\mathbf{y}|\boldsymbol{\theta}; \mathbf{X}) = -\frac{1}{2}(\mathbf{y} - \mathbf{m})^T C^{-1}(\mathbf{y} - \mathbf{m}) - \frac{1}{2}\log\det(C) - \frac{n}{2}\log 2\pi, \tag{3.16}$$

where $\mathbf{m} = m(\mathbf{X})$. Gaussian process are one of the few machine learning methods for which not only an analytic closed expression is available for the marginal likelihood, but

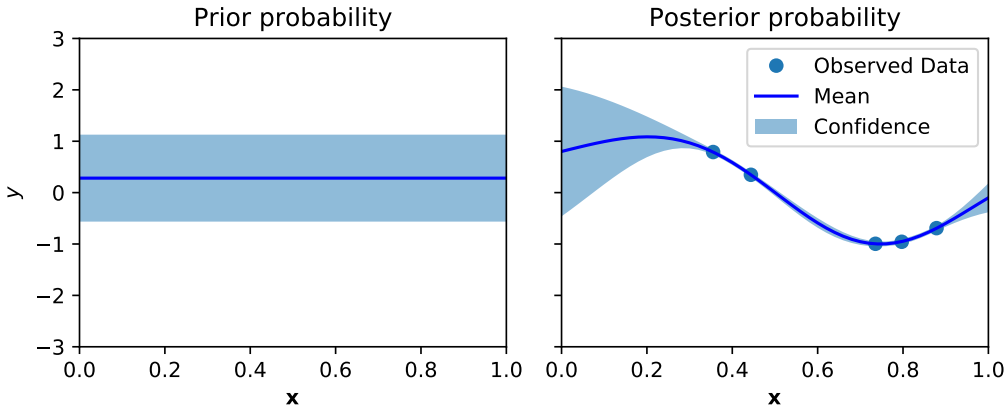


Figure 3.3: Example of Gaussian process regression. The left panel shows the prior probability distribution before any data is known by the model and the right panel shows the posterior distribution, i.e. the fit to the data. The blue line shows the average prediction and the shaded area, the uncertainty. A prior constant function has been used, together with the squared exponential kernel as a function of Cartesian coordinates (see Chapter 5 for details). The hyperparameters used to build the probability distributions in both panels have been obtained by maximizing the marginal likelihood.

also for its partial derivatives with respect to the hyperparameters θ [64], allowing to find the optimal values of the hyperparameters by optimizing the marginal likelihood with a gradient-based optimizer (see Chapter 4) instead of falling back to cross-validation.

Furthermore, in the context of Gaussian process regression for materials and molecules, it is usual to use a descriptor or fingerprint for each point (see Section 3.4), which might be a family of non-linear transformations $\mathbf{x} = \varrho(\mathbf{z}; \theta_{\text{fp}})$ of some simpler descriptor $\mathbf{z}$ and some hyperparameters θ_{fp}. Since the kernel is a non-linear transformation, one can regard the kernel as a function in the space of the simpler descriptor $k(\varrho(\mathbf{z}), \varrho(\mathbf{z}'))$ and see the hyperarameters of the fingerprint θ_{fp} as hyperparameters of the kernel. In this way, the Bayesian approach can be used also to determine the hyperparameters of the fingerprint that describe the target best.

The Bayesian connection and the access to analytical expressions to compute quantities of interest such as the uncertainty or the optimal hyperparameters make the Gaussian process regression a method that is easily interpretable. The appropriate choice of a kernel function can be used to enforce the prior knowledge on the data in a clear and often unbiased way, easing the learning process.

The main drawback of this method is, however, the poor scaling with the size of the training set. Equations (3.11), (3.14), (3.15) and (3.16) involve the inverse of C. Since C has n^2 entries, the computational cost of solving the linear algebra problem typically scales cubically with the size of the training set size $O(n^3)$ (at least, if the usual Cholesky factorization is used [65]) and the cost of storing the matrix scales quadratically wit the size of the training set $O(n^2)$ [64]. As a consequence, the cost of these

methods becomes prohibitively expensive for large data sets.

3.4 Descriptors and fingerprints

So far in this chapter, the methods that have been used to generalize from atomic structure calculations have been introduced. However, the choice of the shape of the input into such methods is of similar importance as the choice of the method itself when machine learning models are built.

However, for most tasks in computational materials science and chemistry, the choice of the input is not straightforward. As discussed in Section 3.1, a good descriptor can reduce the number of examples needed for training, while a bad one can prevent the model from reaching the target accuracy.

One can think of many of the descriptors presented in literature as “foldings” of the input space (as described, for example, with Cartesian coordinates) that map physically equivalent regions into each other. This way a large number of regions can be described with less examples by the subsequent machine learning model, which will generalize non-locally in the input space.

It is crucial that the mapping into the same region only happens for configurations that are physically equivalent. The wish list of properties of a good atomic descriptor includes the following [66–69]:

1. *complete*: The descriptor should include all the information necessary to model the targets. Failure to comply with this requirement will result in underfitting.
2. *non degenerate*: Two physically distinct atomic structures should have different descriptors. Otherwise, the machine learning method will interpret the differences in the target values as noise in the data.
3. *unique*: Physically equivalent configurations should have the same descriptor: descriptors should be invariant under rigid translations or rotations of the full system, under the transformations in the point group of the molecule or the space group of the material and under the permutation of the indexing of two atoms of the same species.
4. *descriptive*: Similar structures should have similar descriptors. In this sense, it would also be desirable the descriptors were *continuous* (since discontinuities in the descriptor may propagate through the machine learning algorithm to produce discontinuous predictions of physical properties) and *compact*, that is, introducing the minimum amount of redundant unnecessary features.
5. *computationally efficient*: Otherwise, it might be faster and more accurate to run the electronic structure calculation instead.
6. *general*: They should be applicable to many different atomic systems instead of being system specific, making the learning transferable. In addition, being useful to model more than a single property of the material is a beneficial property of descriptors.

The first two properties are fundamental to any representation: if the descriptor is not complete and non degenerate, the machine learning model may not be able to extract all the patterns present in the underlying data because of the biased representation. The relevance of the other properties may depend on the machine learning algorithm used for the purpose and the target application. For example, the application could use a machine learning algorithm that is invariant under a certain symmetry transformation instead of enforcing the symmetry on the fingerprint, or the descriptor of the atomic structure used may be computationally expensive and not transferable (i.e. general) in an application for which no deterministic algorithm is known.

There is an increasing amount of descriptors for atomic structures, be it molecules, materials or both; proposed in literature, along with an also large number of descriptors of other inputs (such as the density of states) relevant for molecular and materials science. Among the more popular fingerprints one finds the Coulomb matrix [70], the bag-of-bonds descriptor [71], the Many Body Tensor Representation (MBTR) [67], the Oganov-Valle fingerprint [72], the Behler-Parrinello Atom-Centered Symmetry Functions (ACSF) [73] or the Smooth Overlap of Atomic Potentials (SOAP) [68]. These and many other fingerprints have also been benchmarked against each other [69, 74, 75] and compiled into libraries [69].

In the following, the descriptors used in this thesis are discussed. First, the Cartesian coordinates, one of the simplest atomic descriptors, is discussed, followed by the Oganov fingerprint (a vectorial fingerprint) and the quotient graph of a material, of non-vectorial nature.

3.4.1 Cartesian atomic coordinates as a descriptor

One of the simplest descriptors of the structure of materials and molecules is the positions of the atoms in Cartesian coordinates. A simple way of obtaining a vectorial input is to map all the atomic positions into a unit cell and then concatenate them, so that the descriptor $\mathbf{x}$ becomes:

$$\mathbf{x} = (\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_N)\,, \tag{3.17}$$

where $\mathbf{R}_i$ is the position in Cartesian coordinates of the $i-$th atom and N is the number of atoms in the unit cell.

It is easy to see from the Hamiltonian of the atomic problem (2.10) that the Cartesian coordinates as a descriptor is already complete, i.e. it contains all the information we might need, if the system is not periodic or if the lattice vectors are kept fixed. For periodic systems, it suffices to add the lattice vectors at the end of the array to achieve completeness. It is also non-degenerate, in the sense that it can distinguish physically different configurations.

The main drawback of the Cartesian coordinates as descriptor is the absence of invariance under symmetries: the same physical configuration has infinitely many descriptors associated to all the possible translations and rotations of the reference frame. However, their simplicity, computationally efficiency and general good behaviour make them a frequent choice in some contexts, specially when describing local neighborhoods of the PES.

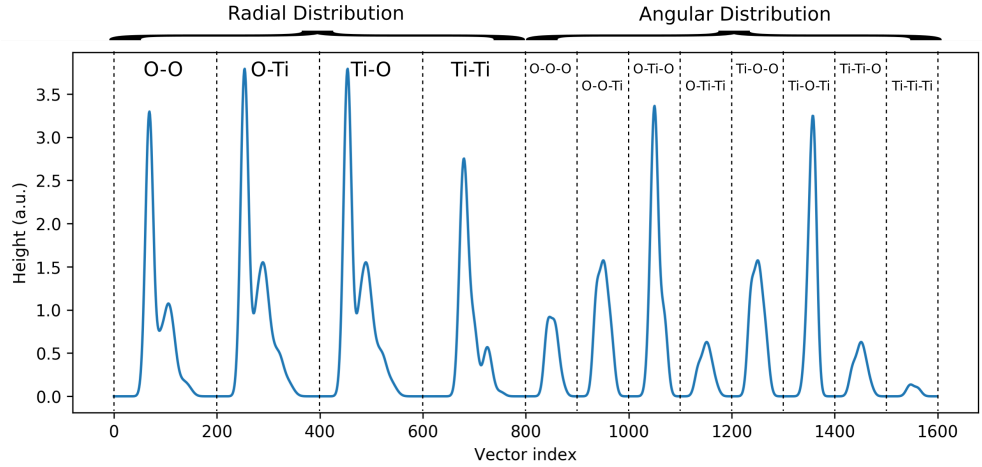


Figure 3.4: The modified Oganov fingerprint for TiO_2. The x axis shows the vector index and the y axis the value of that index in arbitrary units. The labels on the plot indicate if the index corresponds to the radial or the angular distribution and the atomic species involved. The values of the parameters are described in paper V. Figure courtesy of Sami Kaappa.

The Cartesian coordinates have been used in Papers II, III and IV in this thesis. As noted in Paper IV *Machine Learning with bond information for local structure optimizations in surface science*, the use of Cartesian coordinates in combination with an invariant kernel can still result in a machine learning method that enforces certain symmetries, such as the translation invariance.

3.4.2 Modified Oganov fingerprint

In Paper V *Global optimization of atomic structures with gradient-enhanced Gaussian process regression* we have used an extension to the fingerprint developed by Oganov and Valle [72] (from here on, it is referred as *modified Oganov fingerprint*). The fingerprint for a given atomic structure used there can be seen as a concatenation of the radial fingerprints and angular fingerprints.

The radial part of the modified Oganov fingerprint ρ^R_{AB} for atomic species A and B is the vector:

$$\rho^R_{AB}(r) = \sum_{\substack{i\in A\\ j\in B}} \frac{e^{-|r-r_{ij}|^2/2\delta_R^2}}{r_{ij}^2} f_c^R(r_{ij}) \tag{3.18}$$

where i and j run over atoms in the atomic structure of the atomic species A and B, r_{ij} is the distance between atoms i and j, δ_R is a smearing parameter and f_c^R is the radial cutoff function that goes to zero for some finite value of its argument, ensuring the sum is finite. r is a "sample" interatomic distance that acts as an indexing for the vector ρ^R_{AB}.

The angular part of the modified Oganov fingerprint ρ^{α}_{ABC} as a function of the indexing angle θ reads:

$$\rho^{\alpha}_{ABC}(\theta) = \sum_{\substack{i\in A\\ j\in B\\ k\in C}} e^{-|\theta-\theta_{ijk}|^2/2\delta^2_{\alpha}} \; f^{\alpha}_c(r_{ij})\, f^{\alpha}_c(r_{jk}). \tag{3.19}$$

Similarly, i, j and k run over the atoms in species A, B and C, θ_{ijk} is the angle between atoms i, j and k, δ_{α} is the the angular smearing parameter and f^{α}_c is the angular cutoff function.

The two cutoff functions f^{R}_c and f^{α}_c, that avoid introducing into the fingerprint terms where the interatomic distance r_{ij} is large, have been chosen as a piece-wise polynomial:

$$f_c(r) = \begin{cases} 1-(1-\gamma)\left(\frac{r}{R_c}\right)^{\gamma} + \gamma\left(\frac{r}{R_c}\right)^{\gamma+1} & \text{when } r \le R_c \\ 0 & \text{when } r > R_c \end{cases} \tag{3.20}$$

and depend on two different parameters, the cutoff radius R_c and the degree of the polynomial γ, that need to be optimized separately in the radial and angular case to obtain the best representation. We have chosen to use a degree of the polynomial $\gamma = 2$ in the radial part in order to compensate for the r^{-2} decay in the radial fingerprint and $\gamma = 0.5$ in the angular part to ensure a smooth decay.

Combining all together, for an atomic structure with $A, B, \ldots, X$ atomic species, the modified Oganov fingerprint ρ reads:

$$\begin{aligned}\rho = (&\rho^{R}_{AA}, \rho^{R}_{AB}, \ldots, \rho^{R}_{AX}, \rho^{R}_{BA}, \ldots, \rho^{R}_{XX} \\ &\rho^{\alpha}_{AAA}, \rho^{\alpha}_{AAB}, \ldots, \rho^{\alpha}_{AAX}, \rho^{\alpha}_{ABA}, \ldots, \rho^{\alpha}_{XXX}).\end{aligned} \tag{3.21}$$

An illustration of the modified Oganov fingerprint for TiO_2 is shown in Figure 3.4.

Since it only depends on the atomic positions through the interatomic distance and the angles between them, the modified Oganov fingerprint is invariant under translations and under rotations and reflections of the frame of reference (group $SO(3)$).

The modified Oganov fingerprint is also invariant under permutations of the labelling of the atoms of the same species: the functional form of expressions (3.18) and (3.19) as weighted sums of Gaussian functions of the interatomic distances results in this characteristic.

3.4.3 Voronoi tessellation and quotient graphs

It is also possible to use non-vectorial representations of materials and molecules. In fact, it could be argued that a graph structure where the atoms are represented by the nodes of the graph and the edges stand for the bonds between atoms may be a more natural way to encode the topology of an atomic structure. The graph can be used, then, by a message-passing neural network algorithm to predict the properties of the material (see Section 3.2), as it has been shown in recent publications [44, 61, 63, 76, 77].

Graph representations have been extensively used as molecular representations, where the conversion of the structure into nodes and edges results more natural. However,

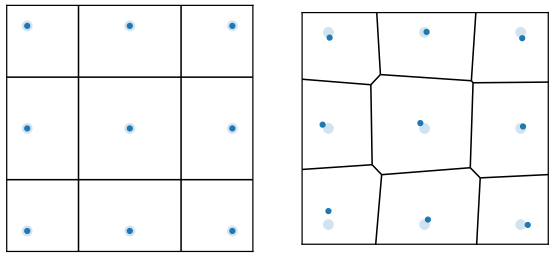


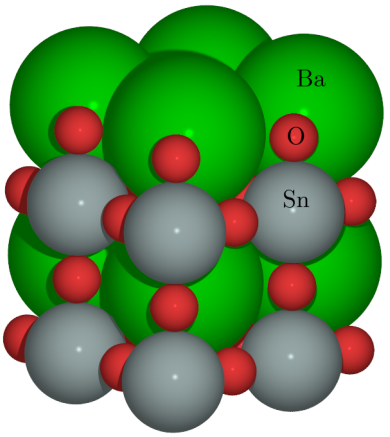
Figure 3.5: Voronoi tessellation of the two dimensional tetragonal lattice with and without noisy inputs. The points are shown in blue and their Voronoi cells are depicted with lines. The Voronoi cells for the tetragonal lattice have squared shape, as shown in the left panel. Adding white noise to the positions, as shown in the right panel, introduces small faces in the resulting Voronoi cells. The original noise-free positions are also shown in the right panel in lighter bigger circles for visual reference.

the question of whether two atoms should be connected or not by bond turns out to be more challenging in the context of defining the representation of materials.

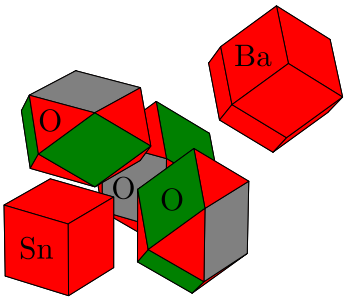
A possibility is to use a *Voronoi tessellation* of the space to define which atoms should be connected to each other in a systematic way. Given a set of reference points (in this case, the atomic positions), the Voronoi tessellation is a partitioning of the space that assigns to each reference point all the points in the space that are closer to it than to any other reference point. When the Euclidean norm is used in the three dimensional space, the Voronoi tessellation defines a polyhedron centered at each atom whose faces result from the intersection between all the bisector planes to the lines connecting the central atom to all its neighbors. We note that the tessels that result from this construction correspond to the Wigner-Seitz cell [78, 79] for structures with a single atom in the unit cell, since the Wigner-Seitz cell is also a Voronoi tesselation. The construction of the Voronoi tessellation for the 2d tetragonal lattice is illustrated in Figure 3.5 and the tessellation for $BaSnO_3$ is illustrated in Figure 3.6(b).

The Voronoi tessellation is then used to define a graph: Each atom is assigned a node and two nodes are connected by an edge if the Voronoi cells of their parent atoms share a face. This representation has the advantage of being invariant under translations and rotations, but however, it has the disadvantage of being very sensitive to noise in the atomic positions [80].

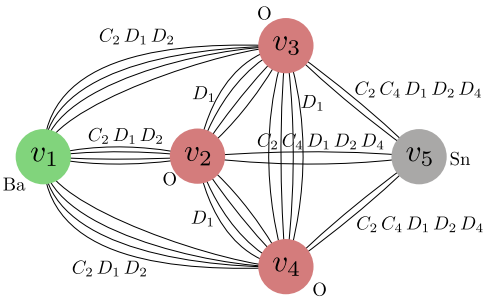
As illustrated in Figure 3.5, small changes in the position of the atoms might result in the creation of small spurious faces. Malins *et al.* [80] have proposed disregarding those connections between two atoms where the two atoms share a face but the line between them does not intersect them, while Isayev *et al.* [81] have proposed to consider only those bonds that are shorter than the sum of the Cordero covalent radii [82] plus a tolerance to alleviate this problem. Another solution, which we propose in Paper I, is



(a) Atomic structure



(b) Voronoi tesellation



(c) Quotient graph

Figure 3.6: Computation of the symmetry-labelled quotient graph from the atomic structure of $BaSnO_3$, in the perovskite structure. Given an atomic structure (a), the Voronoi tessellation using the atomic positions as centers is computed (b). Subfigure (b) shows a depiction of the resulting polyhedra, where additional spacing between faces has been introduced to improve visualization. The atomic symbol indicates the species of the atom in the center of the polyhedron and the colors of the faces indicate the atomic species of the neighbor the face is shared with. The Voronoi tessellation is used to obtain the quotient graph (c): each atom in the atomic structure is assigned a node and the nodes corresponding to two atoms are connected with an edge if they share a face in the Voronoi tessellation. In panel (c) the edges of the graph have been labelled according to the symmetry group of the Voronoi face. Source: Paper I: *Materials property prediction without atomic positions using graph neural networks*.

to only consider the connections defined by “large” faces, those for which the solid angle defined by that face is larger than some threshold. Additionally, we show this is a robust scheme, since the prediction of the heat of formation of the message-passing neural network seems to be robust under small changes of this threshold when it is trained on OQMD [30].

Since the number of atoms in a crystal is infinite, the procedure described above produces infinite graphs, that is, graphs that have infinitely many nodes. It is possible to obtain a finite representation of the graph, known as *quotient graph* [83], by choosing as nodes of the quotient graph the atoms in the irreducible unit cell. Likewise, every connection between a pair of nodes in the infinite graph is then mapped to an edge between the two nodes of the quotient graph that represent the two unit cell atoms the original atoms can be mapped into by a lattice translation. It might happen that two nodes that share an edge in the infinite graph are mapped into the same node in the same quotient graph. The edge then becomes a loop connecting the atom to itself. Thus, the quotient graph for gold in the face centered cubic structure results in a gold node with 12 connections to itself, for example.

The full method for obtaining the quotient graph from the atomic structure followed in Paper I: *Materials property prediction without atomic positions using graph neural networks*, is illustrated in Figure 3.6 for the perovskite structure of $BaSnO_3$.

The advantages of this descriptor, as mentioned above, include the translation and rotation invariance that are inherent to the Voronoi tessellation. Additionally, when used in combination with the message-passing neural network, the method is also invariant under permutations of the atoms of the same species.

More over, this construction is invariant under changes of scale if no additional labels are added. This characteristic makes it possible to use the Voronoi tessellation quotient graphs to represent abstract atomic structures or prototypes by not including information about the interatomic distances.

However, the quotient graph representation that only involves the connectivity and the atomic species has the drawback of being degenerate: physically distinct structures may have the same decriptor. As a counterexample, it is easy to see that the hexagonal close-packed and the face centered structure, both having 12 nearest neighbors collapse into the same representation when the permutation invariance is enforced by the machine learning method. Klee [84] provides further examples of sets of different structures that are described by the same quotient graph and proposes how symmetry may alleviate this drawback. Consequently, machine learning methods based on quotient graph connectivity solely will not be able to reduce the training error further than the energy difference between pairs of structures with the same representation: i.e. it cannot be expected that the error on gold face centered cubic to be smaller than half of the energy difference between the fcc and the hcp structures if hcp structures are also included on the training set.

For the prediction of the heat of formation, the connectivity only quotient graphs are also not complete representations. Since the potential energy of the atoms includes a Coulomb repulsion potential, it is easy to see that a change in the scale of the crystal structure will also change its energy. In this way, the invariance under changes of scale of the representation becomes a blessing and a curse at the same time: it enables to make predictions directly on prototypes of structures, but it must be used with caution, knowing that the machine learning method will never be able to go beyond certain accuracy.

4 Optimization of atomic structures

The minima of the potential energy surface are of great importance to materials science and molecular physics research, as discussed in Chapter 2. Given the complicated nature of the *ab-initio* (from electronic structure calculations) potential energy surface, its optimization usually requires the use of a numerical method.

There are many numerical methods for optimization proposed in literature [12, 85]. It is important to choose an adequate method for the problem at hand that makes the most of the information available and is able to solve it fast. In the optimization of potential energy surfaces, the high computational cost of each energy/force evaluation makes makes it important to reduce the number of such evaluations to the minimum in order to reduce the computational cost of the optimization method.

In this chapter, some of the most common approaches for optimization used on potential energy surfaces are introduced.

4.1 Local optimization

Local optimization methods are numerical methods that do not directly attempt to find the global minimum, but just aim to find an atomic configuration whose potential energy surface is lower than the surroundings [86]. Given that, in general, the potential energy surface is not a convex function of the positions of the atoms[1], a local optimization method will not find the global minimum, but rather a local one.

The most commonly used methods address the problem in a iterative manner: starting from a initial configuration they take a series of steps aiming to reduce the energy compared to the previous steps until they reach convergence to a given configuration. Most implementations of local optimizers on potential energy surfaces, including those in ASE [16, 17], terminate the iteration when all the forces on the atoms fall below a predefined threshold. In other words, the stopping criterion for local optimizers is given by:

$$h \leq \max_{i \in \text{atoms}} |\mathbf{f}_i|, \tag{4.1}$$

where h is the threshold for convergence and the right hand side of the inequality is the maximum of the modulus of the force among the atoms in the unit cell.

The size of the step taken by the optimization method at each iteration is one of the main indicators of its computational cost. Large steps typically may result in a reduction of the number of iterations needed to reach convergence, reducing the computational cost by reducing the number of expensive *ab-initio* calculations. However, too large

[1] It is said that a function f is convex if its domain is convex and for any two points x and y in the domain of f, f satisfies:

$$f(ax + (1-a)y) \leq af(x) - (1-a)f(y) \qquad \text{for all } a \in [0,1].$$

The domain is said to be convex if the straight line connecting any two points in the set also lies in the set [86].

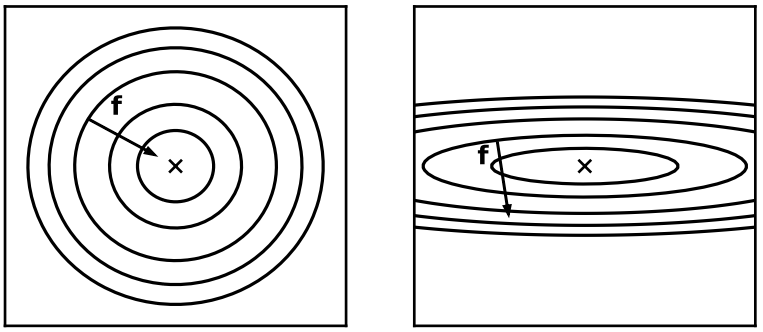


Figure 4.1: Illustration of the direction of the force for potential energy surfaces that are isotropic (left panel) or anisotropic (right panel) around the minimum (marked with a cross). The curves show the level sets. When the curvature around the minimum is similar in all directions, the force points towards the best direction for a local optimization step. However, in poorly scaled problems where the curvature is very different in two directions, the direction of the force might not lead to the minimum.

steps may reduce the robustness of the method, since the algorithm might get confused and not perform well depending on the details of the potential energy surface. In the following, the strategies to control the computational cost used by the most popular local optimization algorithms are reviewed.

4.1.1 The gradient

The use of a gradient based optimization method is very advisable for atomic structure optimization problems. By virtue of the Hellmann-Feynman theorem (2.13), the gradients of the potential energy surface come at little additional computational overhead when the energy values are computed. The use of the gradient information in the local optimization usually reduces the number of steps needed to find the potential energy minimum, reducing the overall computational cost.

The gradient provides with the direction along which the energy is decreasing the fastest, and thus it would seem like the most obvious choice for the direction of the step. However, it is worth noting that under many circumstances, just following the direction of the forces to find the minimum (known as gradient descent method [86]) leads to many optimization steps.

In some systems, the curvature of the potential energy surface in some direction is larger than the curvature in a different direction by several orders of magnitude. This can happen, for example, when in a system presents bonds with very different degrees of stiffness, such as a molecule loosely bonded to a metallic surface. This situation is known as a *poorly scaled* optimization problem, and it can cause difficulties to converge to certain algorithms, such as to gradient descent. In poorly scaled problems, the forces do not point, in general, towards the minimum: as a consequence, gradient descent trajectories inefficiently zigzag their way towards the minimum. An illustration of this phenomenon can be found in figure 4.1.

The problem of the scaling in atomic structures with different types of bonds is central to the local optimization of potential energy surfaces. In the following sections, some of the most common methods addressing this question are presented.

Before moving forward, let us review some properties of the algorithms and the solutions they find.

A point $\mathbf{x}^*$ is said to be a *stationary point* of the objective function $E(\mathbf{x})$ if it has zero force $\mathbf{f}^* = 0$. We note that such points are called stationary points since the algorithm will stop its iteration and consider that convergence to a solution has been reached. It is important to note that, even though all local minima of a smooth potential energy surface are stationary points, the converse is not true, since saddle points and maximum points also fall in this category. Local optimizers usually converge to a local minimum or saddle point in the same basin as the initial configuration.

A related and important concept is that of *global convergence*: A local optimization iterative method is said to be globally convergent if it generates a sequence of points that converges to a stationary point from any given initial point, [86], this is, the sequence of forces fulfills:

$$\lim_{k\to\infty} \|\mathbf{f}_k\| = 0. \tag{4.2}$$

This is a robustness requirement for any optimization algorithm: if a method is globally convergent, it will find a solution starting from any initial structure.

However, global convergence is not the only important requirement for a good optimization method, since a method might be guaranteed to find the solution but might also require a large number of DFT evaluations to find it. For this reason, it is important to study the reduction of the distance to the minimum or the reduction of the magnitude of the gradient with each step when the starting point is in the neighborhood of the minimum. This property is called *convergence rate* (also referred to as local convergence). A usual form to express the convergence rate of an iterative algorithm to a stationary point $\mathbf{x}^*$ is by showing that the method reduces the error in each step, leading to an expression in the following form:

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq L\|\mathbf{x}_k - \mathbf{x}^*\|^p, \tag{4.3}$$

where L is a positive constant and p is the convergence rate. If $p = 1$, the method is said to have linear convergence, if it is $p = 2$, the method has quadratic convergence.

4.1.2 Newton and Quasi Newton Methods

A way to correct for the difference in curvature along different directions is to use information about the second order derivatives of the potential energy surface. It is possible to use the second order Taylor approximation around the current configuration $\mathbf{x}_k$:

$$E(\mathbf{x}_k + \mathbf{p}) \approx E_k - \mathbf{p}^T\mathbf{f}_k + \frac{1}{2}\mathbf{p}^T H_k \mathbf{p}, \tag{4.4}$$

to find the step with the optimal direction $\mathbf{p}$ within this model:

$$\mathbf{p} = H_k^{-1}\mathbf{f}_k. \tag{4.5}$$

Here, E_k, $\mathbf{f}_k$ and H_k are the energy, the force and the Hessian matrix at $\mathbf{x}_k$. A local optimizer using a second order model like (4.4) of the potential energy surface to guide the search for the minimum is called a *Newton method*, and the direction defined in expression (4.5) is called the *Newton direction*.

It is easy to see how the Newton method solves the scaling problem, specially, in the neighborhood of a minimum. Moreover, Newton methods are often globally convergent (provided that the Hessian matrix is not ill-conditioned) and usually exhibit quadratic local convergence rate towards the minimum [86]. However, Newton methods have the drawback that they require the knowledge of the Hessian matrix at every step. Since the Hessian of a potential energy surface must be computed using finite difference methods (as explained in Section 2.2.1), the computational cost of Newton methods makes them unpractical for most atomic systems.

A possibility is to substitute the Hessian matrix H_k in equations (4.5) and (4.4) for an approximation to it B_k. A common practice is to start with an initial estimation of the Hessian (by using finite differences [12, 87] or by choosing an initial estimation of the Hessian matrix that is proportional to the identity, $B_0 \propto \mathbb{I}$) and then to update this initial estimation along the way with the force and the energy information:

$$B_{k+1} = B_k + f(B_k, \Delta\mathbf{x}_{k+1}, \Delta\mathbf{f}_{k+1}) \tag{4.6}$$

where f is a function of the previous estimation of the Hessian matrix B_k and the difference between the atomic coordinates $\Delta\mathbf{x}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_k$ and the forces $\Delta\mathbf{f}_{k+1} = \mathbf{f}_{k+1} - \mathbf{f}_k$ of the previous step. Local optimization methods that proceed in this way are called *quasi-Newton* methods, and they have been extensively used in potential energy surface optimization [12, 13]. Probably, the most used expression of the form of equation (4.6) is the Broyden, Fletcher, Goldfarb and Shano (BFGS) formula (4.8), that will be discussed later, but it is noteworthy that several of these expressions have been proposed in literature [86, 88, 89].

Once one has a local model such as the one presented in equation (4.4), there are two families of strategies to build an iterative method that leads to the minimum: *line search* methods and *trust region* methods [86].

As noted above, Newton and quasi-Newton methods provide with a preferential optimal direction, (4.5) (if the Hessian is positive definite, as we will discuss below). The *line search* method for local optimization then proposes a step along this direction,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k, \tag{4.7}$$

where α_k is known as the step length. Once fixed the direction, the step length α_k has to be determined so that the step results effective: the algorithm minimizes the function along the line of choice, hence the name.

In order to reduce the number of DFT calculations needed in the line search, methods often take a new step if the new point $\mathbf{x}_{k+1}$ is suitable instead of aiming for the exact line minimization [90]. Starting with $\alpha_k = 1$ (which leads to the minimum of the Taylor expansion), approximate line search methods accept or reject a step size α_k

if the resulting point fulfills two conditions: a *sufficient decrease* condition, ensuring that the step was long enough to produce enough decrease of the energy and a *curvature condition*, that requires sufficient change of the projection of the force along the direction of search, preventing very small steps [86]. If the point does not satisfy one of these conditions, a new point is generated until a point fulfilling both conditions is found. These conditions ensure the global convergence of a Newton or quasi-Newton algorithm together with a reasonable convergence rate.

The other approach is the one used by *trust region* methods. The minimization algorithm fixes the maximum step size δ_k the method is allowed to take and then finds the point $\mathbf{x}_{k+1}$ that minimizes the model inside a ball of radius δ_k. The size of the region where the model can be trusted is then increased or decreased depending on the predicting performance of the method in the previous step: If the prediction of the model was very close to the value of the DFT energy, the trust region radius δ_k can be increased in the following iteration (since steps that are both "good and large" lead to a reduction of the DFT steps needed to minimize the function), meanwhile if the prediction differs too much from the DFT calculation, the point is rejected and the trust radius shrinked. For an intermediate difference between the prediction of the model and the DFT calculation (i.e., the value is within a tolerance), the point is accepted and the trust radius left unchanged [90].

This step size control method is then coupled with a trust region sub-problem solver, which is then in charge of minimizing the quadratic Newton or quasi-Newton model within the constrained region. Even though methods for finding sufficiently decreasing approximations to the solution of the trust region subproblem exist [86, 91], they are not necessary in the context of potential energy surface optimization, since minimizing a quadratic function of the form (4.4) is much more computationally efficient than a taking a single DFT calculation to reevaluate the energy and forces, and will appear negligible in comparison.

Probably, the most widely used method for local optimization of atomic structures is BFGS line search. As explained above, the BFGS formula is an expression to compute updates to the approximation of the Hessian matrix in the context of quasi-Newton methods:

$$B_{k+1} = B_k - \frac{B_k\,\Delta\mathbf{x}_{k+1}\Delta\mathbf{x}_{k+1}^T\,B_k}{\Delta\mathbf{x}_{k+1}^T\,B_k\,\Delta\mathbf{x}_{k+1}} - \frac{\Delta\mathbf{f}_{k+1}\,\Delta\mathbf{f}_{k+1}^T}{\Delta\mathbf{f}_{k+1}^T\,\Delta\mathbf{x}_{k+1}}. \tag{4.8}$$

The BFGS formula, which is a rank 2 update to the previous estimation of the Hessian matrix, has the property that if the initial estimate of the Hessian is positive definite (as, for example, $B_0 \propto \mathbb{I}$), the BFGS update produces a sequence of positive definite matrices B_k, provided that $\Delta\mathbf{x}_k \cdot \Delta\mathbf{f}_k < 0$. This inequality is always fulfilled when the BFGS formula is combined with a line search strategy of the kind described above [86]. This is important, since the direction $B_k^{-1}\mathbf{f}_k$ is only a descent direction, in general, if B_k is positive definite. Other Newton and quasi-Newton implementations need special corrections to deal with non positive definite Hessians, but the BFGS line search method solves this problem in a natural way.

Summarizing, the BFGS line search method proceeds in the following way: Starting with a step parallel to the initial gradient, at every step it produces a convex quadratic model and it chooses the direction towards the minimum of the model as the direction of the next step. The method then evaluates the configuration that minimizes the quadratic model ($\alpha_k = 1$ in equation (4.7)) and performs a DFT calculation for it. If the energy of the this configuration shows a sufficient decrease and the change in the forces is sufficient to fulfill the curvature condition, the step is accepted and the quadratic model is updated using the BFGS formula (4.8) (which is rank 2, so it updates "two directions" per step). If the new energy and force do not fulfill these conditions, as it will happen if the model and the true potential energy surface are very different, the point is rejected and the step length is changed, resulting in a new DFT calculation along the same direction as the original one. This process is iterated until the energy and the force of a new point fulfil the conditions for the line search, and then a new convex quadratic model is obtained from the information of the last point.

In regions where the potential energy surface does not have a positive definite Hessian, the convex quadratic model that BFGS line search builds is doomed to fail. Surprisingly, BFGS has been shown both analytically and experimentally to have very good self-correcting properties [86, 92]: when the approximation of the Hessian B_k differs substantially from the true Hessian $H(\mathbf{x}_k)$, the line search needs more evaluations to converge, but after a few iterations the BFGS approximation to the Hessian tends to correct itself. As a result, BFGS line search also tends to be robust to numerical noise.

Good global convergence properties have been experimentally observed for methods of the BFGS line search class, and the global convergence for a general class of non-convex functions has been mathematically proved for a specific set of line search conditions recently [93, 94]. In addition, BFGS line search also exhibits good local convergence properties, with a superlinear convergence rate in the vicinity of a stationary point [86].

Storing the approximation to the Hessian can be computationally expensive for systems with a large number of atoms. An alternative is to use the light memory BFGS algorithm (L-BFGS). In this method, the list of the last n configurations **x** and forces **f** are used to compute the product $\mathbf{B}_k^{-1}\mathbf{f}_k$ at each step without computing the matrix B_k explicitly.

4.1.3 Other approaches for local optimization of atomic geometries

Local optimization literature has abundant examples of optimization methods other than the Quasi-Newton family. One of the most common methods outside the quasi-Newton class is the *conjugate gradient* method [65, 86]. In the conjugate gradient method, the line search for the minimum takes place along directions that are conjugate to each other, in the sense that $\mathbf{p}_i^T H \mathbf{p}_j = 0$ for $i \neq j$. These directions are not only linearly independent, but can be shown to be the optimal subsequent line search directions to minimize a quadratic function [86]. Since this method only needs to store the previous search direction, energy evaluation and force, as opposed to storing an estimation of the Hessian in the Quasi-Newton methods, makes this method suitable for larger systems.

A number of minimization algorithms based on molecular dynamics is also available [95–97]. The molecular dynamics minimizer proposed by Jónsson *et al.* [95] performs a molecular dynamics calculation with all atomic masses equal to one and with "restarts": whenever the dot product of the force and the velocity is negative, the minimum has been "overshot" and the velocity is set to zero. Bitzek *et al.* have proposed an improvement of such method, called *fast internal relaxation engine* (FIRE) by introducing a velocity-dependent friction term in the velocity update in the molecular dynamics step, while maintaining the velocity restarts when the motion turns out to be uphill. This method has been shown to be faster than conjugate gradient while having a low computational cost, and its performance is competitive with limited memory BFGS line search methods[97].

Finally, a way to help with the ill conditioned nature of the optimization of some atomic geometries is to introduce a *preconditioner*. A preconditioner of a local optimization method C is a transformation of the input space of the form $\mathbf{x}_{new} = C(\mathbf{x}_{old})\,\mathbf{x}_{old}$ such that the eigenvalue distribution of the eigenvalues of the Hessian in the transformed space is more favorable to the optimization method than the original one [86]. Along these lines, Packwood *et al.* introduced in 2016 a preconditioner based on the laplacian matrix of a weighted graph derived from the atomic structure and showed that it lead to a significant reduction of the number of DFT evaluations needed to find a minimum when combined with a quasi-Newton line search method [98]. During our investigations, we have found that this method, as implemented in ASE [16] is not globally convergent, since the curvature condition is not enforced in the line search and this allows the method to take very small steps without converging to a minimum for a small percentage of the systems studied. More recently, Mones *et al.* [99] have introduced a preconditioner based on the Hessian of a faster but less accurate interatomic potential and shown that this leads to a further reduction of the DFT evaluations needed to find the minimum energy configuration.

4.2 Global optimization

Finding a local minimum of the PES, which is in general a non-convex function, often provides with little information about the global minimum. Except for the cases in which the search can be guided by physical intuition, in order to find the true global minimum one needs to use a set of techniques that are very different to those introduced in the previous section.

A possibility would be to explore all the relevant configuration space in a deterministic manner. However, it is easy to see that this approach will be limited to systems with just a few atoms. The curse of dimensionality (the dimensionality of the PES is $3N-6$ and the number of possible configurations grows combinatorially with the number of atoms [100, 101]) together with the computational cost of quantum chemistry methods often make this approach impractical [102].

In contrast, meta-heuristic approaches to the global optimization problem [90, 103] have been quite successful at overcoming the dimensionality problem [85]. Some examples of such methods include random searching [14], evolutionary algorithms [100, 104–106], swarm particle optimization [107] and basin hopping methods [108],

which have been shown to be successful methods for global optimization of atomic structures.

This kind of methods include a random component governed by a deterministic strategy, in an attempt to map the general structure of the PES with as few as possible energy-force calculations. Thus, they combine an *exploration* strategy aimed to explore as many local minima as possible with an elitist or *exploitative* strategy, aimed to use the information from the already explored areas to determine which areas are worth exploring more. Since there is no guarantee that the global minimum has been found unless the full PES has been mapped, which is often impractical, it is important to find a balance between exploration and elitism in order to achieve the convergence of the method within a reasonable computational expense [109].

Recently, the use of machine learning methods in the identification of the global minimum has become a more widespread technique [85, 101]. As explained in Chapter 3, it is possible to build machine learning models that accurately reproduce the potential energy surface and to use them to make predictions at a fraction of the cost of a quantum chemistry method. The following section introduces *Bayesian optimization*, a technique that benefits from a Bayesian model to guide the heuristics of the global search.

4.2.1 Bayesian optimization

Bayesian optimization is an global optimization method that uses a surrogate probabilistic model to guide the exploration of the target function. It has two main components: a *Bayesian model* of the target function and a policy to decide which point to sample next, called *acquisition function* [110, 111].

Bayesian optimization uses Bayesian probabilistic models: the models include a prior probability distribution over possible functional forms of the target function and the data that is acquired during the optimization is used to update the posterior probability. A common choice in the context of potential energy surface optimization is to use Gaussian process regression (see Section 3.3) as a probability distribution model. The Gaussian process framework provides with a prediction and an uncertainty measure in a natural way, since it models the distribution over target functions at every point $\mathbf{x}$ as a Gaussian probability distribution with average $\mu(\mathbf{x})$ and variance $\sigma^2(\mathbf{x})$.

The acquisition function is a function of $\mathbf{x}$ that estimates the utility of the point to the global search: it is the feature that balances exploration and exploitation in Bayesian optimization. The optimization policy usually then proceeds by choosing the configuration $\mathbf{x}$ at each stage that optimizes the acquisition function, and adds it to the training set of the Bayesian model.

For example, in Paper V, we have used the *lower confidence bound* (LCB) acquisition function in combination with Gaussian process regression:

$$\alpha_{\text{LCB}}(\mathbf{x}) = \mu(\mathbf{x}) - \kappa\sigma(\mathbf{x}), \tag{4.9}$$

where κ is the coefficient that dials the explorative character of the method. It is easy to see that the values of $\mathbf{x}$ that minimize α_{LCB} have either low predicted energy (exploitation) or large uncertainty (exploration), and thus, are the configurations a global

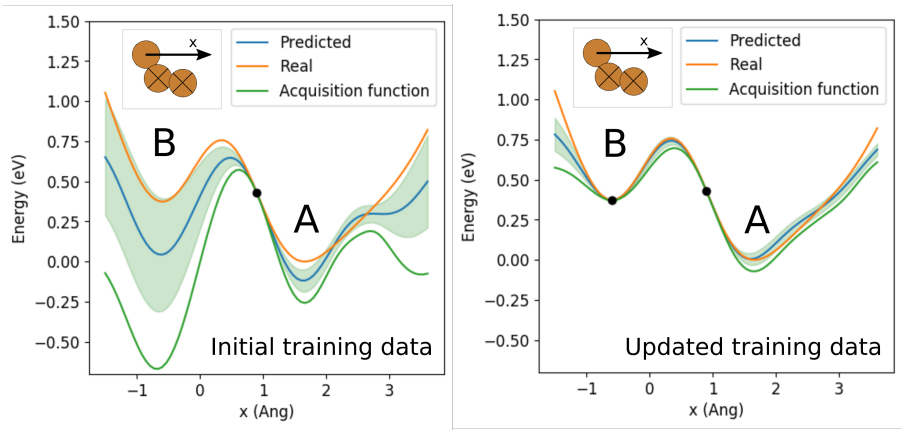


Figure 4.2: Example of Bayesian optimization for a global optimization problem. The target function in this problem, represented in orange, is potential energy surface of a copper atom that slides over the surface created by two static copper atoms. The blue curve shows the Gaussian process regression model that has been fitted to the training points, represented in black. The left panel shows the model and the acquisition function when there is only one point sampled, belonging to the basin of the minimum **A**. The acquisition function, represented in green, predicts the global minimum to be in the basin **B**, since it has a higher uncertainty. The right panel shows the result of probing the minimum of the model with the lowest acquisition function, and updating the model and the acquisition function accordingly. Given the new information, minimum **B** is identified as a local minimum, and the new acquisition function now suggest probing the location of the global minimum **A**. Source: Paper V, *Global optimization of atomic structures with gradient-enhanced Gaussian process regression*

optimization algorithm is interested in sampling. Jørgensen *et al.* have reported that the value $\kappa = 2$ provides a good balance between exploration and exploitation in the context of global optimization of atomic structures [109].

Figure 4.2 illustrates the development of a global optimization run. Basin B has a higher true energy than basin A, but with the information provided with the first point, it also has a larger uncertainty and thus, it minimizes the acquisition function. The global optimization method then samples one structure at a time, the one that minimizes the acquisition function. The minimum of basin B is then sampled and its energy and force are Incorporated to the training set (see Section 5.1 for more details). The probability model is then updated and the new acquisition function minimum matches the global minimum of the potential energy surface, solving the global optimization problem.

It is important to carefully chose the prior to the Bayesian probability model in order to ensure efficient convergence properties. For example, in Figure 4.2 the posterior probability with just one training point is already able to correctly identify the position of the basins of the different local minima thanks to a good choice for the kernel (see Section 5.2 for a discussion on kernel choices). This reduces the amount of structures

needed to sample to ensure the correct exploration of the PES.

Due to their data efficiency, Bayesian optimization methods have shown successful results in many fields where the sampling of the target function is expensive [110]. In particular, there are several examples of successful applications of Bayesian optimization to atomic structure problems [112–117].

4.3 Transition state optimization

Identifying transition structures is, in general, more challenging than identifying local minima. Algorithms for the identification of first order saddle points (see Section 2.2) must maximize the function in one direction and minimize it in every other direction, making many of the strategies presented in section 4.1 no longer useful. In this section, a summary of some of the most relevant methods for this thesis is presented.

Generally speaking, transition structure optimization methods can be divided into two classes: *single ended methods* and *double ended methods* [12]. The former try to converge to the transition state iteratively, typically starting at the neighborhood of the product or the reactant of the chemical reaction, while the later require the knowledge of the product and the reactant atomic structures and try to find the transition structure by estimating the minimum energy path.

Single ended methods must climb the potential in one direction while they minimize in all other directions. If the starting structure is in the basin of a minimum (starting from the reactants or the products), the Hessian matrix of the potential energy surface is positive definite and the gradient points towards the minimum, so the usual gradient descent or Newton dynamics would not take the algorithm towards the saddle point.

A common strategy for single ended methods to address this problem is to follow the smallest eigenvalue of the Hessian. However, the bare computation of the eigenvalues of the Hessian might be very computationally demanding, since there is no computationally efficient strategy to compute the Hessian from electronic structure calculations analogous to the Hellmann-Feynman theorem 2.13.

There is a number of numerical methods to estimate the softest mode without directly computing the Hessian Matrix [118–121]. In Paper III, we have used the *dimer method* [122] to identify transition states. In this method, two atomic configurations, or images, are considered at each step, hence the name. At each step, the dimer is rotated keeping its center fixed to align the vector between the images with the minimum eigenvalue of the Hessian, by minimizing the sum of the energies of the two images. The center of the dimer is then pulled in the uphill direction, by reflecting the resulting force on the center of the dimer about the axis perpendicular to the dimer, to converge to the saddle point.

Double ended methods aim to obtain the minimum energy path, by representing it with a set of discrete images [12]. The problem of the discrete representation of the path comes from the fact that if all images were allowed to move in the directions of their forces, they would collapse into each other. Thus, double ended methods need

to constrain the movement of the images along the direction of the path to ensure its correct representation.

The *nudged elastic band* (NEB) [95] solves this problem by adding a spring potential between the images. The images are moved along the direction perpendicular to the path using their true forces to update their position and the update uses the spring constant in the direction parallel to the path. In this way, the method is able to converge the initial string of images to represent the minimum energy path. A version of this method, called *climbing image nudge elastic band* was proposed shortly after the initial publication to force one of the images, known as climbing image, to converge to the saddle point while retaining the convergence of the full band to the minimum energy path. When the regular NEB is close to converging, the image with the highest energy climbs up along the band while minimizing its energy in the direction perpendicular to the band [123]. The rest of the images retain the behaviour of the regular NEB, converging to the minimum energy path.

Recently, a number machine learning methods have been proposed to speed up nudged elastic band calculations [124–129]. In Paper III (see Section 6.3) we discuss how to combine these methods with local optimization ones to reduce the computational overhead of the computation of reaction networks.

5 Gaussian process models of potential energy surfaces

In Chapter 3 the notion of Gaussian process regression has been introduced and how to obtain the hyperparameters has been explained. However, the details on the *a priori* choices of the model that lead to a useful model of the potential energy surface has not been discussed.

In this chapter, the main choices that lead to useful models are discussed. After introducing the formalism to include the information of both the energy and the force of each point is introduced, the different choices for kernels and priors used in this thesis are discussed. The discussion also includes several tricks to choose the hyperparameters or to obtain them in an efficient way, as well as the main challenges regarding numerical stability.

5.1 The forces: Gaussian process regression including gradient information

It is possible to extend the Gaussian process formalism introduced in Chapter 3 to include information about the forces. By noting that a linear transformation of a kernel is also a kernel itself, a method to model a function and its gradients can be modelled as a multivariate Gaussian process [64, 130].

Let ϱ denote the descriptor used to represent an atomic structure with N atoms, and let $\mathbf{R}_i$ be the position of the $i-$th atom. Let E be its energy and $\mathbf{f}_i$ the force on the $i-$th atom. Thus, it is possible to write the gradient of the energy with respect to the Cartesian coordinates $\mathbf{x} = (\mathbf{R}_1, , \mathbf{R}_2, \ldots, \mathbf{R}_N)$ as the negative concatenation of the forces over individual atoms $-\mathbf{f} = -(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N)$.

The prior probability over functions of ϱ becomes now a multivariate distribution for the joint energies and forces:

$$(E, -\mathbf{f}) \sim \mathcal{N}\left(\mathbf{m}(\varrho), K(\varrho, \varrho')\right), \tag{5.1}$$

where $\mathbf{m}(\varrho)$ becomes now a $3N+1$ dimensional vector field where its first entry models the prior expectation on the potential energy and the remaining entries stand for the prior expectation on the forces and $K(\varrho, \varrho')$ is the prior covariance, which in this framework becomes a $(3N+1) \times (3N+1)$ matrix.

It is possible to obtain the kernel matrix function of the multivariate problem $K(\varrho, \varrho')$ by extending the correlation model for the energies. Let us denote the energy-energy correlation by the kernel function $k(\varrho, \varrho')$:

$$\langle E(\varrho), E(\varrho') \rangle = k(\varrho, \varrho'). \tag{5.2}$$

Then, the energy-force and the force-force correlations can be obtained by differentiating [64, 130, 131]:

$$\langle (E(\varrho), -\mathbf{f}_i(\varrho')\rangle = \nabla_{\mathbf{R}_i'} k(\varrho, \varrho') \tag{5.3}$$

and

$$\langle \mathbf{f}_i(\varrho), \mathbf{f}_j(\varrho')\rangle = \nabla_{\mathbf{R}_i} \left(\nabla_{\mathbf{R}_j'}(k(\varrho, \varrho')\right)^T, \tag{5.4}$$

where $\nabla_{\mathbf{R}_i}$ represents the differentiation operator with respect to the $i-$th atomic Cartesian coordinate $\mathbf{R}_i$. The full kernel including derivative information can be obtained by arranging equations (5.2), (5.3) and (5.4) in matrix form [130]:

$$K(\varrho, \varrho') = \begin{pmatrix} k(\varrho, \varrho') & \left(\nabla_{\mathbf{x}'} k(\varrho, \varrho')\right)^T \\ \nabla_{\mathbf{x}} k(\varrho, \varrho') & \nabla_{\mathbf{x}} \left(\nabla_{\mathbf{x}'} k(\varrho, \varrho')\right)^T \end{pmatrix}. \tag{5.5}$$

The rest of the procedure is an extension to multivariate distributions of the method exposed in Section 3.3. Let $\boldsymbol{\rho} = (\varrho_1, \varrho_2, \ldots, \varrho_n)$ be the design matrix. As usual, let $K(\varrho, \boldsymbol{\rho}) = (K(\varrho, \varrho_1), K(\varrho, \varrho_2), \ldots, K(\varrho, \varrho_n))$ and let $K(\boldsymbol{\rho}, \boldsymbol{\rho})$ represent the Gram matrix, whose elements are given by $(K(\boldsymbol{\rho}, \boldsymbol{\rho}))_{ij} = K(\varrho_i, \varrho_j)$. The energies and forces of the configurations of the training set are also arranged in an extended target vector $\mathbf{Y}$ of the form $\mathbf{Y} = (E_1, -\mathbf{f}_1, E_2, -\mathbf{f}_2, \ldots, E_n, -\mathbf{f}_n)$. In this framework, the regularization becomes a matrix Σ_n:

$$(\Sigma_n)_{ij} = \begin{cases} \sigma_n^E, & \text{if } i = j \text{ and } i \bmod (3N+1) = 0 \\ \sigma_n^{\mathbf{f}}, & \text{if } i = j \text{ and } i \bmod (3N+1) \neq 0 \\ 0, & \text{if } i \neq j, \end{cases} \tag{5.6}$$

where σ_n^E is the regularization on the energies and $\sigma_n^{\mathbf{f}}$ is the regularization on the forces.

The equations on the average prediction and the uncertainty at a trial configuration ϱ look as usual:

$$(E(\varrho), -\mathbf{f}(\varrho)) = \mathbf{m}(\varrho) + K(\varrho, \boldsymbol{\rho})\, \mathbb{C}^{-1} \left(Y - \mathbf{m}(\boldsymbol{\rho})\right) \tag{5.7}$$

$$\mathbb{V}(\varrho) = K(\varrho, \varrho) - K(\varrho, \boldsymbol{\rho})\, \mathbb{C}^{-1} K(\boldsymbol{\rho}, \varrho), \tag{5.8}$$

where $\mathbf{m}(\boldsymbol{\rho}) = (m(\varrho_1), m(\varrho_2), \ldots, m(\varrho_N))$, $\mathbb{C} = K(\boldsymbol{\rho}, \boldsymbol{\rho}) + \Sigma_n^2$ is the regularized matrix and $\mathbb{V}(\varrho)$ is the variance of the Gaussian process at configuration ϱ. $(\mathbb{V}_{11})^{1/2}$ is the uncertainty on the energy and the square root of the remaining $3N$ diagonal entries are the uncertainties on the forces.

An example of the use of derivatives in Gaussian process regression for a one dimensional problem can be found in Figure 5.1. This figure makes evident the advantages of training also to the derivatives in Gaussian process regression: Not only the resulting model has the right values at the training points, but it is also forced to have the right derivatives. As a result, with the same amount of training configurations, the error of interpolation reduces significantly. The uncertainty, interpreted as a lack of

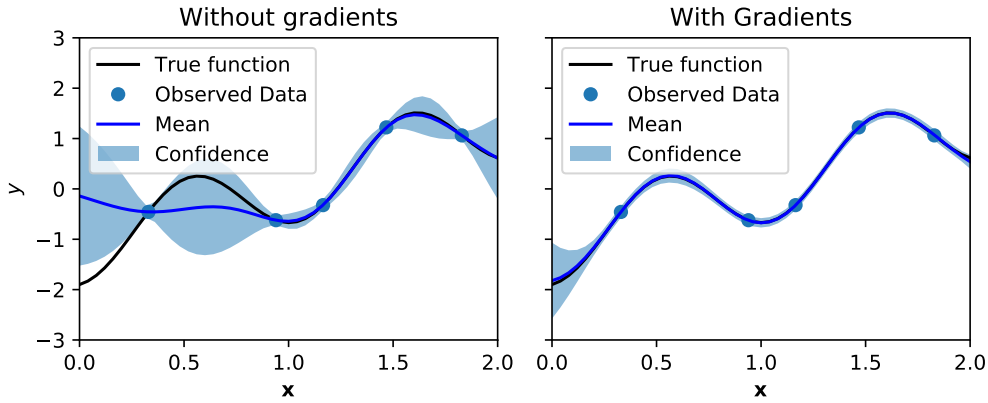


Figure 5.1: Gaussian process regression with and without gradient information. The target function is plotted in black, while the model (the average and the uncertainty) are plotted in blue. The models here use the squared exponential function as a kernel (see Section 5.2 for details). The hyperparameters of each model have been obtained by maximizing the marginal likelihood with noise value $\sigma_n = 10^{-3}$.

information in a sampled point that does not belong to the training set, also reduces significantly around the training points.

In higher dimensions of the input space, the improvements are even more interesting: by training also to the forces, the model trains to "$3N + 1$ pieces of information" as compared to a single quantity when it is only trained to the energies, making the predictions of the model converge to the target manifold faster.

A recent publication by Christensen and von Lilienfeld has confirmed that the aforementioned characteristics result in an increase of the accuracy of Gaussian process regression models for potential energy surfaces of molecules when the forces are included in the training set [132]. As the authors note, the Hellmann-Feynman theorem (see Section 2.2.1) provides numerical forces at a reduced additional computational cost for most quantum chemistry codes, and hence this information is already available in most training sets. For these reasons, Gaussian process regression with forces has recently become a preferred option for the modelling of potential energy surfaces [125–127, 131–135]. It is possible to extend the formalism to include higher derivatives, as Denzel and Kästner have recently shown [129], but the computational overhead of computing higher derivatives numerically makes this technique only suitable for certain applications.

Papers II to V in this thesis develop and use different methods for the navigation of potential energy surfaces based on Gaussian process regression models that are trained both on energies and forces. For this reason, in the following it is assumed that the forces are included in the training set, unless otherwise stated.

5.2 The kernel: Correlation models and prior information

In Chapter 3 the notion of the kernel in a Gaussian process was introduced, but no examples were presented nor a methodology to pick one was introduced.

The kernel of the Gaussian process represents the *a priori* correlation between points of the potential energy surface. For this reason, choosing a good prior model can result in a better posterior model, which has a reduced error with a given number of points, or that requires less points (and thus, less computational effort) to achieve the same accuracy. In the following, some of the common choices of kernel are discussed and compared.

5.2.1 Stationary kernels

A common choice of the kernel function is to choose a stationary kernel, that is, a kernel that only depends on the distance between the input configurations. Stationary kernels have the property of being invariant under translations of the frame of reference of their input space, since they only depend on the distance between the configurations [136].

There are many possible definitions of distance, but in the following discussion we will restrict ourselves to consider stationary kernels those that are a function of the distance in Cartesian coordinates $|\mathbf{x} - \mathbf{x}'|$, since it is common to represent and visualize potential energy surfaces in this coordinate set and they are naturally involved in the computation of the force. Other coordinate sets can be found in the next section. Thus, the predictions of the kernels presented in this section are independent of the origin chosen for the Cartesian coordinates, which is a desirable property to have when working in this coordinate set.

Probably, one of the most common choices of a correlation model is the *squared exponential kernel*, also known as radial basis function (RBF) kernel:

$$k(\mathbf{x}, \mathbf{x}') = k_0^2 e^{-\|\mathbf{x}-\mathbf{x}'\|^2/2\ell^2}, \tag{5.9}$$

which depends on two hyperparameters: the prefactor k_0 and the scale ℓ. A Gaussian process with squared exponential kernel is smooth, i.e. all its derivatives exist and are continuous, in the mean squared sense (see references [64, 136]). As a consequence, the average prediction from a Gaussian process with squared exponential kernel are very smooth functions, and this characteristic makes the kernel a popular choice for the modeling of potential energy surfaces, where it has lead to good results [125, 127, 133]. This is, in fact, the kernel adopted in Papers II, *Local Bayesian optimizer for atomic structures* and III, *An active-learning approach to combine surrogate machine learning algorithms for probing potential energy surfaces*.

The squared exponential kernel can be regarded as a limit of the Matérn class of kernels, which encode different degrees of smoothness in the prior class of functions. The *Matérn kernel* of degree ν is given by the following expression:

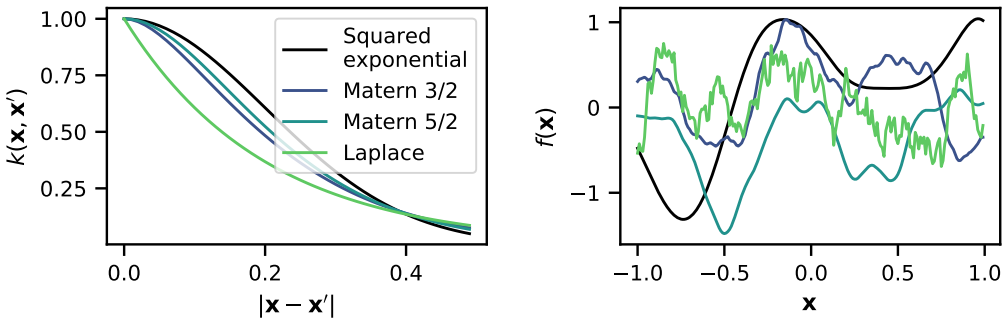


Figure 5.2: Common stationary kernels and the smoothness of the models they produce. The left panel shows the kernel as a function of the distance between the kernel inputs, and the right panel shows an example of function such kernel generates. All curves have been obtained with scale parameter $\ell = 0.4$ and prefactor $k_0 = 0.5$

$$k_\nu(\mathbf{x}, \mathbf{x}') = k_0^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)^\nu K_\nu \left(\sqrt{2\nu} \|\mathbf{x} - \mathbf{x}'\| / \ell \right) \tag{5.10}$$

where $\Gamma()\dot{}$ is the gamma function and K_ν is the modified Bessel function. The hyperparameters k_0 and ℓ are again called prefactor and scale. The hyperparameter ν encodes the differentiability of the sample functions of a Gaussian process using this kernel [110]: a process with a Matérn kernel of order ν is $nu - 1/2$ times mean square differentiable. A particular case of the Matérn family is the *Laplacian kernel*:

$$k(\mathbf{x}, \mathbf{x}') = k_0^2 e^{-\|\mathbf{x} - \mathbf{x}'\| / \ell} \tag{5.11}$$

which is not differentiable.

The sample functions of the squared exponential and Laplacian kernel, together with the most common Matérn kernels $\nu = 3/2,\ 5/2$ are shown in Figure 5.2. From this figure, it is evident the relationship between parameter ν and the differentiability of the prediction of the Gaussian process. Stein [136] has used this argument to claim the squared exponential kernel may be too smooth to represent most physical functions accurately. Along these lines, some authors have noted that even though the potential energy surface is often assumed to be of class C^∞ on an open set, it does not need to be so smooth [133], and its surrogate model could benefit for relaxing this constraint to just being twice differentiable or even only C^1. The Matérn kernels with $\nu = 5/2$ and $\nu = 3/2$ have been tested for geometry optimization [133] and transition state search [126, 128] resulting in somewhat moderate gains compared to the squared exponential kernel results reported on the same works.

It is important to note that the potential energy surface is not even a continuous function of the Cartesian coordinates, since it diverges as the positions of any two atoms approach each other. A Gaussian process surrogate potential energy surface would break close to this points. This is illustrated in Figure 5.3, where one of the points in

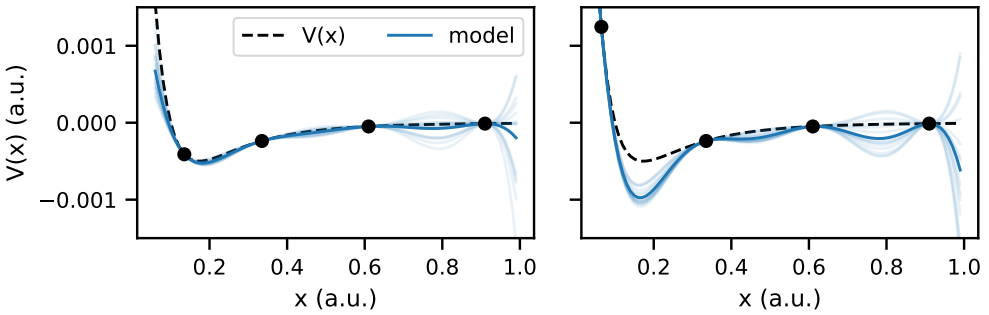


Figure 5.3: Gaussian process regression around a discontinuity of the potential energy surface using the squared exponential kernel (5.9). The dashed line represents the Lennard Jones potential (in arbitrary units) while the blue one is the average prediction of the Gaussian process regression fitted to four points. The training set is the same in both panels except for the left most point. As one of the training points gets closer to the discontinuity, it becomes more difficult to find a C^∞ function that goes through all the points, producing a "weird" behaviour of the model even in regions far away from the discontinuity. Parameters: $\ell = 0.4$, $k_0 = 0.5$, zero prior, zero regularization.

the training set is moved closer to the origin of a Lennard-Jones potential. As a point of the training set moves into the discontinuity, the smoothness constraint becomes more difficult to fulfill. In this example, where the squared exponential kernel with zero regularization is used, forcing the model to describe every point produces spurious local minima in other parts of the input space. However, if there are no points in the training set in the part where the potential energy surface diverges, the same hyperparameters produce fine results. Consequently, sampling these kind of regions should be avoided. Some strategies to achieve this are described in Section 5.3.

The kernels presented in this section depend on two hyperparameters: the prefactor k_0 and the characteristic scale ℓ. The prefactor is the uncertainty in points that are far enough of any other point in the training set, i.e. $\mathbb{V}_{11}^2(\mathbf{x}) \simeq k_0^2$ if $|\mathbf{x} - \mathbf{x}'| \ll \ell$ for all $\mathbf{x}'$ in the training set. It is easy to see that k_0 is also the maximum uncertainty that the trained model can attain at any point. The kernel of a point with itself in equation (5.8) for this family of kernels is always the prefactor squared, and since $\mathbb{C}$ is positive definite, the uncertainty is the squared prefactor minus a quantity that is equal or greater than zero $K(\mathbf{x}, \mathbf{X})\mathbb{C}^{-1}\ K(\mathbf{X}, \mathbf{x})$.

It is worth noting that the prefactor of the kernel does not play any direct role on the average prediction. It is possible to define a normalized kernel $\tilde{k} = k/k_0$ and then rewrite expression (5.7) in terms of the normalized kernel instead. Interestingly, the resulting expression is identical to the original if the regularization is also normalized $\tilde{\Sigma}_n = \Sigma_n/k_0$. Hence, the main role of the prefactor is to tune the uncertainty around the prediction.

The characteristic scale encodes the characteristic distance at which two points are correlated. This is illustrated in Figure 5.4 for the squared exponential kernel. In panels

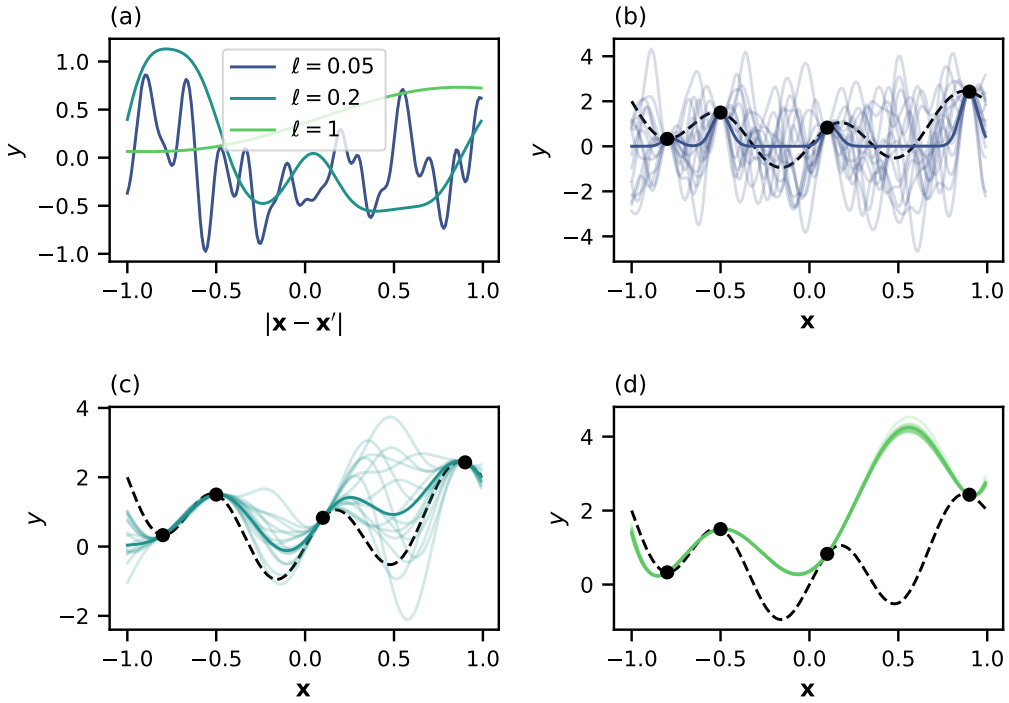


Figure 5.4: Gaussian process regression with the squared exponential kernel using different scales. Panel (a) shows an example of a sample prior function for three scales: $\ell = 0.05, 0.2, 1$. Longer scales in the kernel produce more slowly varying functions than shorter scales. Panels (b), (c) and (d) show the result of the Gaussian process regression to four training points (black dots) of the same underlying function (dashed black line). The average prediction function is marked in a darker colour in each panel, while 15 sample functions of the posterior distribution are shown in the background. Scale $\ell = 0.02$ in panel (b) is clearly varying too fast compared to the underlying function, while scale $\ell = 1$ in panel (d) is too slow. Only a scale of a similar order of variation as the underlying function (as $\ell = 0.2$ in panel (c)) produces a satisfactory model of the target function. The kernel prefactor k_0 in each panel has been chosen to maximize the marginal likelihood. The zero prior function $m(\mathbf{x}) = 0$ has been used and in panel (d) a regularization $\sigma_n^E = \sigma_n^{\mathbf{f}} = 0.01$ has been used to avoid numerical instability.

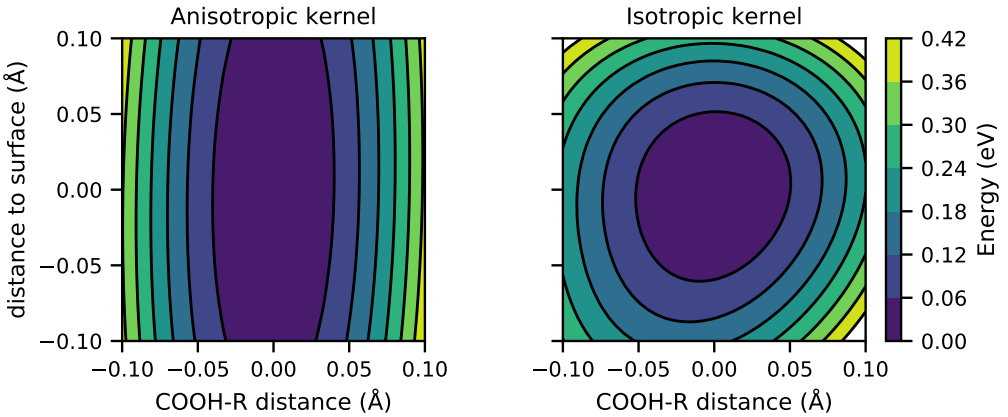


Figure 5.5: Comparison between the potential energy surface model of an isotropic squared exponential kernel of the form (5.9) and an anisotropic one of the form (5.12). The underlying potential energy surface is produced by a DFT calculation of the potential energy of acrylic acid on a palladium surface. The x axis corresponds to the dissociation of acrylic acid into carboxyl and vinyl radicals (breaking a single $C-C$ bond) and the y axis corresponds to the separation of the molecule from the surface. The $C-C$ bond is stiffer than the softer molecule-surface interaction, which is correctly captured by the anisotropic kernel but missed by the less flexible isotropic one. Source: Paper III: *Machine Learning with bond information for local structure optimizations in surface science*.

(b), (c) and (d) of this figure, the result of the Gaussian process prediction is shown for three different scales, $\ell = 0.05, 0.2, 1$. Scale $\ell = 0.05$ is clearly too short and it leads to over-fitting of the training points: the variation of the functions generated by this kernel is too fast compared to the true underlying function and the result is an average prediction that fits every point individually. On the other side of the spectrum, the scale $\ell = 1$ is too long, leading to under-fitting: too slow varying functions cannot capture the variation of the underlying function. The scale $\ell = 0.2$ in panel (c) shows a good compromise between the two extremes, producing a reasonable ensemble of fitting functions.

It is possible to include more than one scale in stationary kernels to model for anisotropy of the target function in different directions of space. For example, this can be done by introducing a positive semi-definite matrix M in the distance measure between $\mathbf{x}$ and $\mathbf{x}'$ [64]. In this framework, the squared exponential kernel reads as:

$$k(\mathbf{x}, \mathbf{x}') = k_0^2 e^{-(\mathbf{x}-\mathbf{x}')^T M (\mathbf{x}-\mathbf{x}')/2}. \tag{5.12}$$

An example of the effect of introducing an anisotropic kernel instead of an isotropic one with the same training set is shown in Figure 5.5.

There are two interesting observations about this point. First, since M is positive semidefinite, it can be decomposed as $M = \Lambda^T \Lambda$, and thus, one could define a fin-

gerprint as a linear transformation of the Cartesian coordinates: $\varrho = \Lambda \mathbf{x}$, and then the kernel becomes the isotropic kernel in the fingerprint space. Second, by introducing a matrix M with one or more zero eigenvalues, it is possible to have a surrogate model of the potential energy surface that only varies along some directions, having some linear combinations of the inputs suppressed from the kernel.

5.2.2 Non stationary kernels

It is also possible to include a model for the covariance function that is not a function of $\mathbf{x} - \mathbf{x}'$. In that case, the scale and magnitude of the variation of the target function are not assumed to vary uniformly along the unit cell (in Cartesian coordinates). This can be of an advantage, for example, to represent more accurately the regions of the PES where the atoms are too close, as shown by Koistinen *et al.* [126]. In that work, the authors replace the distance between configurations in Cartesian space in the squared exponential kernel (5.9) by a measure of distance involving the difference of the inverse of the interatomic distances between configurations. In this way, they achieve a better description of the PES with less points, leading to a faster convergence of the NEB algorithm they are using.

Another example of the use of non stationary kernels (with respect to Cartesian coordinates) is the use of non trivial fingerprints in a stationary kernel. For example, in Paper V we have used the modified Oganov fingerprint (Section 3.4.2) ρ as an input to the squared exponential kernel:

$$k(\rho(\mathbf{x}), \rho(\mathbf{x}')) = k_0^2 e^{-\|\rho(\mathbf{x}) - \rho(\mathbf{x})'\|/2\ell^2}. \tag{5.13}$$

An example of the kind of models this kernel produces can be found in Figure 4.2. The use of a fingerprint, as mentioned in Section 3.4, allows to incorporate the symmetry information into the prior information of the Bayesian model. This way, all the predictions of the Gaussian process will be already symmetrical.

The use of non-stationary kernels often means that the effect of adding new data is non-local in Cartesian coordinates: in this depiction, a new point changes the full PES and not only a neighborhood of the point. In this sense, a careful choice of a non-stationary kernel can lead to a global model of the potential energy surface.

5.2.3 Comparison between kernels with and without fingerprints

In this section, the effect of having a fingerprint as a descriptor is compared to using the squared exponential kernel with Cartesian coordinates as a descriptor. In particular, the modified Oganov fingerprint presented in Section 3.4.2 is used in combination with the squared exponential kernel (as used in paper V) for this illustration.

Figure 5.6 shows a heat map with the entries of the Gram matrix for both kernels for a 7 atom cluster and three points in the training set. The first thing that calls the attention is the block structure of the matrix, which arises as a consequence of the use of energies and forces in both models. Each block represents the correlation between two configurations as in equation (5.5). The diagonal blocks represent the kernel of a structure with itself.

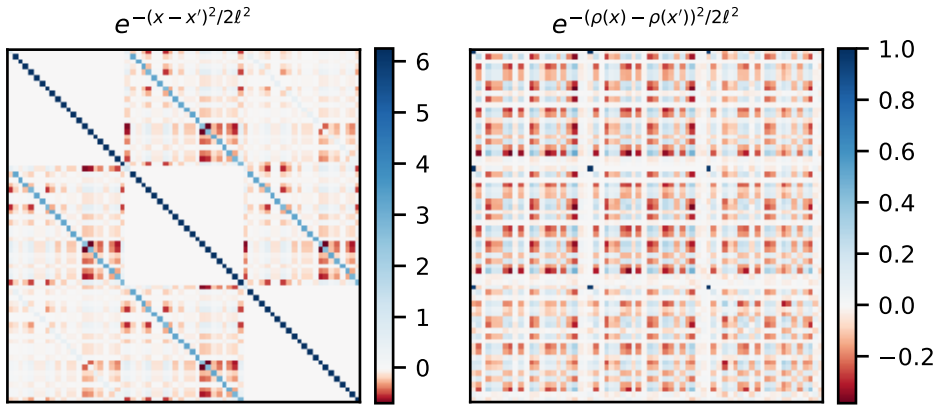


Figure 5.6: Visualization of the Gram matrix of the Gaussian process with two different kernels for three rattled versions of a randomly generated 7 atom gold cluster. The left panel shows the Gram matrix using the squared exponential kernel with Cartesian coordinates. The right panel shows the results for the squared exponential kernel with the modified Oganov fingerprint. Parameters: $\ell = 0.4$ Å for the left panel and $\ell = 0.3$ for the left one. Parameters of the modified Oganov fingerprint: $\delta = 0.4$ Å, $n_{\text{bins}} = 200$, $R_c = 8$ Å.

In Cartesian coordinates, the correlation of a structure with itself is the diagonal matrix $\text{diag}(k_0^2, k_0^2/\ell^2, \ldots, k_0^2/\ell^2)$, and as we see, structures that are "close" in Cartesian space retain the structure where the main entries are still on the diagonal of the matrix. In contrast, the permutation symmetry of the fingerprint results in a model that includes correlations between components of the force even within the same structure. In general, more entries have values that are larger in absolute value for the kernel with a fingerprint than for the one with Cartesian models, leading to richer patterns with less points in the resulting model.

The effect of the correlation between forces can be appreciated in Figure 5.7. The figure compares the surrogate PES with the two mentioned kernels and 4 structures in the training set to the true potential as computed with effective medium theory [10]. The system studied is a gold adsorbate on a gold fcc (100) slab. The atoms in the slab are constrained and the adsorbate has its movement limited to a $x - y$ plane parallel to the slab.

The symmetry conserving properties of the modified Oganov fingerprint results in a model of the potential energy surface that retains the four-fold rotational symmetry of the original potential, even when fitting to just four points that are not symmetrically distributed. In this way, and also thankfully to the relatively long scale, as compared to the distance between the configurations in fingerprint space, the surrogate potential energy surface has the qualitative right structure along all the space.

Conversely, the Cartesian coordinates build a local model of the potential energy sur-

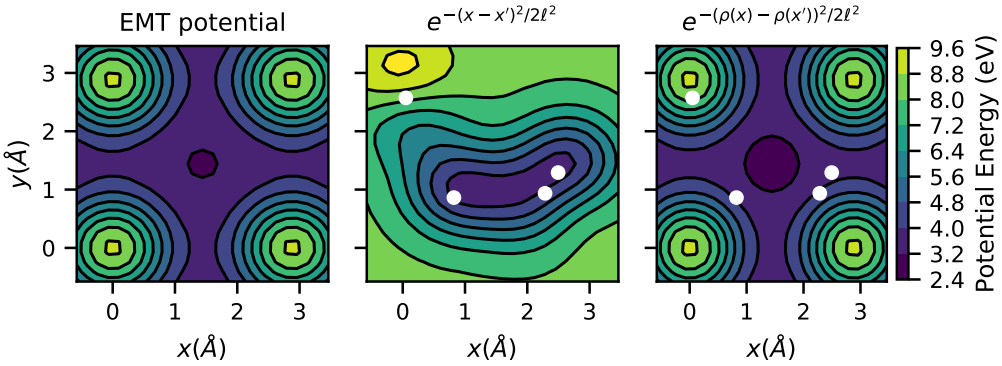


Figure 5.7: Models of the potential energy surface of a gold atom on a fcc (100) gold surface as the adsorbed atom moves at fixed distance, parallel to the surface. x and y represent the coordinates of the adsorbate with respect to the ”on top” position to one of the gold atoms in the surface. The left panel shows the potential energy surface as obtained with effective medium theory(EMT). The remaining panels show the prediction of GPR with different kernels fitted to the four points marked in white and their corresponding EMT energies and forces. The kernels, and their hyperparameters, are: the squared exponential kernel with Cartesian coordinates ($\ell = 0.6$ Å), and the squared exponential kernel with the modified Oganov fingerprint ($\ell = 30$, $\delta = 0.4$ Å, $n_{\text{bins}} = 200$, $R_c = 8$ Å).

face. The description of the potential energy surface is accurate around the configurations in the training set, but it goes back to the prior (which in this case is a constant with the energy of the configuration in the top left) far away from them. However, the model does not encode any symmetry, so the four-fold rotation symmetry is lost. Even though this issue could be addressed by using *data augmentation*, i.e. by using all the symmetry transformations of the configurations to enlarge the training set; this is not advisable in general, since it would increase the computational cost of the Gaussian process model.

The Cartesian coordinates present an advantage over this particular fingerprint under some settings, though. Figure 5.8 shows the learning curves for the two kernels for the same system. The advantages of using the modified Oganov fingerprint for small training sets are evident: for training sets of about 10 configurations, the mean absolute error over the whole unit cell is about an order of magnitude smaller for the modified Oganov fingerprint with optimal hyperparameters than for the usual set up in GPMin (see Paper II).

However, the performance of the model that uses the modified Oganov fingerprint with the squared exponential kernel saturates when the size of the training set increases, even when the hyperparameters are optimized (see Paper V for more learning curves and details over this issue). The Cartesian coordinates perform worse with less data, as it is evident from Figure 5.7: the error will remain high over the full unit cell until there

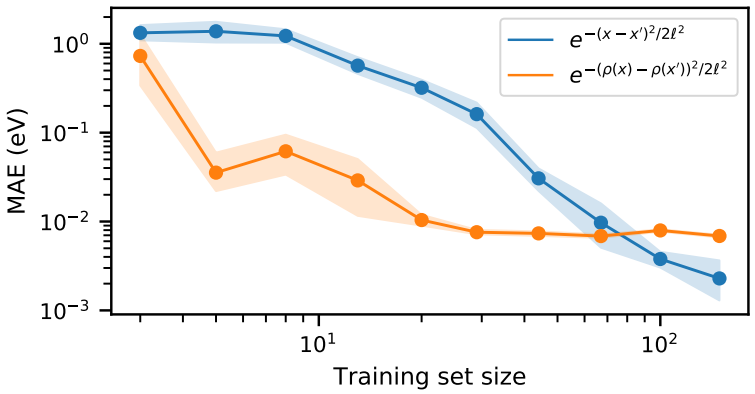


Figure 5.8: Learning curves of two Gaussian process models of the potential energy surface with the squared exponential kernel with two different fingerprints: the Cartesian coordinates and the modified Oganov fingerprint. The atomic system studied is the same as in Figure 5.7, and the training set has been generated sampling the x and y coordinates of the adsorbate within the unit cell. The solid line shows the average over 10 randomly generated training sets of the indicated size and the shaded area shows the 95% confidence interval around the average as estimated using boostrapping. Parameters: Cartesian coordinates ($\ell = 0.4$ Å), and modified Oganov fingerprint (ℓ optimized , $\delta = 0.4$ Å, $n_{\text{bins}} = 200$, $R_c = 8$ Å), initial $\sigma_n = 10^{-5}$eV/Å.

is data enough so that the support of the kernel function around the points covers the unit cell. However, once the full area of the unit cell is covered, the precision of the model increases as more data points are added.

From these plots we can infer that choosing an adequate kernel for each application is advisable when using Gaussian process regression to build surrogate models of the potential energy surface.

5.3 The prior function

The prior function may also play an important role in the resulting potential energy surface. One of the most usual choices is to choose a constant prior function of the form:

$$m(\varrho) = m_0 \qquad \forall \varrho, \tag{5.14}$$

which acts as a zero-energy reference.

The Gaussian process regression, thus, learns the distance to this offset when regression is built. Furthermore, this constant can be optimized by maximizing the marginal log-likelihood, but it can also be used to enforce some expert knowledge of the potential energy surface. In particular, during local optimization, if it is used in combination with a stationary kernel it can be used to dial the exploration/exploitation balance of

the method by penalizing extrapolation. Keeping this constant high compared to the energies in the training set forces the method to search for low lying points close to the already sampled ones, since far away from the sampled points the model will predict very high energies. This characteristic has been used both by Denzel and Kästner [133], who set the constant m_0 to a value higher than the maximum energy in the training set, and in Paper II: *Local Bayesian optimizer for atomic structures*, where m_0 is set to the highest sampled energy.

More sophisticated prior functions can be used to include expert knowledge of the potential energy surface in the surrogate model [110]. In general, any potential energy surface model could be used as a prior function, and then the task of the Gaussian process would be to learn the error between the model potential and the output of the more sophisticated electronic structure code.

A particularly relevant application to this is the use of a short-range diverging potential in the vicinity of the point in Cartesian space where two nuclei have the same coordinate, in combination with a constant prior. In the context of global optimization of atomic systems, Bisbo and Hammer [114] have recently introduced the following expression for the prior potential energy surface:

$$m(\varrho) = \sum_{ij}^{N} \left(0.7 \, \frac{r_i^{(c)} + r_j^{(c)}}{|\mathbf{R}_i - \mathbf{R}_j|} \right)^{12}, \tag{5.15}$$

where $r_i^{(c)}$ is the covalent radius of the $i-$th atom and $\mathbf{R}_i$ its Cartesian coordinate. This repulsive potential produces very high energies in regions where two atoms are close, preventing that any minimization algorithm will suggest these regions as potential minima, while maintaining the uninformed constant prior elsewhere. This solves the problem that many stationary kernels encounter when one of the points in the training set is close to the divergence, which was exposed in the previous section (see Figure 5.3). This prior has also been used in Paper V: *Global optimization of atomic structures with gradient-enhanced Gaussian process regression*.

5.4 Maximizing the marginal likelihood

The kernels and the priors introduced in this chapter, together with the descriptor of the atomic system (as discussed in Chapter 3), may depend on some hyperparameters $\boldsymbol{\theta}$: For example, the squared exponential kernel (5.9) depends on the scale ℓ and the prefactor k_0, and a constant prior has a constant m_0 that needs to be determined.

The hyperparameters of the model can be found in the Bayesian framework by optimizing the marginal likelihood $p(\mathbf{Y}|\boldsymbol{\theta};\boldsymbol{\rho})$, that is, by finding the hyperparameters that maximize the probability to obtain the targets given the inputs. Expression (3.16) can be rewritten in the notation of the Gaussian process regression with forces for atomic systems introduced in this chapter as:

$$\log p(\mathbf{Y}|\boldsymbol{\theta};\boldsymbol{\rho}) = -\frac{1}{2}(\mathbf{Y} - \mathbf{m}(\boldsymbol{\rho}))^T \mathbb{C}^{-1}(\mathbf{Y} - \mathbf{m}(\boldsymbol{\rho})) - \frac{1}{2}\log\det(\mathbb{C}) + \mathfrak{N}, \tag{5.16}$$

where $\mathfrak{N}$ is the normalization, which does not depend on the hyperparameters. This expression depends on the hyperparameters through the expressions for $\mathbf{m}$, $\mathbb{C} = K(\boldsymbol{\rho}, \boldsymbol{\rho}) + \Sigma_n^2$, and the descriptor ϱ itself.

The marginal likelihood in expression (5.16) has three terms. The first one is the only term involving the targets $\mathbf{Y}$, and thus, it is a measure of the quality of the fit. The second term depends only on the inputs, and since $\mathbb{C}$ is a positive definite matrix, it grows as one or more eigenvalues of $\mathbb{C}$ approach the regularization (their minimum possible value). Thus, the second term is a complexity penalty [64], some sort of Occam's razor term to favor hyperparameters that lead to "simpler" models: those that assume points in the training set are very correlated with each other. The last term, as mentioned above, is just a normalization factor.

Even though it is possible to find closed expressions that maximize marginal likelihood for some of the hyperparameters, the optimum can be found using a gradient based optimizer, since it is posible to find a closed expression for the derivatives of the marginal likelihood with respect to any hyperparameter θ_j:

$$\begin{aligned}\frac{\partial}{\partial \theta_j} \log p(\mathbf{Y}|\boldsymbol{\theta};\boldsymbol{\rho}) = &\frac{\partial \mathbf{m}(\boldsymbol{\rho})^T}{\partial \theta_j}\mathbb{C}^{-1}(\mathbf{Y}-\mathbf{m}(\boldsymbol{\rho})) \\ &+ \frac{1}{2}(\mathbf{Y}-\mathbf{m}(\boldsymbol{\rho}))^T\mathbb{C}^{-1}\frac{\partial \mathbb{C}}{\partial \theta_j}\mathbb{C}^{-1}(\mathbf{Y}-\mathbf{m}(\boldsymbol{\rho})) \\ &- \frac{1}{2}\mathsf{Tr}\left(\mathbb{C}^{-1}\frac{\partial \mathbb{C}}{\partial \theta_j}\right),\end{aligned} \tag{5.17}$$

where $\mathsf{Tr}(\cdot)$ stands for the trace of the matrix. This expression, in combination with expression (5.16), results in a computationally affordable way to find optimal hyperparameters using a gradient based optimizer, such as the ones described in Chapter 4.

The marginal likelihood is not, in general, a convex function of the hyperparameters [64]. There might be more than one local minimum, leading to different models that are locally optimal in their hyperparameters. If there is no expert knowledge on where the globally optimal basin may lie, the global optimum can be found, for example, by using a gradient based optimizer with restarts [114].

The following sections discuss the optimal value of some of the most commonly used hyperparameters and how to find them.

5.4.1 The prefactor of the kernel

The prefactor of the kernel k_0 can be determined analytically under some assumptions. If it is assumed that the ratio between the prefactor and the regularization are kept fixed, i.e. $k_0/\sigma_n^{(E)}$ and $k_0/\sigma_n^{(\mathbf{f})}$ are kept constant, it is possible to factor the prefactor out of the regularized Gram matrix $\mathbb{C}$. Under these conditions, it can be shown that the value of the prefactor for which the partial derivative of the marginal likelihood with respect to the prefactor vanishes is:

$$k_0 = \sqrt{\frac{\left(\mathbf{Y} - \mathbf{m}(\boldsymbol{\rho})\right)^T \mathbb{C}_1^{-1} \left(\mathbf{Y} - \mathbf{m}(\boldsymbol{\rho})\right)}{n}}, \tag{5.18}$$

where $\mathbb{C}_1$ is the regularized version of the Gram matrix with prefactor equal to 1.

This value optimizes the prefactor given the rest of the hyperparameters. As a consequence, equation (5.18) can be used to reduce the dimensionality of the optimization problem by one.

5.4.2 The constant of the prior

It is also possible to obtain the value of the constant if the prior function is set to be the constant prior:

$$m_0 = \frac{\mathbf{U}^T \mathbb{C}^{-1} \mathbf{Y}}{\mathbf{U}^T \mathbb{C}^{-1} \mathbf{U}} \tag{5.19}$$

where **U** is the constant prior with constant equal to 1:

$$\mathbf{U} = \begin{cases} 1 & \text{if mod}(3N+1) = 0, \\ 0 & \text{otherwise.} \end{cases} . \tag{5.20}$$

Note that the constant of the prior does not depend on the prefactor of the kernel if the ratio between the prefactor and the regularization is kept fixed (which was also assumed in the expression for the optimal prefactor), since it can be factored out. Expression (5.19) can be used to reduce the complexity of the optimization of the marginal likelihood even further, since the constant of the prior can be determined analytically and then its value can be used to determine the prefactor of the kernel.

It is straightforward to extend expression (5.19) to find the constant in the prior when the prior is a sum of a constant with a potential, as has been done in paper V: *Global optimization of atomic structures with gradient-enhanced Gaussian process regression*.

5.4.3 The scale

There is no direct expression for the optimization of the scale, whose optimal value must be obtained using a numerical method.

It is important to note that good values of the scale can only be obtained by optimizing the prefactor of the kernel along with it, for example by using expression (5.18). Even though the prefactor does not influence the prediction directly and its main contribution comes from the uncertainty, the prefactor should be optimized along with the scale even when the uncertainty is not used in the application. This is because the marginal likelihood depends on the scale in a way that cannot be decoupled from the value of the prefactor. If the prefactor is not updated along with the scale, the optimal scale becomes a function of the prefactor, and thus it is possible to obtain virtually any scale by varying the value of the prefactor. This behaviour is illustrated in Figure 5.9.

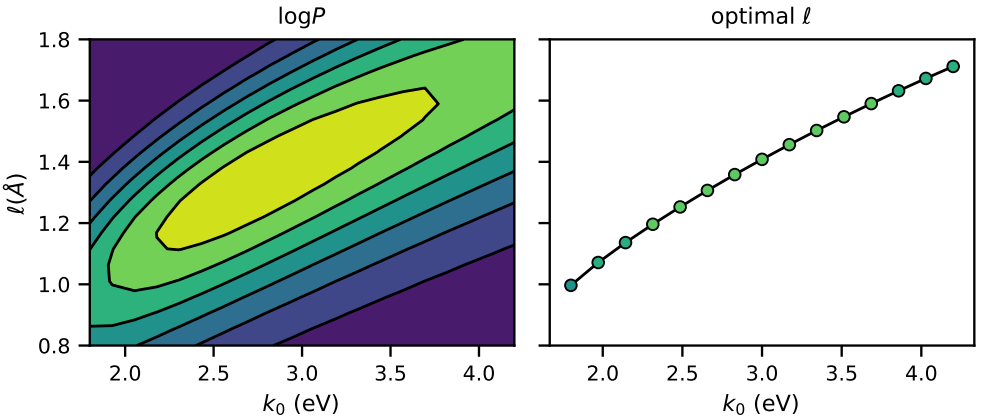


Figure 5.9: If the prefactor is fixed, the optimal scale is a function of it. The left panel shows the logarithm of the marginal likelihood as a function of the prefactor of the squared exponential kernel k_0 and the scale ℓ. The right panel shows the scale that is numerically found to optimize the marginal likelihood if the prefactor k_0 is kept fixed (all optimizations start at $\ell = 0.7$ Å. The color in both panels marks the value of the logarithm of the marginal likelihood for the current hyperparameters. The training set for both panes is composed of 10 slightly rattled copies of a nine atom aluminum cluster. The energies and forces of the aluminum clusters have been computed using the EMT potential.

5.5 The numerical inversion of the Gram matrix

All the expressions needed to fit a Gaussian process to a potential energy surface presented in this chapter, that is, the average prediction (5.7), the uncertainty (5.8), the marginal likelihood (5.16) and its derivative (5.17); depend on the inverse of the regularized Gram matrix $\mathbb{C}$. In this section, how to solve this problem and the challenges associated to it are discussed.

5.5.1 The regularization

In the absence of regularization, some Gaussian process models may present a Gram matrix that is ill-conditioned for inversion. If two points in the training set have exactly the same descriptor $\varrho = \varrho'$, then the Gram matrix $K(\boldsymbol{\rho}, \boldsymbol{\rho})$ will have two equal rows, and thus, not be invertible. Even if the two descriptors are not identical, it might happen that depending on the choice of the kernel, the hyperparameters and the composition of the training set, the Gram matrix is ill-conditioned for inversion. This may result on the numerical routine solving the linear algebra problem producing unstable and noisy results or even failing all together to invert the matrix.

This issue can be solved by adding a small, non-zero regularization factor. The fitness of the Gram matrix to be inverted can be quantified by its condition number, which in the ℓ_2-norm is the quotient between the largest and the smallest eigenvalue $\lambda_{max}/\lambda_{min}$. If the inverse of the condition number is close to the machine precision, numerical round-off errors may arise, making the surrogate PES noisy (enough to make a gradient

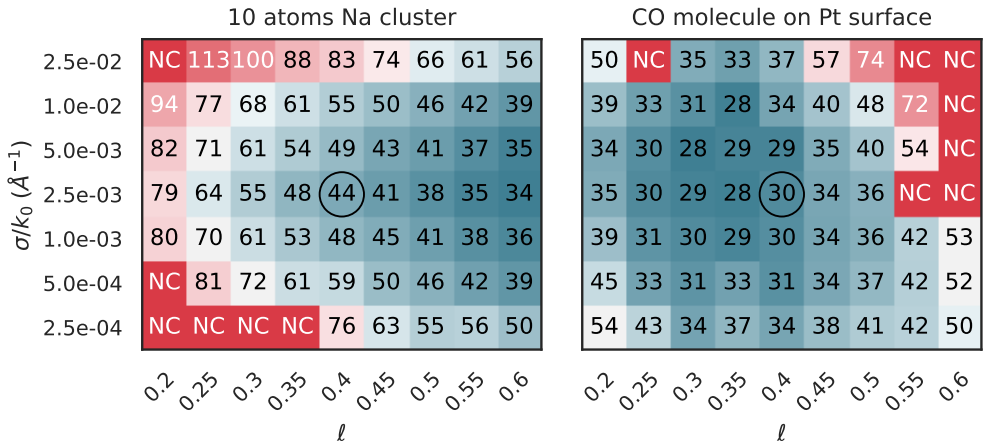


Figure 5.10: Performance of a local optimization method (BondMin, see Chapter 6) based on a surrogate Gaussian process model for two atomic structures. The color and the number on each square represents the number of DFT evaluations (the smaller the better) needed to minimize the function and NC shows that the optimization did not converge. All hyperparameters have been kept fixed, with $k_0 = 1$eV, such that the y axis has eV/Å units. The tolerance on the convergence for the forces of the SCF iteration has been fixed to 10^{-4} eV/Å. In the plot it can be appreciated how the performance of the optimization diminishes as the regularization approaches this value. Source: Paper IV: *Machine Learning with bond information for local structure optimizations in surface science*.

based method fail) [137]. It is easy to show that by adding a regularization $\sigma_n^2\mathbb{I}$ to the matrix, the new condition number becomes $(\lambda_{max} + \sigma_n^2)/(\lambda_{min} + \sigma_n^2)$ which is approximately λ_{max}/σ_n^2 if $\lambda_{min} \ll \sigma_n^2 \ll \lambda_{max}$.. Hence, by choosing a suitable value of the regularization is crucial for obtaining smooth potential energy surfaces.

As noted above, the prefactor k_0 of the kernel can be factored out of some of the Gaussian process equations. In fact, the prefactor k_0 can also be easily factorized out of the condition number of the Gram matrix $K(\rho, \rho)$, *i.e.*, the condition number only depends on the distribution of the data and the other hyperparameters of the kernel. Hence, the effective amount that has to be added to the Gram matrix to ensure inversion is σ_n^2/k_0^2: If k_0 is changed to describe the data better, if σ_n is not changed accordingly, there is a risk that matrix inversion will fail. For this reason, in the work presented in this thesis, the ratios σ_n^E/k_0 and $\sigma_n^{\mathbf{f}}/k_0$ have been kept fixed during the optimization of the hyperparameters, so that the robustness of the methods that are presented here is ensured.

In addition, it is worth noting that the potential energy surface as described by the result of the DFT calculations can be, in fact, noisy. As exposed in Section 2.3.3, the self-consistent field iteration terminates at a given tolerance, and differences between energies and forces of different configurations, even if they are close, may be dominated by numerical noise.

This effect can be observed in Figure 5.10, where the position of the potential energy surface minimum becomes more difficult to capture for the Gaussian process model as the regularization becomes comparable in magnitude to the tolerance on the force of the SCF iteration.

5.5.2 The Cholesky factorization and beyond

Even though in the previous section the terminology "inversion" of the Gram matrix has been used, solving the algebraic problem $\mathbb{C}\mathbf{z} = \mathbf{b}$ is known to be more robust against numerical noise than computing the inverse of $\mathbb{C}$ and then computing the product $\mathbb{C}^{-1}\mathbf{b}$. In this section, a way to solve this problem efficiently is discussed.

The Gram matrix $\mathbb{C}$ is real, symmetric, and positive definite, and thus, it can be factorized as:

$$\mathbb{C} = LL^T \tag{5.21}$$

where L is a lower triangular matrix (this is, a matrix whose only non-zero elements are in the diagonal or below) with positive diagonal [137]. This factorization is called the *Cholesky factorization*.

The advantages of factorizing the Cholesky matrix are evident: Once factorized, the algebraic problem $\mathbb{C}\mathbf{z} = \mathbf{b}$ can be solved by finding the intermediate solution $\hat{\mathbf{z}}$ first by solving

$$L\hat{\mathbf{z}} = \mathbf{b}, \tag{5.22}$$

and then finding the unknown $\mathbf{z}$ from

$$L^T\mathbf{z} = \hat{\mathbf{z}}. \tag{5.23}$$

The two subproblems are then trivial to solve, given the structure of the Cholesky factor L: starting with the first element of the first subproblem (5.22), $\hat{z}_1 = b_1/l_{11}$, the problem is solved iteratively as:

$$\hat{z}_i = \left(b_i - \sum_{j=1}^{i-1} l_{ij}\hat{z}_j \right) / l_{ii}. \tag{5.24}$$

A similar expression can be derived for subproblem 5.23.

This way, the factorization can be computed once for each training set and set of hyperparameters, with computational cost $O(n^3N^3)$, and then this factorization can be used to solve all the algebraic problems involved in prediction of the energy and the uncertainty for any test point ϱ that is required by the application at a much lesser cost. The need to store the matrix in order to use the Cholesky implementation in Scipy [138] produces a memory cost of that scales quadratically with the size of the matrix.

The computational cost of the Cholesky factorization can become comparable to the DFT one, since both scale cubically with the number of atoms in the unit cell and the growing prefactor with the number of points in the training set for the potential energy surface can outgrow the cost of the number of times the Hamiltonian needs to be diagonalized in the self consistent field iteration. Furthermore, the need of storing the Gram matrix often becomes the main bottle neck for the modelling of potential energy surfaces for large systems when the Scipy implementation is used.

Modern implementations of density functional theory are highly parallelized since the calculations that use them are usually executed in large supercomputer facilities. Thus, it is natural to devise methods that can take advantage of these kind of architectures. Along these lines, there have been recent developments to develop highly parallelizable implementations of Gaussian process regression that would improve the computational scaling and reduce the overall cost of the exact regression [139, 140].

6 Summary of the contributions

6.1 Paper I: Materials property prediction using symmetry-labeled graphs as atomic-position independent descriptors

In this paper, we introduce a descriptor of the atomic structure that does not depend on the interatomic distances, but rather only on the connectivity of the atoms and the local symmetry of each atomic environment. These descriptors are no longer good descriptors of individual atomic structures, since they are degenerate for atomic with similar configurations, becoming a representation for *prototypes*: a materials "templates" that do not depend on the scale of the material and admit small distortions used in materials science to classify materials with similar structures. By using such representations, we show that it is possible to build a machine learning method to identify the prototype with the minimum formation energy, providing with a way to identify global minimum basins.

To this end, we have used the quotient graph of the material, using the Voronoi tessellation to define the connectivity between atoms, as described in Section 3.4.3. The nodes of the graph represent the atoms of the atomic structure and the edges represent the connectivity. Each node has a vector assigned that represents its state, that is, the atom type. In the paper, we introduce a novel encoding of the symmetry of the atomic environment of each atom, by labelling each edge with the point group of the face of the Voronoi tessellation it traverses.

We have used a message passing neural network, whose details are introduced and discussed in Section 3.2, to predict the formation energy of bulk materials. Even though the absence of interatomic distance information is clearly a challenge for the prediction (since the descriptors are degenerate and not complete), we obtain mean absolute prediction error as low as 22 meV per atom for OQMD [29, 30], and 43 meV per atom in Materials project [32].

These results are shown in Figure 6.1, where the novel descriptor (labelled as Sym) is compared with the same descriptor without the symmetry information (No sym), and the quotient graph with distance information as presented in a previous publication (Distance) [43]. As a base line, we have also compared the out results with a random forest method trained on a descriptor based on the Voronoi tessellation of the material introduced in reference [76]. The MAE for each set has been computed using 5-fold cross validation. From this plot, it is easy to see that the larger the amount of information incorporated to the descriptor, the lower the prediction energy for all subsets. Another trend along all subsets is that the method has a larger prediction error in unary systems than in binary or ternary systems. This trend could be explained by the relatively smaller number of unary entries in any of the training sets, as compared to ternaries, and the high relative proportion of unstable structures.

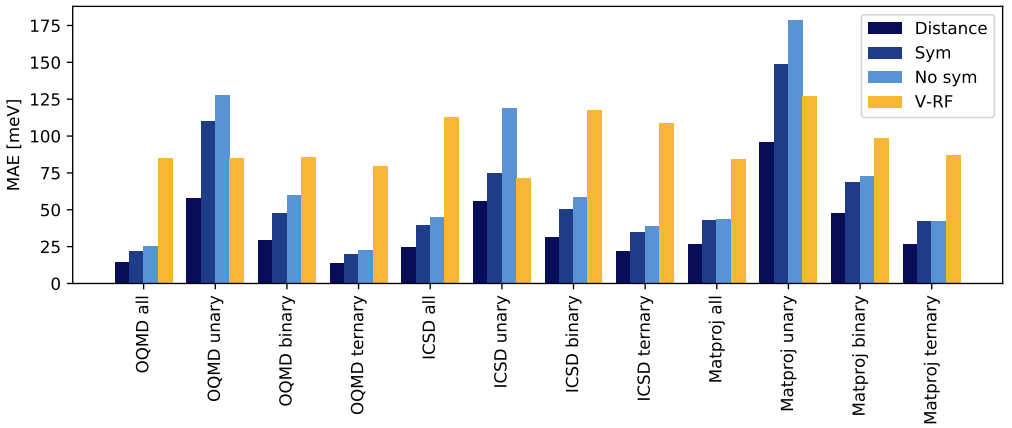


Figure 6.1: Mean absolute error of the message passing neural network model with different levels of information: interatomic distances, symmetry and connectivity and connectivity only; and the Voronoi random forest as presented in reference [76]. Source: *Paper I: Materials property prediction using symmetry-labeled graphs as atomic-position independent descriptors.*

For similar reasons, the errors are smaller on the full OQMD (systematic + experimental structures) than in the ICSD [54] part of OQMD (experimental structures). Because of the way the systematic part of OQMD is built, some symmetry groups, for example those in cubic prototypes, are more common in this subgroup than they are among experimental structures. Conversely, some experimental structures might be the only representative of their prototype in OQMD. The more examples of a certain prototype there are in the training set, the more precise the prediction of the energy is for structures in the test set in the same prototype. As a consequence, the mean absolute error is lower for the systematic subset of OQMD than for the experimental subset.

We have demonstrated the utility of this method on screening studies: that is, materials science research in which many materials are simulated with the hope of identifying candidate materials for a target application. A common problem of these studies is that many of the materials studied are not even stable, resulting in a waste of computational resources. In this paper, we have demonstrated how the expensive DFT calculations can be substituted by our machine learning method in the context of determining the stability different prototypes of the same material.

Figure 6.2 shows the learning curve of the method on a data set of 5976 ternary selenides ($ABSe_3$ with A and B transition metals). These materials have been generated by choosing the six most common $ABSe_3$ prototypes in ICSD [54] and generating all the atomic substitutions of A and B transition metals and then relaxing the structure, in a procedure similar to the one exposed in reference [141]. We note that only six $ABSe_3$ out of the 5976 materials are present in OQMD as well, so this test takes place mostly on unseen data.

Figure 6.2 explores two different strategies: to train on the ternary selenides solely, and

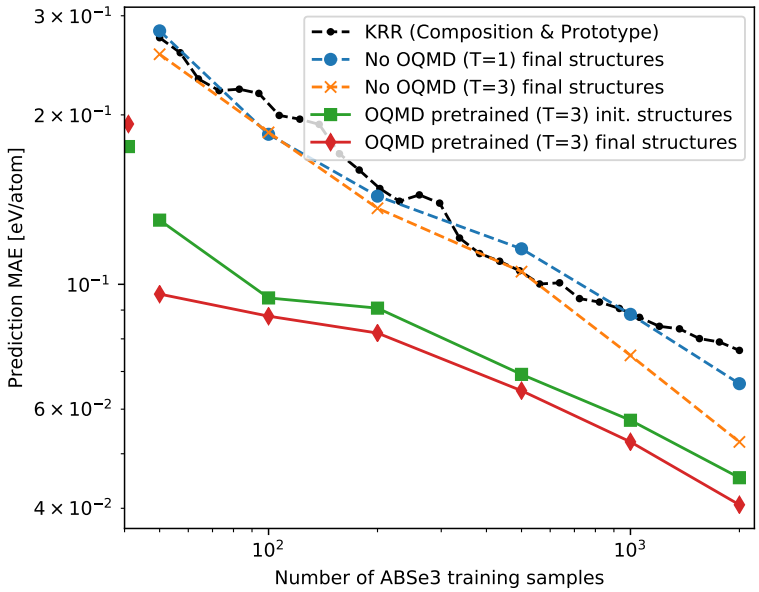


Figure 6.2: Learning curve of the message passing neural network method on the symmetry-labelled position independent graph descriptors with different training strategies on a dataset made up of $ABSe_3$ structures with little overlap with OQMD. The blue and yellow curves show the learning curves for the method trained only on the ternary selenides data set with T=1 and T=3 message passing layers, respectively. The green and red curves show the learning curve of the method on the selenides when it has been previously trained on the OQMD, the green curve for the initial prototypes and the red curve for the relaxed structures. The dashed black like represents a baseline kernel ridge regression method, used for comparison. Source: *Paper I: Materials property prediction using symmetry-labeled graphs as atomic-position independent descriptors.*

to pretrain on OQMD and then add the ternary selenides to the training set. We find that pretraining on OQMD and then adding a small percentage of selenides presents a significant advantage with respect to just training on OQMD: The MAE drops from 176 meV per atom for the model trained on OQMD only to 95 meV per atom for a model trained on OQMD plus 100 selenides (out of 6 thousand), which is already comparable to the accuracy of DFT. Furthermore, by adding more structures to the training set, which could be gathered as the screening progresses, for example; the accuracy of the model improves. The figure also shows that the learning curves obtained by training on the graph of the initial and the relaxed structure run parallel, showing that using the initial graphs for screening is possible with only a small accuracy loss.

6.2 Paper II: Local Bayesian optimizer for atomic structures

In this work, we introduce GPMin, a new method to find the local minima of the potential energy surface based on Gaussian process regression. The method is based on an active learning strategy: a surrogate model of the potential energy surface is built. Then, the surrogate model is used to find a local minimum neighboring the starting point, since optimizing the model is computationally cheaper than minimizing the DFT potential itself. Then, the energy and forces of the minimum of the Gaussian process model are computed using density functional theory, and the structure is included in the training set. These steps are repeated until a structure fulfilling the convergence criterion (described in Section 4.1) is met. We note that the accuracy of the method is equivalent to the one of any other local optimization method, since the convergence criterion is applied to the result of a DFT calculation meaning that the result of the method is an atomic structure whose forces are below a threshold as described by DFT, and not a machine learning prediction.

The surrogate model is built as a Gaussian process regression with energy and force information. We have used the squared exponential kernel with Cartesian coordinates and a constant prior. The prior constant has been kept fixed to the maximum energy sampled of the training set. This choice, combined with the choice of always starting the surrogate minimization from the point with the lowest energy benefits the convergence of the method: at each optimization step, all the points that are very far away (as compared to the length of the kernel) from the points that are in the training set will have high energies. Thus, the algorithm always suggests points that are close to the points in the same basin, preventing extrapolation. The high prior naturally creates some sort of trust region, with the scale being the parameter that controls the trust radius.

Figure 6.3 compares the results of the optimization of one thousand 10 atom gold clusters with GPMin to other usual optimization method for atomic structures introduced in sections 4.1.2 and 4.1.3. Here, the clusters are described with effective medium theory (EMT) [10]: there is no noise in the PES derived from the SCF iteration and the PES is probably smother than the one obtained with DFT, but it serves as an illustration of the possibilities of the method.

For the version of GPMin with fixed hyperparameters, the figure shows an optimal value of the scale of the kernel of about 0.5 Å. For this scale, both the average number of EMT evaluations and its dispersion are lower than the traditional optimizers we have compared it with. However, we note that if the scale is far from the optimal one, the efficiency of the method is reduced.

GPMin is able to find a suitable scale by optimizing the hyper parameters: in spite of having an initial scale that is too short or too long, the method is able to self-correct along the way and find the minimum faster. The value of the optimal scale for the updated version of GPMin is almost identical to the version without updates, 0.5 Å, and for scales smaller than this value, the ability of the method to self correct yields results that are very similar to those of the optimal version. In contrast, we note that

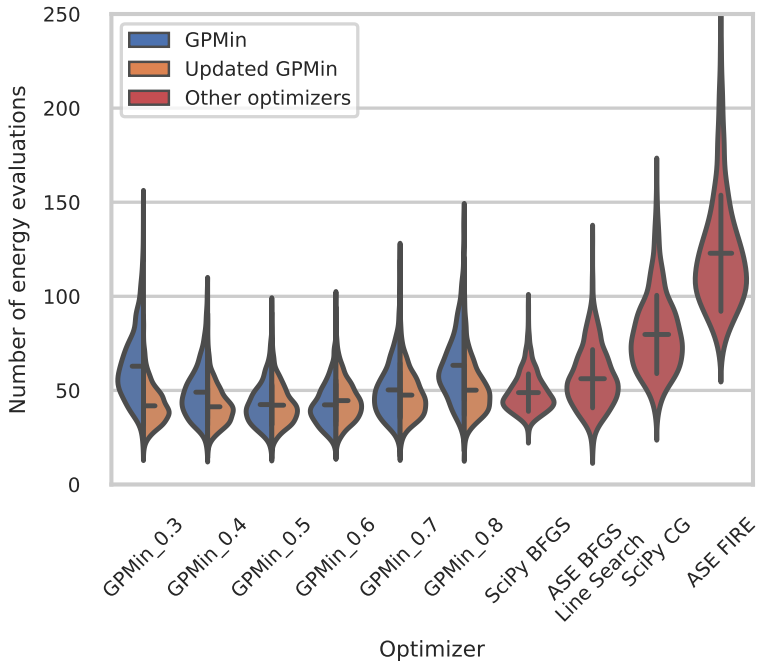


Figure 6.3: Distribution of the number of EMT evaluations necessary to optimize 1000 randomly generated ten atom gold clusters with different optimization methods. The performance of GPMin is shown both with fixed parameters and with updated scale and prefactor, for different values of the (initial) scale. The line in the center of the violins represents the average of the distribution. Source: *Paper II: Local Bayesian optimizer for atomic structures*.

the method with longer initial scales also needs more EMT evaluations.

We have used these findings and the knowledge about the optimal regularization exposed in Section 5.5.1 to find a good set of default hyperparameters for the versions with and without hyperparameter updates for the optimization of DFT systems. To this aim, we have used two different systems and found a set of parameters that minimize the number of DFT calculations needed to relax them.

We have further validated the results of GPMin on a set atomic structures of different nature (including clusters, molecules, surfaces, bulk systems and molecules on surfaces), showing a consistent speed up in the relaxation of those of about a 20% with respect to the fastest methods. We attribute this characteristic to the ability of the Gaussian process regression to describe accurately landscapes where the curvature of the potential energy does not have a well defined sign, in contrast with the competing BFGS strategy, which can only yield convex models. Thus, we have demonstrated the potential of Gaussian process regression models to speed up the optimization of

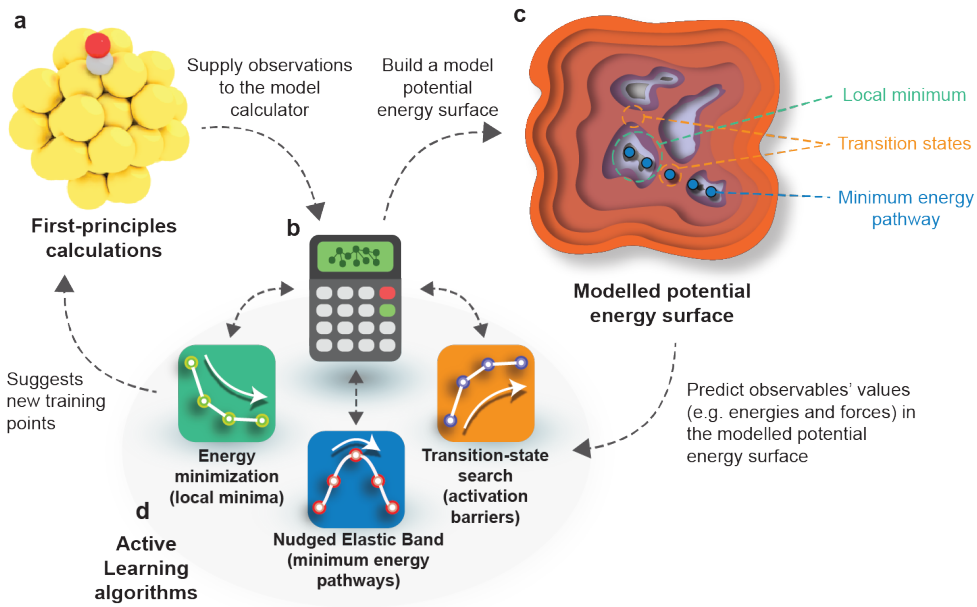


Figure 6.4: AID framework: illustration of the different components of the framework. The package consists on a Gaussian process calculator (**b**) that can be queried by the active learning optimization algorithms (minimization, nudged elastic band and transition state search). The calculator produces a surrogate model of the potential energy surface (**c**) that is used to guide the searches. In addition, the active learning optimization algorithms can suggest useful training points, for which a first principle calculator is executed (**a**). The energies and the forces obtained by the first principles calculator are then supplied to the Gaussian process calculator, which uses this information to produce more accurate models. Source: *Paper III: An artificial intelligence-driven approach for the exploration of potential energy surfaces.*

atomic systems.

6.3 Paper III: An artificial intelligence-driven approach for the exploration of potential energy surfaces

Following the successful reduction of DFT evaluations in local optimization of the GP-Min method and the Gaussian process accelerated nudged elastic band presented in reference [127], we have presented a unified artificial intelligence-driven (AID) framework for Gaussian process assisted local optimization, transition state search and minimum energy pathway discovery.

A scheme of the framework is presented in Figure 6.4. The central characteristic of the framework is the Gaussian process calculator, which stores the energies and forces

of all the previously visited configurations. This feature can be then used by any of the algorithms, facilitating any method to take advantage of previous DFT calculations. For example, a local optimization run aiming to determine the energy of an intermediate structure in a reaction can benefit of the energies and forces of the structures explored by a former NEB calculation of a neighboring minimum energy path to produce a more accurate intermediate surrogate model and, thus, speed up convergence.

In order to keep the memory consumption of the Gaussian process regression, we have devised a limited memory scheme by training only to a subset of all the points. The full discussion of the limited memory scheme can be found in Section 6.4.1.

The Gaussian process calculator can then be queried by any of the active learning algorithms (the local optimization, the NEB or the dimer method) in order to obtain a prediction or an uncertainty for a test configuration, or be updated so that it includes new configurations with DFT information in the training set.

We show that sharing the data between the different active learning algorithms can lead to reduction of the number of DFT calculations more than an order of magnitude when the problem involves the exploration of a large portion of the potential energy surface. This is illustrated in Figure 6.5. The problem addressed here is to determine the reaction network of a two dimensional system involving three final states, labelled S1, S2 and S3, which have been previously relaxed.

The data from the relaxations of S1 and S2 helps to speed-up the determination of the minimum energy path between S1 and S2, and images from the NEB, in turn, speed up the determination of the geometry of the intermediate state I1. The benefits of this approach become even more evident when the NEB calculation to study the S1-S3 minimum energy path is run. The path connecting the two states also runs through intermediate state I1, which has already being converged in the previous calculation. This way, the computational effort can be concentrated in the area around intermediate state I3, instead of evenly distributing the computational cost throughout the space.

The full reaction network in this example has been determined with 142 energy-force evaluations, as compared to 1687 energy-force evaluations needed by MDMin [95], the most efficient amongst the traditional methods we have compared it to.

We further demonstrate the power of the AID framework by addressing the dissociation of the absorbed CH radical on a copper surface with a step (fcc (211)), as described with DFT. Our investigations have revealed that there are 5 different configurations where CH is bound and 21 where the carbon and the hydrogen atoms are adsorbed to different sites, out of 159 initial candidate structures. We have then computed the minimum energy pathways between every bound and dissociated configurations, needing 105 NEB calculations. We were able to determine this network with less than 9000 DFT evaluations, where only running the local optimization part with the traditional optimizers we tested required over 10 000 DFT evaluations.

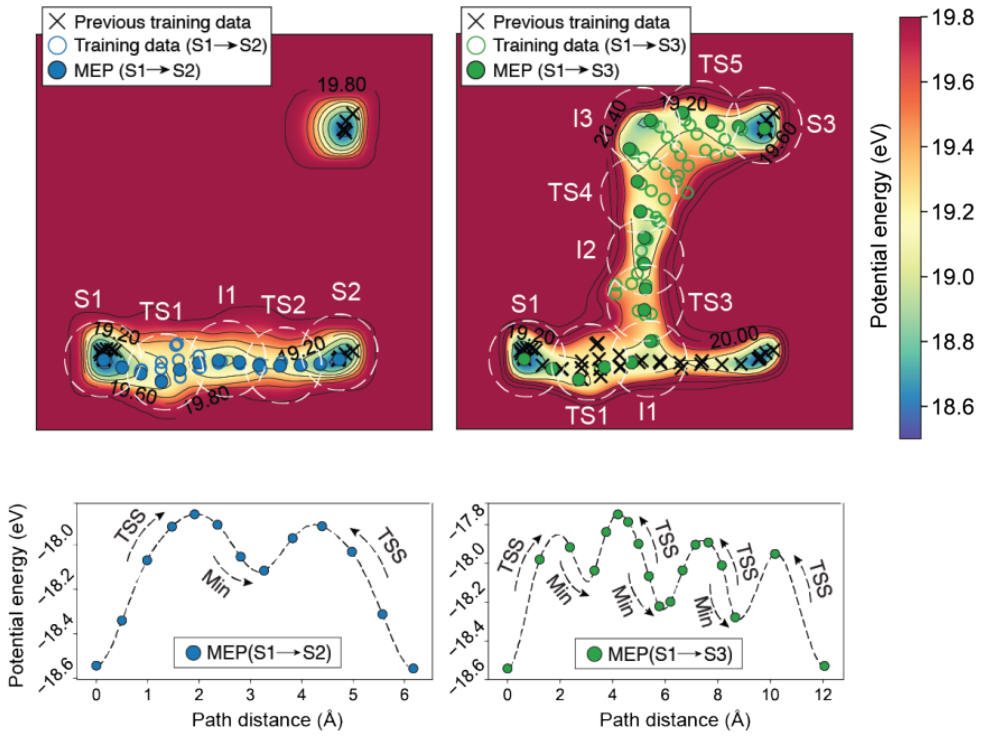


Figure 6.5: Example of data reuse between different active learning algorithms in the AID framework for the determination of a reaction network. The top panels show surrogate potential energy surface, obtained from the information gathered with the minimizations of the three structures (S1-S3) and the new data gathered with the new active learning methods at each stage of the calculation. Image adapted from *Paper III: An artificial intelligence-driven approach for the exploration of potential energy surfaces.*

6.4 Paper IV: Machine Learning with bond information for local structure optimizations in surface science

In this work, we address the problem of capturing the anisotropy of the potential energy surface in the context of local optimization with Gaussian process guidance. In addition, we explore the consequences of the limited memory Gaussian process calculator introduced in Paper III. As a result, we propose two novel local optimization methods based on the new Gaussian process surrogate model. We have termed *BondMin* to the new method with improved anisotropy description and *LBondMin* to the limited memory version of the former.

The summary of Paper IV is structured in two subsections: we first discuss the limited memory approach in connection with paper IV and then we explore how the inclusion of bond information in the model can result in the faster optimization of adsorbates on surfaces, as compared with the results presented in Paper II.

6.4.1 Limited memory Gaussian processes for local optimization

The computational time needed to perform a Gaussian process regression with gradient information scales cubically with the number of points in the training set and the number of force components, which is 3 times the number of atoms in the unit cell, $O(n^3(3N)^3)$. The memory scales quadratically with the same quantity. As a consequence, the relaxation of systems with many atoms in the unit cell poses a challenge for methods like GPMin.

In Paper III, we suggest two actions to reduce the computational cost:

1. *A reduction the number of force components used in the training.* In many systems, the optimization does not involve all the atoms in the unit cell, but only those that are not constrained.

2. *Training only on the n_{max} closest structures to the structure whose energy we want to predict.* These structures will be the most correlated ones and, hence, be responsible for the major part of the prediction for stationary kernels that decay with the distance.

The combination of these reduces the scaling in memory to $O(n_{\text{max}}^3 N_{\text{dof}}^3)$, where N_{dof} is the number of degrees of freedom. This way the memory no longer depends on the number of steps, but on a predefined n_{max} factor, which can be adjusted to match the computational resources available, and the prefactor on the number of force components is reduced.

We have tested the performance of these actions in the context of local optimization on two systems described with density functional theory: a 10 atom sodium cluster and a CO molecule on a fcc (100) platinum slab (also a system with 10 atoms in the unit cell).

The results are shown in Figure 6.6. We have found that, if n_{max} is sufficiently large (but still small compared to the number of steps), the performance of the optimizer with

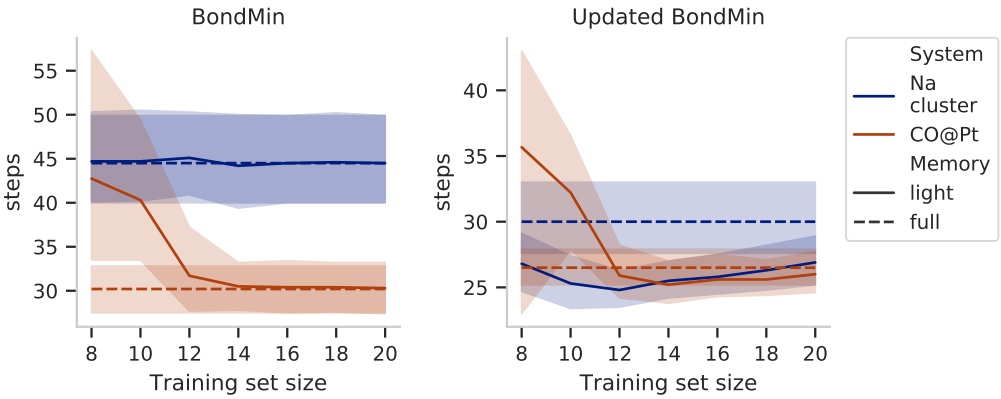


Figure 6.6: Comparison between the full memory and the limited memory approaches for the local optimization of the atomic structure of two atomic systems, with and without hyperparameter updates of the Gaussian process model along the way. The lines indicate the average number of steps over 10 runs and the shaded area, the 95% confidence interval of the mean estimated with bootstrapping. Source: *Paper IV: Machine Learning with bond information for local structure optimizations in surface science*

fixed hyperparameters is equivalent in the full memory and limited memory versions.

Surprisingly, if the hyperparameters are updated, there is a range of training set sizes for which the limited memory method is actually faster than the full memory version, with about 15% less DFT calculations. We think the reason for this speed up might be that the reduction of the training set size may lead to a higher flexibility of the model, by being able to have different hyperparameters to describe different regions. As a result, we believe that the surrogate model may describe the neighborhood of the minimum more accurately, leading to boost of the performance. We note that a speed up related to the use of a set of local Gaussian process regressions has also been reported by Eriksson *et al.* in the context of global optimization [142].

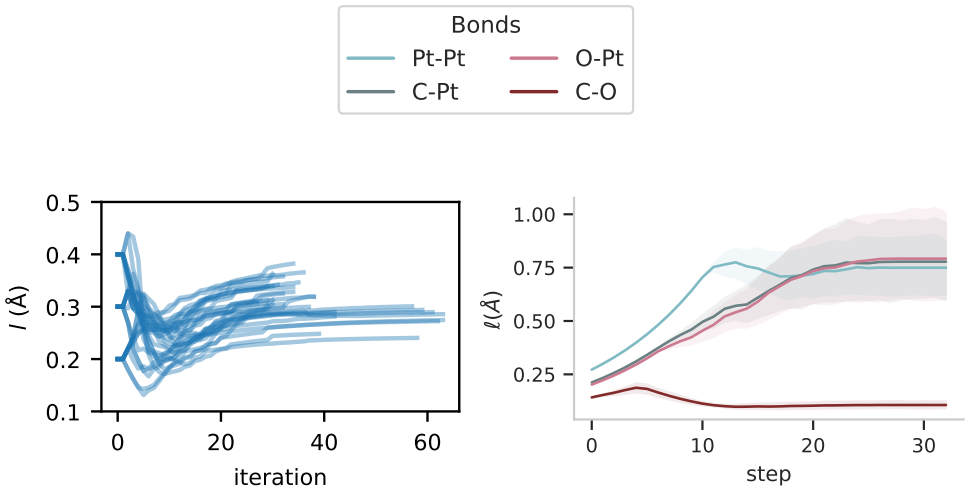
6.4.2 Including bond information to improve local optimization

We have noticed that the performance of the GPMin optimizer is reduced in those systems where the potential energy surface is anisotropic, for example, in systems involving molecules on surfaces. In this kind of systems the variation in the stiffness of the bonds produces large differences in the curvature of the potential energy surface around the minimum among different directions.

If the optimization is guided by an model that is *a priori* isotropic, it will need a larger training set to reproduce the anisotropy of the underlying potential energy surface. This observation has motivated us to introduce the following kernel between configurations ϱ and ϱ':

$$k(\varrho, \varrho') = k_0^2 e^{-d^2(\varrho,\varrho')/2\ell^2}, \tag{6.1}$$

where $d^2(\varrho, \varrho')$ is the squared distance between the two configurations, defined as:



(a) GPMin (b) Bondmin

Figure 6.7: Evolution of the scale in the Gaussian process regression surrogate model as the local optimization progresses. The atomic system studied here is a CO molecule on a fcc metallic slab (made of gold in panel (a) and platinum in panel (b)). Panel (a) shows the evolution of the scale of an isotropic kernel. Three initial scales are considered (0.2, 0.3 and 0.4 Å) and 10 runs are executed for each initial value. Each line marks the evolution of an individual run. Panel (b) shows the evolution of the scale for an anisotropic kernel that has one scale for each pair of atomic species. The initial value of each scale is one fifth of the average of the covalent radii [82]. The solid lines represent the average scale over 10 runs and the shaded area represents the 95% confidence interval of the mean estimated with bootstrapping. Adapted from Paper II: Local Bayesian optimizer for atomic structures and Paper III: Machine Learning with bond information for local structure optimizations in surface science.

$$d^2(\varrho, \varrho') = \frac{1}{N} \sum_{i,j}^{N} \frac{\| \left(\mathbf{R}_i - \mathbf{R}_j\right) - \left(\mathbf{R}'_i - \mathbf{R}'_j\right) \|^2}{\ell^2_{X_i X_j}}, \tag{6.2}$$

where $\mathbf{R}_i$ is the coordinate of the $i-$th atom, X_i is the atomic species of the $i-$th atom, and the double sum runs over the number of atoms in the unit cell N. Note that $\mathbf{b}_{ij} = \mathbf{R}_i - \mathbf{R}_j$ is the vector along the bond between atoms i and j, and thus, the distance measure in equation (6.2) is a weighted distance over atomic bonds. Since all the atoms are included, the kernel in equation (6.1) is equivalent to the anisotropic squared exponential kernel (5.12) introduced in Section 5.2, where the matrix M is the Laplacian matrix of the fully connected graph [143] where the nodes are the atoms in the unit cell. It is easy to see that kernel (6.1) is invariant under rigid translations of all the atoms in the unit cell, and that it does not capture any other symmetry of the system.

This kernel uses a different scale $\ell_{X_i X_j}$ for each pair of atomic species X_i and X_j. It is well known that certain combinations of atomic species tend to form stiffer bonds than others. It is possible to incorporate this prior knowledge into the kernel by using the covalent radii as tabulated by Cordero *et al.* [82], so that the PES is anisotropic *a priori*. These scales can be further updated as the optimization progresses, using the data in the training set to find the scales that describe the potential energy surface of the system best.

Figure 6.7 shows the value of the scale of the kernel that optimizes the marginal likelihood (5.16) as the optimization progresses. Both panels show similar systems, a CO molecule on a fcc transition metal surface. The optimization in subfigure 6.7(a) uses GPMin as an optimizer, which relies on an isotropic model of the PES, while the optimization in subfigure (b) uses BondMin as an optimizer, which uses the equations (6.1) and (6.2) as a model for the kernel. The BondMin optimizer achieves to optimize the system in less steps (all optimizations have converged within 30 steps), by diversifying the scale profile for different directions. The scale for the CO bond stays below 0.2 Å during all the optimization, finishes at an average value of 0.11 Å, while the metal-metal and metal-molecule bond scales grow to reach about 0.77 Å on average. The CO scale is much shorter than the average scale for the isotropic GPMin (about 0.3 Å) and the others are significantly longer, allowing BondMin to take longer steps without perturbing the molecule too much. As a result, the BondMin optimizer needs less steps to optimize structures with very different levels of stiffness across bonds.

This result is farther illustrated in Figure 6.8. We have studied the optimization of 5 different molecules and radicals on different fcc (100) surfaces. For each system, we have studied three different template initial configurations (on top, hollow and bridge), which we have used to create 10 slightly different configurations from each template by adding small Gaussian rattling. All the resulting systems have been relaxed with BFGS Line Search, as implemented in ASE, GPMin and Bondmin.

We observe that for these systems with anisotropic potential energy surfaces, GPMin does not show a consistent speed up as compared to BFGS, but BondMin achieves reductions of the number of DFT calculations of up to a factor 2 as compared to BFGS. This effect is particularly significant if the hyperparameters are allowed to update. We also observe that the reduction in the number of steps is larger for for those systems where the total number of steps needed by BFGS is large. Furthermore, we also observe that the spread of the number of steps over different runs is smaller for BondMin than for other optimizers.

From these observations we conclude that the BondMin optimizer is faster and more robust than the methods we have compared it against for the optimization of adsorption systems, which represented a challenge to GPMin.

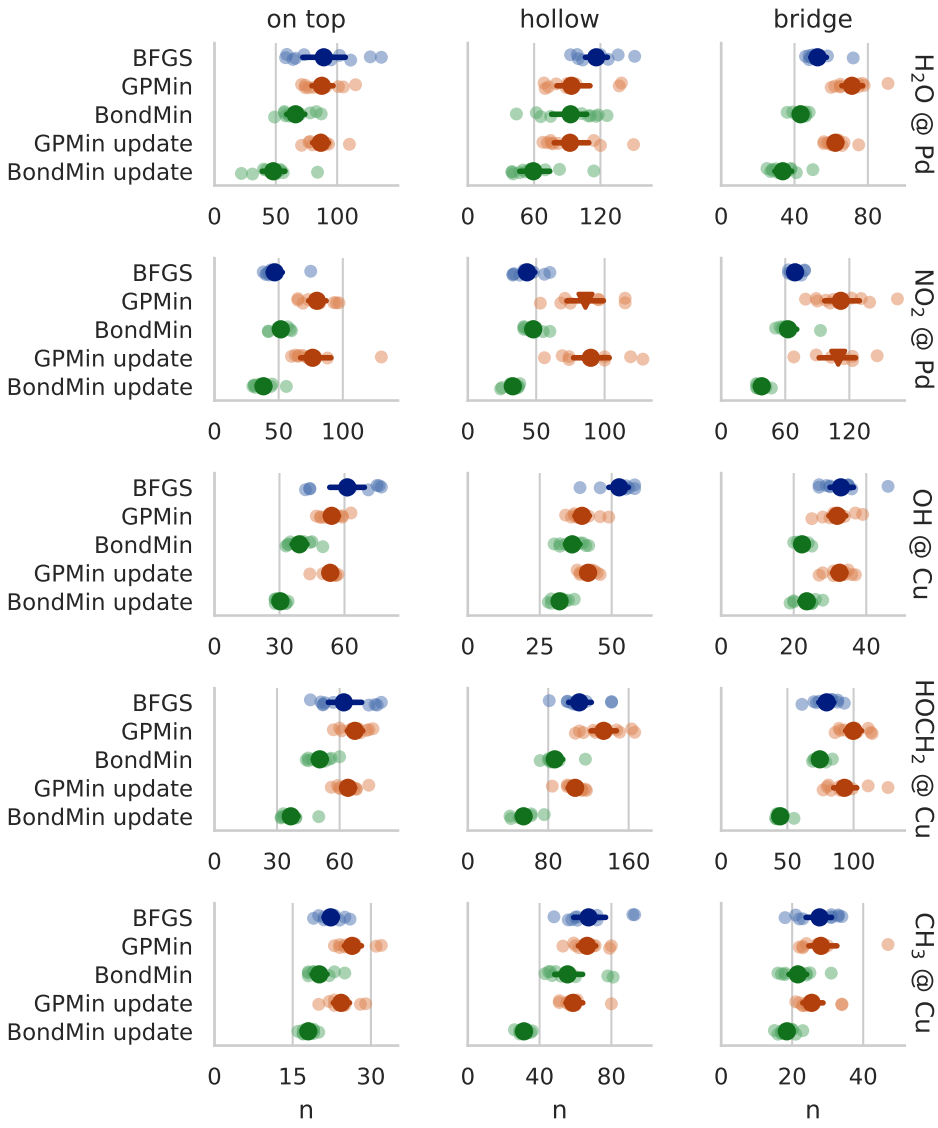


Figure 6.8: Number of DFT calculations needed to relax some molecules and radicals on surfaces with different local optimization methods. The light circles indicate the number of DFT calculations needed to find the minimum structure for ten runs with slightly different initial structures and the dark marker indicates the average over the 10 runs. The circle indicates that all 10 runs succeeded, while the inverted triangle marks that at least one of the run failed. Adapted from : *Paper III: Machine Learning with bond information for local structure optimizations in surface science.*

6.5 Paper V: Global optimization of atomic structures with gradient-enhanced Gaussian process regression

In this paper we present BEACON, a Bayesian optimization method for the global optimization of atomic structures. The method is based on the original work by Bisbo and Hammer [114, 115], but it incorporates the gradient information into the Gaussian regression model.

In order to identify the different local minima that are connected by a symmetry operation with as few DFT calculations as possible, it is of great importance to have a machine learning method that is able to describe the symmetries of the atomic structures. To this end, we have used the modified Oganov fingerprint which includes an angular contribution in addition to the original radial fingerprint (for a detailed description, see Section 3.4.2). We have used the squared exponential kernel with this fingerprint as input and a repulsive potential to avoid the method to sample configurations where the two atoms are very close, as described in Section 5.3.

Figure 6.9 illustrates the optimization procedure for a Ta_6O_{15} cluster as described by DFT. The optimization starts by randomly generating a few atomic structures and obtaining their energies and forces. These configurations are then used to obtain a surrogate model of the potential energy surface, with its corresponding uncertainty estimation. Then, a set of new candidates is randomly generated and relaxed on the average prediction. We then use the lower confidence bound acquisition function (4.9) described in Section 4.2.1 to select the structure with the best balance between exploitation and exploration. The hyperparameters are optimized along the way, as the optimization progresses.

The predicted energy, the predicted uncertainty and the DFT energy of the configurations that the optimization method samples are shown in the top panel of Figure 6.9. The evolution of the optimization can be roughly divided into two phases: a first phase where the energy is minimized until the global minimum is found and a second phase where the method explores the PES to be certain no other minimum has been missed.

During the first part of the run, the optimization method finds lower lying structures as the the optimization progresses, with the exception of some high uncertainty structures. As more information is collected, the hyperparameters are optimized, improving the description of the PES. The geometry of the cluster evolves in parallel, identifying the right coordination of the atoms as more data is collected. The global minimum is identified at approximately step 40. Since the optimizer is forced to continue after identifying the global minimum, it keeps sampling structures, first around the global minimum and then in a different basin. It is interesting to see that between iterations 60 and 80, the Gaussian process regression mistakenly predicts that there are structures whose energy is lying below the minimum, but by gathering more data, the method is able to correct its hyperparameters and to predict average energies above the global minimum again. The optimizer then continues to explore regions with high uncertainty to make sure a lower lying basin has not been missed.

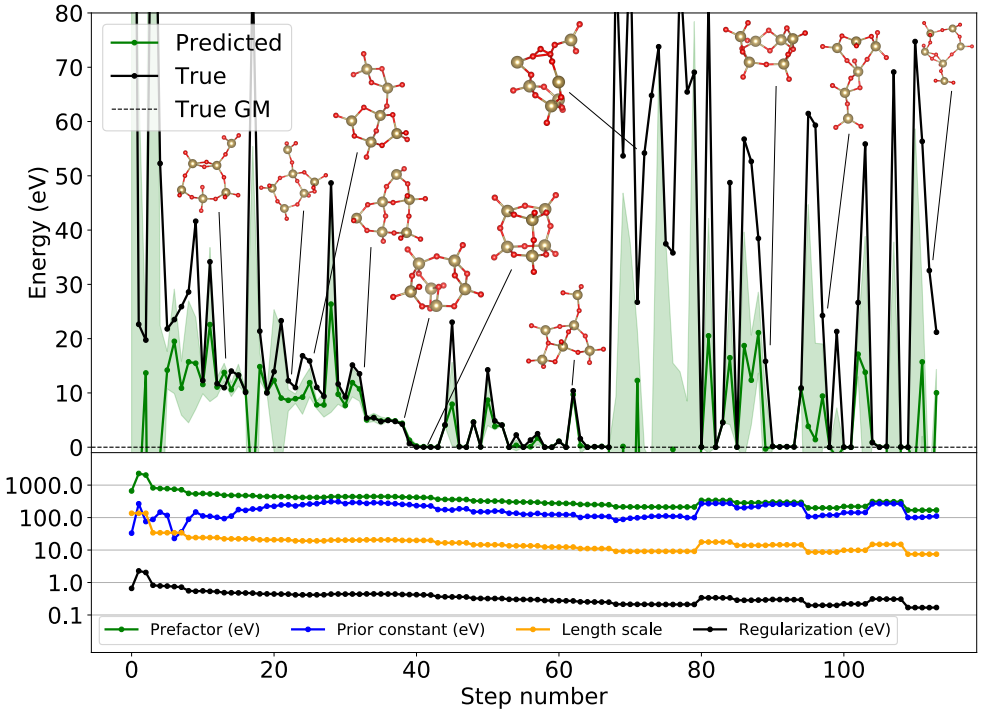


Figure 6.9: Evolution of the potential energy during global optimization of a Ta_6O_{15} cluster and of the hyperparameters of the surrogate model. The green curve in the top model shows the predicted energies and the black curve, the DFT values. The potential energy of the cluster initially converges towards the global minimum, by learning the adequate coordination number as the optimization progresses, until the uncertainty on the prediction is very small and the global minimum is reached (around step 40). In a second phase, the method explores points with high uncertainty and low energy, in an attempt to find an structure with a lower energy. Source *Paper V: Global optimization of atomic structures with gradient-enhanced Gaussian process regression.*

We have found that including the forces in the Gaussian process regression model leads to a reduction of necessary DFT calculation of about a 4-fold as compared to GOFEE [114], in addition to the order of magnitude reduction that GOFEE presents with respect to evolutionary strategies [114]. We have illustrated this by applying the different methods to a copper cluster described with EMT.

We have successfully applied BEACON to find the global minimum of a number of systems, demonstrating the utility of the method. We have used BEACON on the $[Ta_2O_5]_x$ clusters with $x = 1, 2, 3$, as demonstrated in Figure 6.9. We have also used it to determine the structure of the oxidized ZrN surface, where we find that the global minimum structure is the one having an oxygen and a nitrogen atoms in the surface (of a 4 by 4 slab) and a nitrogen vacancy below the oxygen atom. This structure, surprisingly, turns out to be more stable than a the structure that has the oxygen lying on the surface, adsorbed in the hollow position, and no nitrogen vacancy. We believe that this result demonstrates the utility of the global optimization method we have presented to guide research in surface science.

7 Conclusions

In this thesis, we have shown how machine learning models can successfully reduce the computation time necessary to identify the atomic configurations of interest for a given system. We have shown how, without loosing the accuracy of quantum chemistry methods such as density functional theory, it is possible to guide optimization searches by using data driven models.

The machine learning models presented in this thesis can guide searches for local and global minima, as well as transition states. The methods presented, rather than a single-use model for a specific material, constitute a framework that allows to build good models for systems of different compositions. We have shown that the accuracy of all the models presented can be improved by adding more configurations to the training set, thus improving the guidance for the PES exploration.

We have demonstrated that reusing "old" data improves the search. Pre-training on OQMD [29] improves the prediction of the global minimum prototype in screening studies involving bulk materials (as we have demonstrated in Paper I) and reusing the trajectories of previous local optimizations and transition state searches speeds up other local optimizations and transition state searches during the determination of reaction networks (as we have seen in Paper III).

Another guideline that arises from the results is that the inclusion of more "physical" intuition into the prior knowledge of the model reduces the amount of data needed to achieve the desired accuracy. Including the symmetry label in the graph descriptors (Paper I) or including information about the covalent radii of atoms (Paper IV) improves the accuracy of the parts of the PES we are interested in. At a more fundamental level, the results show that the symmetry of a material becomes an important piece of information in the characterization of the different local minima. Thus, the inclusion of symmetry information in the description of an atomic system is key for the identification of the global minimum structure (see Papers I and V).

All the active learning methods based on Gaussian processes presented in this thesis have been collected in the Python package *gpatom* [144] in a ready-to-use form. Those hyperparameters and design choices that lead to the best performance over many different training examples have been selected as default choices in the package. The result of these methods is always a structure whose energy and forces have been obtained by a quantum chemistry method (which is chosen by the user among those available through ASE [16, 17]). Thus, the package is addressed to an audience of both experts and non-experts in machine learning, which can benefit of an speed up in the determination of the geometry of atomic structures.

Bibliography

[1] Anubhav Jain. *The scale of materials design.* Hacking Materials: https://hackingmaterials.com/2013/11/14/the-scale-of-materials-design/. Accessed: 2020-11-20. 2013.

[2] Mark Miodownik. *Stuff matters: Exploring the marvelous materials that shape our man-made world*. Houghton Mifflin Harcourt, 2014.

[3] Richard E. Smalley. "Discovering the fullerenes (Nobel lecture)". In: *Angewandte Chemie International Edition in English* 36.15 (1997), pp. 1594–1601.

[4] Kostya S Novoselov et al. "Electric field effect in atomically thin carbon films". In: *science* 306.5696 (2004), pp. 666–669.

[5] Yuan Cao et al. "Unconventional superconductivity in magic-angle graphene superlattices". In: *Nature* 556.7699 (2018), pp. 43–50.

[6] Geoffroy Hautier, Anubhav Jain, and Shyue Ping Ong. "From the computer to the laboratory: materials discovery and design using first-principles calculations". In: *Journal of Materials Science* 47.21 (2012), pp. 7317–7340.

[7] Anubhav Jain, Yongwoo Shin, and Kristin A Persson. "Computational predictions of energy materials using density functional theory". In: *Nature Reviews Materials* 1.1 (2016), pp. 1–13.

[8] P Hohenberg and W Kohn. "Inhomogeneous Electron Gas". In: *Physical Review* 136.3 (Nov. 1964), pp. 864–871.

[9] W Kohn and L J Sham. "Self-Consistent Equations Including Exchange and Correlation Effects". In: *Physical Review* 140.4 (Nov. 1965), pp. 1133–1138.

[10] Karsten Wedel Jacobsen, Jens K Nørskov, and M. J. Puska. "Interatomic interactions in the effective-medium theory". In: *Physical Review B* 35.1 (May 1987), pp. 7423–7442.

[11] Baron Peters. "Chapter 7 - Potential energy surfaces and dynamics". In: *Reaction Rate Theory and Rare Events Simulations*. Ed. by Baron Peters. Amsterdam: Elsevier, 2017, pp. 157–182.

[12] H. Bernhard Schlegel. "Geometry optimization". In: *WIREs Computational Molecular Science* 1.5 (2011), pp. 790–809.

[13] Yousef Saad, James R Chelikowsky, and Suzanne M Shontz. "Numerical methods for electronic structure calculations of materials". In: *SIAM review* 52.1 (2010), pp. 3–54.

[14] Chris J. Pickard and R.J. Needs. "Ab initio random structure searching". In: *Journal of Physics: Condensed Matter* 23.5 (2011), p. 053201.

[15] Jorge Kohanoff. *Electronic structure calculations for solids and molecules: theory and computational methods*. Cambridge University Press, 2006.

[16] *Atomic Simulation Environment (ASE)*. https://wiki.fysik.dtu.dk/ase/. 2018.

[17] Ask Hjorth Larsen, Jens Jørgen Mortensen, et al. "The atomic simulation environment - a Python library for working with atoms". In: *Journal of Physics: Condensed Matter* 29.27 (2017), p. 273002.

[18] J Enkovaara, C Rostgaard, J J Mortensen, et al. “Electronic structure calculations with GPAW: a real-space implementation of the projector augmented-wave method”. In: *Journal of Physics: Condensed Matter* 22.25 (2010), p. 253202.
[19] Georg Kresse and Jürgen Furthmüller. “Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set”. In: *Physical review B* 54.16 (1996), p. 11169.
[20] Georg Kresse and D. Joubert. “From ultrasoft pseudopotentials to the projector augmented-wave method”. In: *Physical Review B* 59.3 (1999), p. 1758.
[21] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. “Generalized Gradient Approximation Made Simple”. In: *Phys. Rev. Lett.* 77 (18 Oct. 1996), pp. 3865–3868.
[22] John P. Perdew et al. “Restoring the Density-Gradient Expansion for Exchange in Solids and Surfaces”. In: *Phys. Rev. Lett.* 100 (13 Apr. 2008), p. 136406.
[23] B. Hammer, L. B. Hansen, and J. K. Nørskov. “Improved adsorption energetics within density-functional theory using revised Perdew-Burke-Ernzerhof functionals”. In: *Phys. Rev. B* 59 (11 Mar. 1999), pp. 7413–7421.
[24] A. H. Larsen et al. “Localized atomic basis set in the projector augmented wave method”. In: *Phys. Rev. B* 80 (19 Nov. 2009), p. 195112.
[25] Fernando Nogueira, Alberto Castro, and Miguel AL Marques. “A tutorial on density functional theory”. In: *A Primer in Density Functional Theory*. Springer, 2003, pp. 218–256.
[26] P. E. Blöchl. “Projector augmented-wave method”. In: *Phys. Rev. B* 50 (24 Dec. 1994), pp. 17953–17979.
[27] José M. Soler et al. “The SIESTA method for ab initio order-N materials simulation”. In: *Journal of Physics: Condensed Matter* 14.11 (2002), p. 2745.
[28] C. J. García-Cervera et al. “Linear-scaling subspace-iteration algorithm with optimally localized nonorthogonal wave functions for Kohn-Sham density functional theory”. In: *Phys. Rev. B* 79 (11 Mar. 2009), p. 115110.
[29] James E Saal et al. “Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD)”. In: *Jom* 65.11 (2013), pp. 1501–1509.
[30] Scott Kirklin et al. “The Open Quantum Materials Database (OQMD): assessing the accuracy of DFT formation energies”. In: *npj Computational Materials* 1.1 (2015), pp. 1–15.
[31] Stefano Curtarolo et al. “AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations”. In: *Computational Materials Science* 58 (2012), pp. 227–235. ISSN: 0927-0256.
[32] Anubhav Jain et al. “The Materials Project: A materials genome approach to accelerating materials innovation”. In: *APL Materials* 1.1 (2013), p. 011002.
[33] NOMAD. *The NOMAD repository*. http://nomad-repository.eu; https://nomad-coe.eu. Accessed: 2020-11-20. 2020.
[34] David D Landis et al. “The computational materials repository”. In: *Computing in Science & Engineering* 14.6 (2012), pp. 51–57.
[35] Sten Haastrup et al. “The Computational 2D Materials Database: high-throughput modeling and discovery of atomically thin crystals”. In: *2D Materials* 5.4 (Sept. 2018), p. 042002.

[36] M. N. Gjerding et al. *Recent Progress of the Computational 2D Materials Database (C2DB)*. 2021. arXiv: 2102.03029.
[37] Kristian S Thygesen and Karsten W Jacobsen. “Making the most of materials computations”. In: *Science* 354.6309 (2016), pp. 180–181.
[38] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.
[39] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.
[40] Ian Goodfellow et al. *Deep learning*. Vol. 1. MIT press Cambridge, 2016.
[41] Thomas M. Mitchell. *Machine Learning*. 1st ed. USA: McGraw-Hill, Inc., 1997.
[42] Peter Bjørn Jørgensen and Arghya Bhowmik. *DeepDFT: Neural Message Passing Network for Accurate Charge Density Prediction*. 2020. arXiv: 2011.03346 [physics.comp-ph].
[43] Peter Bjørn Jørgensen, Karsten Wedel Jacobsen, and Mikkel N Schmidt. “Neural message passing with edge updates for predicting properties of molecules and materials”. In: *arXiv preprint arXiv:1806.03146* (2018).
[44] Peter Bjørn Jørgensen et al. “Materials property prediction using symmetry-labeled graphs as atomic position independent descriptors”. In: *Phys. Rev. B* 100 (10 Sept. 2019), p. 104114.
[45] K. T. Schütt et al. “SchNet – A deep learning architecture for molecules and materials”. In: *The Journal of Chemical Physics* 148.24 (2018), p. 241722.
[46] Jens Jørgen Mortensen et al. “Bayesian error estimation in density-functional theory”. In: *Physical review letters* 95.21 (2005), p. 216401.
[47] Jess Wellendorff et al. “Density functionals for surface science: Exchange - correlation model development with Bayesian error estimation”. In: *Physical Review B* 85.23 (2012), p. 235149.
[48] Jess Wellendorff et al. “mBEEF: An accurate semi-local Bayesian error estimation density functional”. In: *The Journal of Chemical Physics* 140.14 (2014), p. 144107.
[49] Ryo Nagai, Ryosuke Akashi, and Osamu Sugino. “Completing density functional theory by machine learning hidden messages from molecules”. In: *npj Computational Materials* 6.1 (2020), pp. 1–8.
[50] Ivano Eligio Castelli and Karsten Wedel Jacobsen. “Designing rules and probabilistic weighting for fast materials discovery in the Perovskite structure”. In: *Modelling and Simulation in Materials Science and Engineering* 22.5 (2014), p. 055007.
[51] Peter Bjørn Jørgensen et al. “Machine learning-based screening of complex molecules for polymer solar cells”. In: *The Journal of chemical physics* 148.24 (2018), p. 241735.
[52] Stefan Chmiela et al. “Machine learning of accurate energy-conserving molecular force fields”. In: *Science advances* 3.5 (2017), e1603015.
[53] Claudio Zeni et al. “Building machine learning force fields for nanoclusters”. In: *The Journal of chemical physics* 148.24 (2018), p. 241739.
[54] Igor Levin. *Inorganic Crystal Structure Database (ICSD)*. http://https://icsd.fiz-karlsruhe.de. 2018.

[55] K.-R. Müller et al. “A Numerical Study on Learning Curves in Stochastic Multilayer Feedforward Networks”. In: *Neural Computation* 8.5 (1996), pp. 1085–1106.
[56] Bing Huang and O. Anatole von Lilienfeld. “Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity”. In: *The Journal of Chemical Physics* 145.16 (2016), p. 161102.
[57] Bing Huang and O. Anatole von Lilienfeld. *Ab initio machine learning in chemical compound space*. arXiv 2012.07502. 2020.
[58] Aldo Glielmo et al. “Building nonparametric n-body force fields using gaussian process regression”. In: *Machine Learning Meets Quantum Physics*. Springer, 2020, pp. 67–98.
[59] Aldo Glielmo, Peter Sollich, and Alessandro De Vita. “Accurate interatomic force fields via machine learning with covariant kernels”. In: *Physical Review B* 95.21 (2017), p. 214302.
[60] Andrea Grisafi et al. “Symmetry-Adapted Machine Learning for Tensorial Properties of Atomistic Systems”. In: *Phys. Rev. Lett.* 120 (3 Jan. 2018), p. 036002.
[61] Tian Xie and Jeffrey C. Grossman. “Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties”. In: *Phys. Rev. Lett.* 120 (14 Apr. 2018), p. 145301.
[62] Chi Chen et al. “Graph networks as a universal machine learning framework for molecules and crystals”. In: *Chemistry of Materials* 31.9 (2019), pp. 3564–3572.
[63] Ankit Jain and Thomas Bligaard. “Atomic-position independent descriptor for machine learning of material properties”. In: *Phys. Rev. B* 98 (21 Dec. 2018), p. 214112.
[64] Christopher K. I. Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.
[65] William H Press et al. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd ed. New York, NY, USA: Cambridge University Press, 2007.
[66] Felix Faber et al. “Crystal structure representations for machine learning models of formation energies”. In: *International Journal of Quantum Chemistry* 115.16 (), pp. 1094–1101.
[67] Haoyan Huo and Matthias Rupp. *Unified Representation of Molecules and Crystals for Machine Learning*. arXiv 1704.06439. 2018. arXiv: 1704.06439 [physics.chem-ph].
[68] Albert P Bartók, Risi Kondor, and Gábor Csányi. “On representing chemical environments”. In: *Physical Review B* 87.18 (2013), p. 184115.
[69] Lauri Himanen et al. “DScribe: Library of descriptors for machine learning in materials science”. In: *Computer Physics Communications* 247 (2020), p. 106949.
[70] Matthias Rupp et al. “Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning”. In: *Phys. Rev. Lett.* 108 (5 Jan. 2012), p. 058301.
[71] Katja Hansen et al. “Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space”. In: *The journal of physical chemistry letters* 6.12 (2015), pp. 2326–2331.

[72] Mario Valle and Artem R. Oganov. “Crystal fingerprint space – a novel paradigm for studying crystal-structure sets”. In: *Acta Crystallographica Section A* 66.5 (2010), pp. 507–517.
[73] Jörg Behler. “Atom-centered symmetry functions for constructing high-dimensional neural network potentials”. In: *The Journal of Chemical Physics* 134.7 (2011), p. 074106.
[74] Behnam Parsaeifard et al. “An assessment of the structural resolution of various fingerprints commonly used in machine learning”. In: *Machine Learning: Science and Technology* (2020).
[75] Yunxing Zuo et al. “Performance and cost assessment of machine learning interatomic potentials”. In: *The Journal of Physical Chemistry A* 124.4 (2020), pp. 731–745.
[76] Logan Ward et al. “Including crystal structure attributes in machine learning models of formation energies via Voronoi tessellations”. In: *Phys. Rev. B* 96 (2 July 2017), p. 024104.
[77] Cheol Woo Park and Chris Wolverton. “Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery”. In: *Phys. Rev. Materials* 4 (6 June 2020), p. 063801.
[78] E. Wigner and F. Seitz. “On the Constitution of Metallic Sodium”. In: *Phys. Rev.* 43 (10 May 1933), pp. 804–810.
[79] N.W. Ashcroft, N.D Mermin, and D. Wei. *Solid State Physics: Revised Edition*. CENGAGE Learning, 2016.
[80] Alex Malins et al. “Identification of structure in condensed matter with the topological cluster classification”. In: *The Journal of Chemical Physics* 139.23 (2013), p. 234506.
[81] Olexandr Isayev et al. “Universal fragment descriptors for predicting properties of inorganic crystals”. In: *Nature communications* 8.1 (2017), pp. 1–12.
[82] Beatriz Cordero et al. “Covalent radii revisited”. In: *Dalton Trans.* (21 2008), pp. 2832–2838.
[83] S. J. Chung, Th. Hahn, and W. E. Klee. “Nomenclature and generation of three-periodic nets: the vector method”. In: *Acta Crystallographica Section A* 40.1 (Jan. 1984), pp. 42–50.
[84] W. E. Klee. “Crystallographic nets and their quotient graphs”. In: *Crystal Research and Technology* 39.11 (2004), pp. 959–968.
[85] Jun Zhang and Vassiliki-Alexandra Glezakou. “Global optimization of chemical cluster structures: Methods, applications, and challenges”. In: *International Journal of Quantum Chemistry* 121.7 (2021), e26553.
[86] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
[87] Eric D Hermes et al. “Accelerated saddle point refinement through full exploitation of partial Hessian diagonalization”. In: *Journal of chemical theory and computation* 15.11 (2019), pp. 6536–6549.
[88] Josep Maria Bofill. “Updated Hessian matrix and the restricted step method for locating transition structures”. In: *Journal of Computational Chemistry* 15.1 (1994), pp. 1–11.

[89] Ödön Farkas and H Bernhard Schlegel. “Methods for optimizing large molecules. II. Quadratic search”. In: *The Journal of Chemical Physics* 111.24 (1999), pp. 10806–10814.
[90] Mykel J Kochenderfer and Tim A Wheeler. *Algorithms for optimization*. The MIT Press, 2019.
[91] Ya-xiang Yuan. “Recent advances in trust region algorithms”. In: *Mathematical Programming* 151.1 (2015), pp. 249–281.
[92] Jorge Nocedal. “Theory of algorithms for unconstrained optimization”. In: *Acta numerica* 1.1 (1992), pp. 199–242.
[93] G. Yuan, Z. Wei, and X. Lu. “Global convergence of BFGS and PRP methods under a modified weak Wolfe–Powell line search”. In: *Applied Mathematical Modelling* 47 (2017), pp. 811–825.
[94] G. Yuan et al. “The global convergence of a modified BFGS method for nonconvex functions”. In: *Journal of Computational and Applied Mathematics* 327 (2018), pp. 274–294.
[95] Hannes Jónsson, Greg Mills, and Karsten W. Jacobsen. “Nudged elastic band method for finding minimum energy paths of transitions”. In: *Classical and Quantum Dynamics in Condensed Phase Simulations*. Ed. by Bruce J Berne, Giovanni Ciccotti, and David F Coker. Singapore: World Scientific, 1998, pp. 385–404.
[96] J. H. A. Hagelaar et al. “Atomistic simulations of the formation and destruction of nanoindentation contacts in tungsten”. In: *Phys. Rev. B* 73 (4 Jan. 2006), p. 045425.
[97] Erik Bitzek et al. “Structural Relaxation Made Simple”. In: *Phys. Rev. Lett.* 97 (17 Oct. 2006), p. 170201.
[98] David Packwood et al. “A universal preconditioner for simulating condensed phase materials”. In: *The Journal of Chemical Physics* 144.16 (2016), p. 164109.
[99] Letif Mones, Christoph Ortner, and Gábor Csányi. “Preconditioners for the geometry optimisation and saddle point search of molecular systems”. In: *Scientific reports* 8.1 (2018), pp. 1–11.
[100] Artem R. Oganov and Colin W. Glass. “Crystal structure prediction using ab initio evolutionary techniques: Principles and applications”. In: *The Journal of Chemical Physics* 124.24 (2006), p. 244704.
[101] Artem R. Oganov. “Crystal structure prediction: reflections on present status and challenges”. In: *Faraday Discuss.* 211 (0 2018), pp. 643–660.
[102] Bernd Hartke. “Global optimization”. In: *WIREs Computational Molecular Science* 1.6 (2011), pp. 879–887.
[103] Sean Luke. *Essentials of Metaheuristics*. second. Available for free at http://cs.gmu.edu/~sean/book/metaheuristics/. Lulu, 2013.
[104] Lasse B. Vilhelmsen and Bjørk Hammer. “A genetic algorithm for first principles global structure optimization of supported nano structures”. In: *The Journal of Chemical Physics* 141.4 (2014), p. 044711.
[105] Steen Lysgaard et al. “Genetic algorithm procreation operators for alloy nanoparticle catalysts”. In: *Topics in Catalysis* 57.1-4 (2014), pp. 33–39.

[106] Marc Jäger, Rolf Schäfer, and Roy L. Johnston. “GIGA: a versatile genetic algorithm for free and supported clusters and nanoparticles in the presence of ligands”. In: *Nanoscale* 11 (18 2019), pp. 9042–9052.
[107] Yanchao Wang et al. “CALYPSO: A method for crystal structure prediction”. In: *Computer Physics Communications* 183.10 (2012), pp. 2063–2070.
[108] David J. Wales and Jonathan P. K. Doye. “Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms”. In: *The Journal of Physical Chemistry A* 101.28 (1997), pp. 5111–5116.
[109] Mathias S Jørgensen et al. “Exploration versus exploitation in global atomistic structure optimization”. In: *The Journal of Physical Chemistry A* 122.5 (2018), pp. 1504–1509.
[110] B. Shahriari et al. “Taking the Human Out of the Loop: A Review of Bayesian Optimization”. In: *Proceedings of the IEEE* 104.1 (2016), pp. 148–175.
[111] Eric Brochu, Vlad M. Cora, and Nando de Freitas. *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*. 2010. arXiv: 1012.2599.
[112] Shane Carr, Roman Garnett, and Cynthia Lo. “BASC: Applying Bayesian Optimization to the Search for Global Minima on Potential Energy Surfaces”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 898–907.
[113] Tomoki Yamashita et al. “Crystal structure prediction accelerated by Bayesian optimization”. In: *Phys. Rev. Materials* 2 (1 Jan. 2018), p. 013803.
[114] Malthe K Bisbo and Bjørk Hammer. “Efficient Global Structure Optimization with a Machine-Learned Surrogate Model”. In: *Physical Review Letters* 124.8 (2020), p. 086102.
[115] Malthe K. Bisbo and Bjørk Hammer. “Global optimization of atomistic structure enhanced by machine learning”. In: (2020). arXiv: 2012.15222.
[116] Milica Todorović et al. “Bayesian inference of atomistic structure in functional materials”. In: *Npj computational materials* 5.1 (2019), pp. 1–7.
[117] Lincan Fang et al. “Efficient Amino Acid Conformer Search with Bayesian Optimization”. In: *Journal of Chemical Theory and Computation* (2021).
[118] Jack Simons et al. “Walking on potential energy surfaces”. In: *The Journal of Physical Chemistry* 87.15 (1983), pp. 2745–2753.
[119] Ajit Banerjee et al. “Search for stationary points on surfaces”. In: *The Journal of Physical Chemistry* 89.1 (1985), pp. 52–57.
[120] Rachid Malek and Normand Mousseau. “Dynamics of Lennard-Jones clusters: A characterization of the activation-relaxation technique”. In: *Phys. Rev. E* 62 (6 Dec. 2000), pp. 7723–7728.
[121] R. A. Olsen et al. “Comparison of methods for finding saddle points without knowledge of the final states”. In: *The Journal of Chemical Physics* 121.20 (2004), pp. 9776–9792.

[122] Graeme Henkelman and Hannes Jónsson. “A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives”. In: *The Journal of Chemical Physics* 111.15 (1999), pp. 7010–7022.
[123] Graeme Henkelman, Blas P Uberuaga, and Hannes Jónsson. “A climbing image nudged elastic band method for finding saddle points and minimum energy paths”. In: *The Journal of chemical physics* 113.22 (2000), pp. 9901–9904.
[124] Andrew A. Peterson. “Acceleration of saddle-point searches with machine learning”. In: *The Journal of Chemical Physics* 145.7 (2016), p. 074106.
[125] Olli-Pekka Koistinen et al. “Nudged elastic band calculations accelerated with Gaussian process regression”. In: *The Journal of Chemical Physics* 147.15 (2017), p. 152720.
[126] Olli-Pekka Koistinen et al. “Nudged elastic band calculations accelerated with Gaussian process regression based on inverse interatomic distances”. In: *Journal of chemical theory and computation* 15.12 (2019), pp. 6738–6751.
[127] José A. Garrido Torres et al. “Low-Scaling Algorithm for Nudged Elastic Band Calculations Using a Surrogate Machine Learning Model”. In: *Phys. Rev. Lett.* 122 (15 Apr. 2019), p. 156001.
[128] Alexander Denzel and Johannes Kästner. “Gaussian Process Regression for Transition State Search”. In: *Journal of Chemical Theory and Computation* 14.11 (2018), pp. 5777–5786.
[129] Alexander Denzel and Johannes Kästner. “Hessian Matrix Update Scheme for Transition State Search Based on Gaussian Process Regression”. In: *Journal of Chemical Theory and Computation* 16.8 (2020), pp. 5083–5089.
[130] Jian Wu et al. “Bayesian optimization with gradients”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 5267–5278.
[131] Albert P. Bartók and Gábor Csányi. “Gaussian approximation potentials: A brief tutorial introduction”. In: *International Journal of Quantum Chemistry* 115.16 (2015), pp. 1051–1057.
[132] Anders S Christensen and O Anatole von Lilienfeld. “On the role of gradients for machine learning of molecular energies and forces”. In: *Machine Learning: Science and Technology* 1.4 (2020), p. 045018.
[133] Alexander Denzel and Johannes Kästner. “Gaussian process regression for geometry optimization”. In: *The Journal of Chemical Physics* 148.9 (2018), p. 094114.
[134] Ralf Meyer and Andreas W. Hauser. “Geometry optimization using Gaussian process regression in internal coordinate systems”. In: *The Journal of Chemical Physics* 152.8 (2020), p. 084112.
[135] Albert P. Bartók et al. “Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons”. In: *Phys. Rev. Lett.* 104 (13 Apr. 2010), p. 136403.
[136] Michael L. Stein. *Interpolation of Spatial Data : Some Theory for Kriging*. eng. Springer, 1999, Online–Ressource (XVII, 249 p).
[137] Ward Cheney and David Kincaid. *Numerical mathematics and computing*. Brooks/Cole, Cengage Learning, 2013.
[138] Eric Jones, Travis Oliphant, and Pearu Peterson. *SciPy: Open source scientific tools for Python*. http://www.scipy.org/.

[139] Jacob Gardner et al. “GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.
[140] Ke Wang et al. “Exact Gaussian Processes on a Million Data Points”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.
[141] Korina Kuhar et al. “Sulfide perovskites for solar energy conversion applications: computational screening and synthesis of the selected compound LaYS 3”. In: *Energy & Environmental Science* 10.12 (2017), pp. 2579–2593.
[142] David Eriksson et al. “Scalable Global Optimization via Local Bayesian Optimization”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.
[143] Mark Newman. *Networks*. Oxford university press, 2018.
[144] Ask H. Larsen et al. *GPatom, a Gaussian process framework for potential energy surface optimization*. 2021. URL: https://gitlab.com/gpatom/ase-gpatom.

Paper I

Materials property prediction using symmetry-labeled graphs as atomic-position independent descriptors

Peter Bjørn Jørgensen, **Estefanía Garijo del Río**, Mikkel N. Schmidt, Karsten Wedel Jacobsen

Physical Review B **100**, 104114 – Published 26 September 2019

Editors' Suggestion

Materials property prediction using symmetry-labeled graphs as atomic position independent descriptors

Peter Bjørn Jørgensen,[1] Estefanía Garijo del Río,[2] Mikkel N. Schmidt,[3] and Karsten Wedel Jacobsen[2]
[1]*Department of Energy Conversion and Storage, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*
[2]*Department of Physics, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*
[3]*Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*

(Received 21 May 2019; revised manuscript received 29 August 2019; published 26 September 2019)

Computational materials screening studies require fast calculation of the properties of thousands of materials. The calculations are often performed with density functional theory (DFT), but the necessary computer time sets limitations for the investigated material space. Therefore, the development of machine-learning models for prediction of DFT-calculated properties is currently of interest. A particular challenge for *new* materials is that the atomic positions are generally not known. We present a machine-learning model for the prediction of DFT-calculated formation energies based on Voronoi quotient graphs and local symmetry classification without the need for detailed information about atomic positions. The model is implemented as a message passing neural network and tested on the Open Quantum Materials Database (OQMD) and the Materials Project Database. The test mean absolute error is 22 meV on the OQMD and 43 meV on Materials Project Database. The possibilities for prediction in a realistic computational screening setting are investigated on a data set of 5976 $ABSe_3$ selenides with very limited overlap with the OQMD training set. Pretraining on OQMD and subsequent training on 100 selenides result in a mean absolute error below 0.1 eV for the formation energy of the selenides.

DOI: 10.1103/PhysRevB.100.104114

I. INTRODUCTION

Over the last decades, high-throughput computational screening studies have been employed to identify new materials within different areas such as (photo)electrochemistry [1–3], batteries [4,5], catalysis [6,7], and more [8–10]. Such studies are typically based on density functional theory [11,12] and because of computational requirements they are usually limited to some thousands or tens of thousands of materials. In order to investigate larger parts of the huge space of possible materials, new methods are needed to perform faster calculations or to guide the search in the material space in a more informed way.

One way to circumvent the computationally demanding DFT calculations is to use machine-learning (ML) techniques to predict materials properties, and this approach has been explored intensively the last years. Several descriptors or fingerprints to characterize the atomic structure of a material have been suggested including the partial radial distribution function [13] and the Coulomb matrix [14]. More involved fingerprints combining many atomic properties and crystal structure attributes based on Voronoi graphs have also been developed [15,16], along with graph representations, which are directly mapped onto convolutional neural networks [17–19].

The use of ML to speed up DFT calculations may have several goals in a computational screening setting. If the atomic structure (i.e., the positions of all the atoms) of a material is known, ML may in principle provide the same information about the material as a DFT calculation would: structural stability, phonon dispersion relations, elastic constants, etc. It might even in principle provide data of a better quality than standard (semi)local DFT calculations, comparable to more advanced DFT calculations with hybrid functionals or even higher-level methods as recently demonstrated for molecules [20].

However, the atomic positions of *new* materials will generally not be known. If the atomic positions are known from experiment, the material is not really new (even though many of its properties might be unknown) and if the positions are obtained from a DFT calculations there is no need to use a ML prediction of already calculated properties.

Our focus here will be the prediction of properties of *new* materials where the detailed atomic positions are unknown, and since the most crucial property of a new material is its stability we shall concentrate on prediction of formation energies.

The obvious question of course then is, how we can describe or classify a crystalline material without knowing the explicit positions of the atoms. The most fundamental property of a material is its chemical composition, i.e., for a ternary material $A_xB_yC_z$, the identity of the elements A, B, and C and their relative appearance $x : y : z$. It turns out that based on this information alone a number of predictions about material stability can be made. Meredig *et al.* [21] demonstrated that it is possible to predict thermodynamic stability of new compounds with reasonable accuracy based on composition alone, and a number of new compound compositions were predicted and their structures subsequently determined. However, this approach of course has its limitations as it cannot distinguish between materials with the same composition but different crystal structures.

A rigorous classification of a crystalline material comes from its symmetry. Any periodic material belongs to one of the 230 space groups, and this puts restrictions on the possible atomic positions. In the simplest cases of, say, a unary material

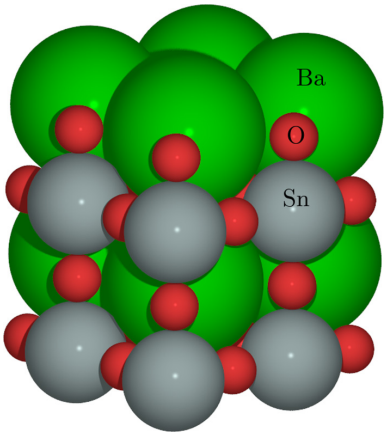


FIG. 1. Structure of $BaSnO_3$. The unit cell contains one Ba atom (green), one Sn atom (gray), and three O atoms (red).

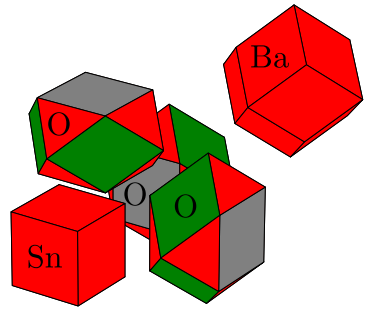


FIG. 2. Voronoi cells of $BaSnO_3$. The cells have been displaced for the visualization. The color of the facets corresponds to the atomic species of the neighboring atom (green for Ba, gray for Sn, and red for O neighbors).

with one atom in the unit cell with space group $Fm\text{-}3m$ (an fcc crystal), all atomic positions are determined up to a scaling of the volume. Similarly, the fractional positions (i.e., relative to the unit cell) of the atoms in materials with several elements can be determined entirely by symmetry as, for example, shown for $BaSnO_3$ in the cubic perovskite structure in Fig. 1. More generally, scaled atomic positions may be fully or partially determined depending on their symmetry, and the symmetry properties can be expressed using the so-called Wyckoff sites. This classification was recently used by Jain and Bligaard [22] to build a machine-learning model based on only composition and the Wyckoff positions, i.e., without any detailed information about the atomic positions. They were able to achieve a mean absolute error of about 0.07 eV/atom on the prediction of the formation energy on a test data set of more than 85 000 materials.

Here, we shall develop a machine-learning model, which does not require knowledge of the detailed atomic positions. However, unlike the model proposed by Jain and Bligaard, it will be based on local information about interatomic bonds and the symmetry of their environments. The bonds will be identified using Voronoi graphs and the symmetry will be classified using the Voronoi facets. The resulting model has a mean absolute error on the heats of formation for the Open Quantum Materials Database (OQMD) of only 22 meV and for the ICSD part of OQMD it is 40 meV.

In Sec. II we describe the proposed graph representation based on quotient graphs and the classification of Voronoi facet point symmetry and in Sec. III we investigate the relation between quotient graphs and prototypes based on data from OQMD. This is followed by an introduction of the machine-learning model and the data sets in Secs. IV and V, respectively. The numerical results are presented in Sec. VI and followed by the conclusions in Sec. VII.

II. GRAPH REPRESENTATION

As representation for the machine-learning algorithm, we use the quotient graph as introduced by [23] and also used in [19]. The quotient graph is a finite graph representation of the infinite periodic network of atoms. Every atom in the unit cell corresponds to a vertex of the quotient graph. We denote the graph G and the set of N vertices $\{v_i\}_{i=1}^{N}$. When two atoms are connected in the network, we draw an edge between the atoms in the quotient graph. In this work we use the Voronoi diagram to decide when two atoms are connected, specifically a pair of atoms are connected if they share a facet in the Voronoi diagram. Due to periodic boundary conditions a pair of atoms may share several facets, and in this case there will be several edges between the atoms. When interatomic distances are available, the edges are labeled with the distance between the atoms.

As an example, we look at $BaSnO_3$ in the perovskite structure as shown in Fig. 1. This material has five atoms in the unit cell. After performing Voronoi tessellation we get a Voronoi cell for each atom in the unit cell as shown in Fig. 2. The Voronoi diagram defines the edges in the quotient graph which is illustrated in Fig. 3.

An inherent problem with Voronoi graph construction method is that small perturbations of the atom positions may lead to different graphs. Classification of different types of instabilities has even been used by Lazar *et al.* [24] to characterize local structure. As shown by Reem [25], small

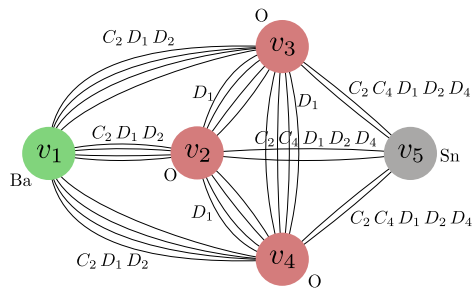


FIG. 3. Quotient graph for $BaSnO_3$. The edge labels show the point groups of the corresponding facets of the Voronoi diagram. For this particular case, the repeated edges between vertices all have the same point groups, but in general they could be different.

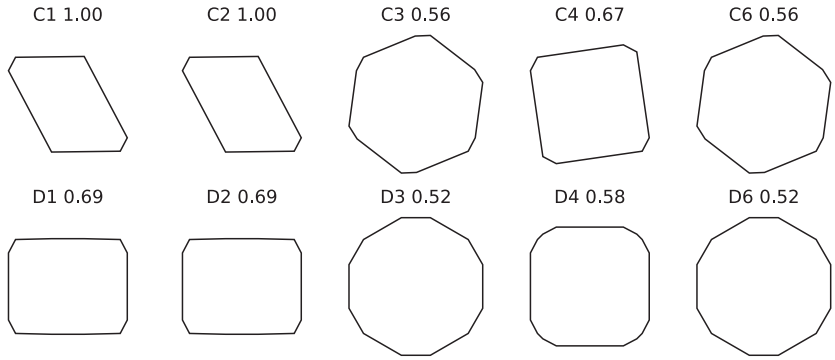


FIG. 4. Each shape is the convex hull of the shape in the top left corner after the symmetry operations of each of the point groups have been applied. The label above each shape denotes the point group and the symmetry measure for that group.

changes in the Voronoi sites lead to only small changes in the Voronoi cell volume. However, small perturbations can still lead to appearances of quite small facets. This is, for example, often the case for structures with high symmetry, where small displacements of the atoms introduce new facets. To increase the stability, we remove these small facets and the corresponding connections in the graph by introducing a cutoff in the solid angle of the facet Ω_{cut}. We use $\Omega_{\text{cut}} = 0.2$, but as we shall see later the results are surprisingly stable with regard to increasing this value. A more advanced method for improving the stability of the Voronoi graph has been proposed by Malins *et al.* [26].

The graph is annotated with the symmetry group of each of the Voronoi facets. In the following section, we describe this symmetry classification in more detail.

A. Symmetry-group classification

To characterize the symmetry of an atomic environment, we classify the symmetry of each Voronoi facet into the nine nontrivial two-dimensional point groups ($C_2, C_3, C_4, C_6, D_1, D_2, D_3, D_4, D_6$). The classification method is inspired by the symmetry measure introduced by Heijmans and Tuzikov [27]. Given the vertices of the two-dimensional Voronoi facet, we go through the following procedure:

(1) Compute centroid and center the shape.

(2) Search for mirror axis and align it with the x axis if it exists.

(3) For each point symmetry group apply all elements of the group and calculate the area of the convex hull of the points generated by this procedure.

The symmetry measure is then the ratio between the area of the original shape and the area defined by the convex hull of the new vertices. When the symmetry measure for a given group is close to unity, we label the facet as having this symmetry. See Fig. 4 for an example shape and its symmetry measure for each group. The search for mirror axis in step 2 is done by computing the moment of inertia and testing the two principal axes for mirror symmetry. When the moments of inertia are the same, for example when the shape is a regular polygon, the principal axes are arbitrary and we fall back to testing for mirror symmetry at all axes going through the centroid and either a vertex or a midpoint of a line segment. For a regular hexagon, these axes are illustrated in Fig. 5.

III. GRAPH REPRESENTATION AND PROTOTYPES

In many applications, prototypes are used as a descriptor for the overall structure of a material and as part of a computational screening procedure some of the atoms of the prototypes may be swapped with other elements. We want to assess whether there is a correspondence between the prototypes and Voronoi graphs, i.e., do two materials with the same prototype have the same Voronoi graph and do two materials with the same Voronoi graph have the same prototype? The question cannot be ultimately answered because prototype naming is not completely well defined: in some cases, several different prototypes are used to describe the same material, and many materials may not have prototypes attached to them. But, we can show to which extent Voronoi graphs are aligned with the use of prototypes.

For this analysis we use the OQMD database with the prototypes assigned in the database. We note that this assignment is not generally unique. For example, an elemental compound in the fcc structure may be labeled with either "Cu" or "A1_Cu" in the database. In other cases, two clearly different structures are classified with the same prototype.

We investigate all unary, binary, and ternary compounds in the database and for each of these sets we study the link between graphs G and prototypes P, i.e., if we know that a given structure has a specific prototype, do we then also know which graph it has and vice versa. One way of measuring this

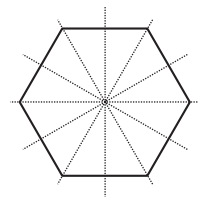


FIG. 5. Mirror axes of a hexagon.

TABLE I. Correspondence between Voronoi graphs and prototypes in OQMD with and without symmetry labels. N denotes the number of entries, $|G|$ the number of unique Voronoi graphs, and $|P|$ the number of different prototypes. $H(G)$ and $H(P)$ are the entropy of the distribution of graphs and prototypes, respectively, while $I(G;P)$ is the mutual information between the two distributions and $U(G|P)$, $U(P|G)$ are the normalized mutual information. (a) For graphs without symmetry labels. (b) For graphs with symmetry labels.

	N	$\|G\|$	$\|P\|$	(a) $H(G)$	$H(P)$	$I(G;P)$	$U(G\|P)$	$U(P\|G)$
Unary	1487	85	67	4.4	4.7	3.7	0.84	0.80
Unary ICSD	196	46	49	4.2	4.2	3.8	0.90	0.90
Binary	53528	1318	871	4.3	4.5	3.8	0.90	0.86
Binary ICSD	5862	1219	850	8.2	8.0	7.6	0.92	0.95
Ternary	339960	4006	1754	2.0	1.9	1.8	0.91	0.98
Ternary ICSD	11500	3487	1740	10.0	9.1	8.8	0.88	0.97
	N	$\|G\|$	$\|P\|$	(b) $H(G)$	$H(P)$	$I(G;P)$	$U(G\|P)$	$U(P\|G)$
Unary	1487	222	67	6.1	4.7	4.3	0.70	0.91
Unary ICSD	196	68	49	4.8	4.2	4.0	0.84	0.96
Binary	53528	2040	871	4.7	4.5	4.0	0.84	0.90
Binary ICSD	5862	1742	850	8.8	8.0	7.8	0.89	0.97
Ternary	339960	5703	1754	2.1	1.9	1.8	0.88	0.99
Ternary ICSD	11500	4504	1740	10.5	9.1	9.0	0.85	0.99

is through the mutual information $I(G;P)$ of G and P. The mutual information is symmetric and can be computed as

$$I(G;P) = H(G) - H(G|P) \quad (1)$$

$$= H(P) - H(P|G), \quad (2)$$

where H denotes the entropy. The mutual information is thus the average decrease in entropy we get from knowing the other variable. We also compute the normalized mutual information known as the uncertainty coefficient $U(X|Y) = I(X;Y)/H(X)$ which can be seen as given Y what fraction of bits of X can we predict. To compute these quantities, we need the distribution over graphs and we obtain these distributions approximately by comparing graph fingerprints.[1] The quantities for OQMD are shown for the unlabeled graph in Table I(a) and for the graph labeled with rotation symmetries in Table I(b).

The uncertainty coefficient is close to 90% in most cases except for the unary compounds $U(P|G)$. In this case, structures with different prototypes map to the same graph and we may be discarding important structural information. Including symmetry information increases the number of unique graphs significantly, which implies that the uncertainty coefficient $U(G|P)$ decreases while $U(P|G)$ increases.

IV. NEURAL MESSAGE PASSING MODEL

In this section we introduce the machine-learning model which takes the labeled graph as input and outputs an energy prediction as a scalar. We describe the model as message passing on a graph following the notational framework introduced by Gilmer *et al.* [28]. We follow the message passing notation, but the model we are going to introduce can be seen as an extension of the SchNet model [18], which can also be cast into this framework as we have shown in prior work [29].

Denote the graph G with vertex features x_v and edge features ε_{vw} for an edge from vertex v to vertex w. Each vertex has a hidden state h_v^t at "time" (or layer) t and we denote the edge hidden state e_{vw}^t. The hidden states of vertices and edges are updated in a number of interaction steps T. In each step, the hidden states of vertices are updated in parallel by receiving and aggregating messages from neighboring vertices. The messages are computed by the message function $M_t(\cdot)$ and the vertex state is updated by a state transition function $S_t(\cdot)$, i.e.,

$$m_v^{t+1} = \sum_{w \in N(v)} M_t\left(h_v^t, h_w^t, e_{vw}^t\right), \quad (3)$$

$$h_v^{t+1} = S_t\left(h_v^t, m_v^{t+1}\right), \quad (4)$$

where $N(v)$ denotes the neighborhood of v, i.e., the vertices in the graph that has an edge to v. The edge hidden states are also updated by an edge update function $E_t(\cdot)$ that depends on the previous edge state as well as the vertices that the edge connects:

$$e_{vw}^{t+1} = E_t\left(h_v^t, h_w^t, e_{vw}^t\right). \quad (5)$$

After T interaction steps the vertex hidden state represents the atom type and its chemical environment. We then apply a readout function $R(\cdot)$ which maps the set of vertex states to a single entity

$$\hat{y} = R\left(\left\{h_v^T \in G\right\}\right). \quad (6)$$

The readout function operates on the set of vertices and must be invariant to the ordering of the set. This is often achieved simply by summing over the vertices. In some architectures the final edge states are also included as an argument to the

[1]The graph fingerprints are computed using the neural message passing model with random weight initialization. We use two instances of neural network weight initialization and six different atomic embedding instances, thus having 12 models in total. The fingerprint is then a vector where each entry is the scalar output of one of these models.

readout function. The message function $M_t(\cdot)$, state transition function $S_t(\cdot)$, edge update function $E_t(\cdot)$, and readout function $R(\cdot)$ are implemented as neural networks with trainable weight matrices. To fully define the model, we just need to define these functions and a number of models can be cast into this framework. We use different weight matrices for each time step t, however, in some architectures the weights are shared between layers to reduce the number of parameters.

In this work we use the model proposed in our prior work [29]. The model is an extension of the SchNet model [18], with the addition of an edge update network. The message function is only a function of the sending vertex and can be written as

$$M_t\left(h_w^t, e_{vw}^t\right) = \left(W_1^t h_w^t\right) \odot g\left(W_3^t g\left(W_2^t e_{vw}^t\right)\right), \tag{7}$$

where $\odot$ is element-wise multiplication, the W's are weight matrices, and $g(x)$ is the activation function, more specifically the shifted soft-plus function $g(x) = \ln(e^x + 1) - \ln(2)$. It can be seen as a smooth version of the more popular rectified linear unit. In this description we omit the bias terms to reduce the notational clutter, but in the implementation a trainable bias vector is added after each matrix-vector product, i.e., there is an appropriately sized bias vector for each of the W's. As an edge update network we use a two-layer neural network and the input is a concatenation of the sending and receiving vertex states and the current edge state:

$$e_{vw}^{t+1} = E_t\left(h_v^t, h_w^t, e_{vw}^t\right) = g\left(W_{\mathrm{E2}}^t g\left(W_{\mathrm{E1}}^t\left(h_v^t; h_w^t; e_{vw}^t\right)\right)\right), \tag{8}$$

where $(\cdot;\cdot)$ denotes vector concatenation. This choice of edge update network means that the edge state for each of the two different directions between a pair of vertices becomes different after the first update. This network is the only architectural difference from the SchNet model [18], i.e., if we set $e_{vw}^{t+1} = e_{vw}^t$ the model we describe here would be identical with the SchNet model. The state transition function is also a two-layer neural network. It is applied to the sum of incoming messages and the result is added to the current hidden state as in residual networks [30]:

$$S_t\left(h_v^t, m_v^{t+1}\right) = h_v^t + W_5^t g\left(W_4^t m_v^{t+1}\right). \tag{9}$$

After a number of interaction steps T we apply a readout function for which we use a two-layer neural network that maps the vertex hidden representation to a scalar and finally we average over the contribution from each atom, i.e.,

$$R\left(\left\{h_v^T \in G\right\}\right) = \frac{1}{N} \sum_{h_v^T \in G} W_7 g\left(W_6 h_v^T\right). \tag{10}$$

In other words, an atom and its chemical environment are mapped to an energy contribution.

A. Initial vertex and edge representation

The initial vertex hidden state h_v^0 depends on the atomic number of the corresponding atom. The atomic number is used to look up a vector representation for that atom. Using a hidden representation of size 256 the initial hidden state is thus the result of a lookup function $\ell(x) : \mathbb{N} \to \mathbb{R}^{256}$. The weights in the vector representation are also trained during the optimization.

We use the model on three different levels of available information. In the most ignorant scenario, we have no labels on the edges of the graph and in this case the edge update function (8) just ignores the edge representation on the first layer, i.e., e_{vw}^0 is a "vector" of length 0 and $e_{vw}^t, t \in 1, \ldots, T$ are vectors of length 256. The next level of information is to include the point-group information as described in Sec. II A. There are nine nontrivial point groups and we encode this information as an indicator vector of length 9, where 1 means that the corresponding facet belongs to the given point group. Finally, we also run numerical experiments with the full spatial information for which the edges of the quotient graph are labeled with the interatomic distance. The distances are encoded by expanding them in a series of exponentiated quadratic functions as also done in [17,18,29]

$$\left(e_{vw}^0\right)_k = \exp\left(-\frac{[d_{vw} - (-\mu_{\min} + k\Delta)]^2}{2\Delta^2}\right), \; k = 0 \ldots k_{\max} \tag{11}$$

where $\mu_{\min}$, Δ, and $k_{\max}$ are chosen such that the centers of the functions cover the range of the input features. This can be seen as a soft 1-hot-encoding of the distances, which makes it easier for a neural network to learn a function where the input distance is uncorrelated with the output. In the experiments we use $\mu_{\min} = 0$, $\Delta = 0.1$, and $k_{\max} = 150$.

V. DATA SETS

For the numerical experiments we use two publicly available data sets and one data set we generate.

a. Materials Project [31]. This data set contains geometries and formation energies of 86680 inorganic compounds with input structures primarily taken from the the Inorganic Crystal Structure Database (ICSD) [32]. We use the latest version of the database (version 2018.11). The number of examples is reduced to 86579 after we exclude all materials with noble gases (He, Ne, Ar, Kr, Xe) because they occur so infrequently in the data set that we consider them as outliers. This brings the number of different elements in the data set down to 84.

b. Open Quantum Materials Database (OQMD) [33,34]. Is also a database of inorganic structures and we use the currently latest version (OQMD v1.2) available on the project's website. Again, we consider materials with noble gases as outliers and we also exclude highly unstable materials with a heat of formation of more than 5 eV/atom. Some entries in the database are marked as duplicates and we filter them in the following way: When a set of duplicates is encountered we use the first entry of the database, but if the standard deviation of the calculated heat of formation exceeds 0.05 eV/atom, we discard the whole set of duplicates. This leaves us with a total of 562134 entries.

For both data sets we split the entries into five parts of equal size to be used for fivefold cross validation, where the machine is trained on $\frac{4}{5}$ of the data, and the remaining $\frac{1}{5}$ is used for testing. For OQMD we also distribute the entries of OQMD that originate from ICSD equally between the five folds.

c. Ternary selenides $ABSe_3$. For further testing, we have developed a third data set of selenides. The intention behind

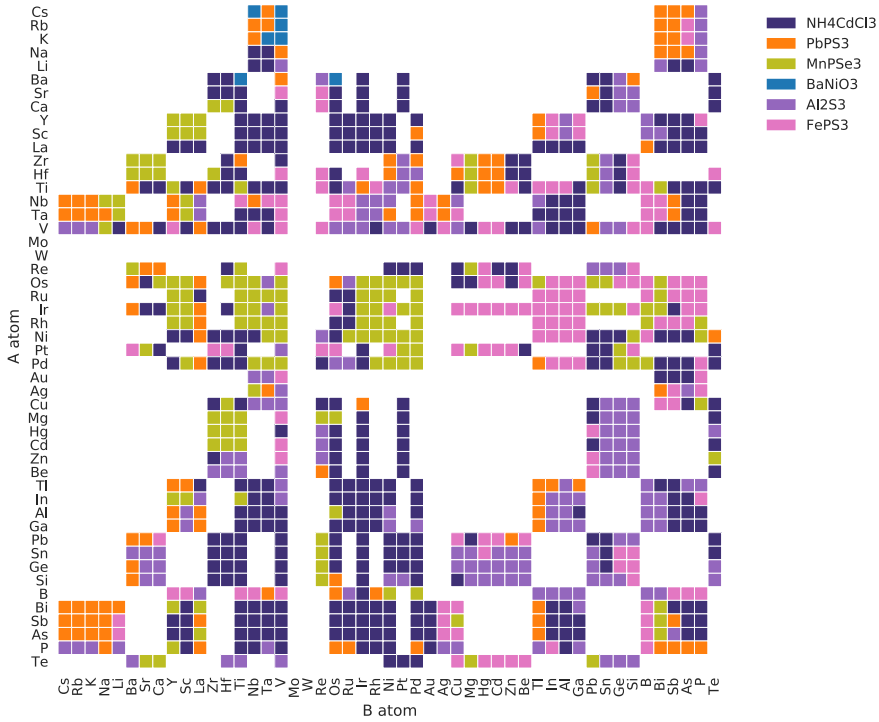


FIG. 6. Map of the most stable prototype for each composition $AB\mathrm{Se}_3$. The compositions that do not fulfill the valence rule have not been studied and, thus, they are not colored.

this set is to test the ability of the model in a realistic computational screening setting. This data set has only very limited overlap with OQMD, and predictions are made exclusively based on the symmetry-labeled graphs of the new materials without any detailed information about the atomic coordinates.

The data set contains the structures and formation energies of 5976 ternary selenides with stoichiometries $AB\mathrm{Se}_3$, where A and B are different transition metals in six different prototypes.

The procedure for generating this data set resembles the one presented in [3]. We start by looking up the $AB\mathrm{Se}_3$ compounds reported in ICSD [32], and selecting the six prototypes that appear more than once: hexagonal $P6_3/mmc$ structure of $BaNiO_3$, orthorhombic $Pnma$ structure of NH_4CdCl_3/Sn_2S_3, monoclinic $C2/m$ $FePS_3$, monoclinic Pc structure of $PbPS_3$, trigonal $R\bar{3}$ structure of $MnPSe_3$ and hexagonal $P6_1$ structure of Al_2S_3.

These structures are then used as templates, and we substitute the transition metal atoms A and B by 49 transition metals. Here, we avoid for simplicity Cr, Mn, Fe, and Co, which usually lead to structures with large magnetic moments. We also limit ourselves to those combinations $AB\mathrm{Se}_3$ for which the valences of cations and anions add up to zero. This leads to a total of 512 $AB\mathrm{Se}_3$ compounds: 484 ternaries, which are then studied in 12 structures (6 for the $AB\mathrm{Se}_3$ and 6 for the $BA\mathrm{Se}_3$) and 28 binaries, for which we study 6 different structures. A map to the compositions and structures studied can be found in Fig. 6.

The resulting 5976 structures have then been relaxed using density functional theory (DFT) as implemented in the codes ASE [35] and GPAW [36]. We perform two different kinds of electronic structure calculations: a coarse-grained calculation with the exchange-correlation functional PBESOL [37] for the steps of the optimization and fined grained at the relaxed structure with the PBE exchange-correlation functional [38]. The cutoff energy for the plane-wave basis set used to expand the wave functions is 800 eV in both cases. For the sampling of the Brillouin zone we use a Monkhorst-Pack mesh [39] with a density of $5.0/(\text{Å}^{-1})$ k points in each direction for the relaxation steps and of $8.0/(\text{Å}^{-1})$ k points for the refined calculation at the relaxed structure. All structures have been relaxed until the forces on the atoms are less than 0.05 eV/Å.

VI. NUMERICAL RESULTS AND DISCUSSION

To assess the loss in accuracy going from full spatial information to unlabeled quotient graph we train/test the model in three different settings, as mentioned in Sec. IV A. In the most ignorant setting, the quotient graph has only unlabeled edges. On the next level we label the edges with the symmetry of the corresponding Voronoi facet. With full spatial information, the edges of the quotient graph are labeled with the distance between the atoms. The model is trained

TABLE II. MAE in meV/atom of test set energy predictions obtained through fivefold cross validation. The ICSD results are for the model trained on OQMD and tested only on the ICSD part of OQMD.

Data set	Dist.	Sym.	No sym.	V-RF
OQMD all	14	22	26	85
OQMD unary	58	110	128	85
OQMD binary	30	48	60	86
OQMD ternary	14	20	23	80
ICSD all	24	40	45	113
ICSD unary	56	75	119	72
ICSD binary	32	51	58	118
ICSD ternary	22	35	39	109
Matproj all	26	43	43	84
Matproj unary	96	149	179	127
Matproj binary	48	69	73	99
Matproj ternary	27	43	43	87

with the Adam optimizer [40] for up to 10×10^6 steps using a batch size of 32. The initial learning rate is 1×10^{-4} and it is decreased exponentially so at step s the learning rate is $10^{-4} \times 0.96^{\frac{s}{10^5}}$. When training on OQMD and materials project we use 5000 examples from the training data for early stopping. More specifically, this validation set is evaluated every 50 000 steps and if the mean absolute error (MAE) has not improved for 1×10^6 steps, the training is terminated. When training on the ternary selenides $ABSe_3$ data set the 10% of the training data is used as a validation set and the validation set is evaluated every training epoch. In some of the results we use a model that has been pretrained on OQMD. In that case, the model is trained on four out of five OQMD folds until the stopping criterion is met and the weights of the model are then used as initialization for training on the selenides data set. The implementation of the model as well as the code used for generating the input graphs are available on GITHUB.[2]

[2]https://github.com/peterbjorgensen/msgnet; https://github.com/peterbjorgensen/vorosym

A. OQMD

The mean absolute errors (MAE) and root-mean-squared errors (RMSE) of the test set predictions are shown in Table II and the MAE is further visualized in Fig. 7. As expected, the lowest prediction errors are obtained with the model where distance information is provided. If we focus on the OQMD, the overall MAE is as low as 14 meV with distance information. This is lower than the SchNet model [18] by almost a factor of 2 because of the edge updates as discussed in Ref. [29]. Two versions of the models without distance information are also shown. In one of them, the symmetry information has not been used, but in the other one the symmetry classification of the Voronoi facets has been included as edge information. These two models do of course have larger errors than the one benefiting from the distance information, but still the error is surprisingly small. The MAE is only 22 meV for the model using symmetry information. For comparison, the results for the model proposed by Ward *et al.* [15] are also shown in the figures (labeled V-RF for Voronoi–random forest). This model also builds on a Voronoi graph construction, but since the fractional areas of the Voronoi cells are provided, information about the distances is provided. Furthermore, many other attributes are added as information to the random forest algorithm applied. When this machine is applied to OQMD (using the same fivefold splitting of the data as applied to the other algorithms), the resulting error is considerably larger, 85 meV, for all of OQMD.

To understand more about the behavior of the ML algorithms investigated here, we have considered the test errors on different subsets of OQMD and also on the material project database [31]. Let us first note that the OQMD contains two different types of structure sources. One type, which gives rise to the largest number of materials, consists of a number of fixed crystal structures or prototypes decorated by the different chemical elements. There are 16 elemental prototypes, 12 binary ones, and 3 ternary ones. For two of the ternary ones, one of the elements is predefined to be oxygen. This generates a very large number of materials of varying composition and stability, but in a fairly small number of different crystal structures. The other type of structures comes from materials from the experimental ICSD database. This group is characterized by a much greater variation in

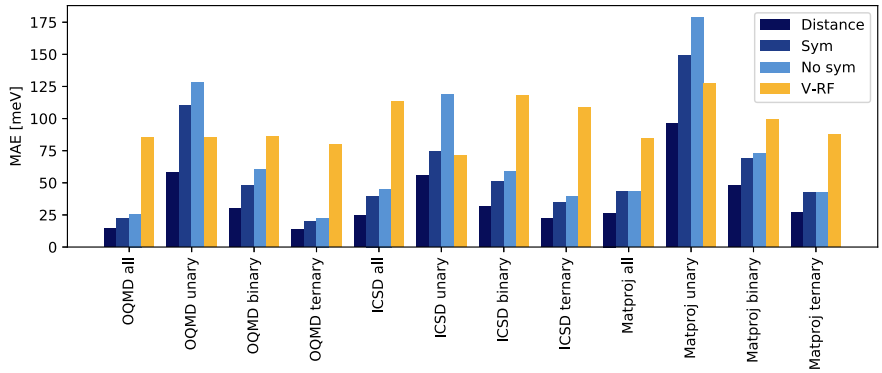


FIG. 7. The figure shows the data in Table II.

the crystal structures, but is naturally limited to mostly stable materials since they have been experimentally synthesized.

We first consider the test error on the subsets of OQMD consisting of the unary, binary, and ternary systems, and we shall focus on the model where the symmetry information is included, but the distances are not. As can be seen from Table II, the test error is considerably larger on the unary systems (110 meV) than on the database as a whole. This also holds for the binary ones, but to a smaller degree (48 meV). It is not clear to us at the moment exactly why this is so, but we shall discuss some possible explanations. The unary and binary systems only constitute a fairly small part of the total database, and the weight of these systems during the training is therefore also limited. Another factor may be that a large fraction of the unary and binary systems belong to the group of materials where the crystal structures are systematically generated as mentioned above. This means that many rather "artificial" and unstable materials are generated, where the atoms are situated in environments, which do not occur in reality, and the resulting energies may be far above more stable structures. This could potentially be difficult for the machine to learn.

B. ICSD/OQMD

Table II also shows the results for the ICSD subset of the OQMD database. The results shown are for the model trained on all of OQMD but tested only on the ICSD subset. The overall MAE is seen to be roughly a factor of 2 larger than for all of OQMD. This is probably due to the fact that the ICSD is a subset with a large variation of structures, and this makes prediction more difficult on average. We see the same trend as for all of OQMD, that the error decreases going from unary to binary to ternary systems. For the unary systems, the test error is in fact lower for the ICSD subset than for all of OQMD, which may be due to the fact that physically artificial high-energy systems appear in OQMD but not in ICSD. For the binary systems there is a balance: the ICSD does not contain so many high-energy systems, which could make predictions better, but on the other hand, the larger variation of crystal structures is more difficult to predict.

C. Materials project database

The models have also been trained and tested on the Materials Project data set [31]. The overall error is fairly similar to the one obtained for the ICSD subset of OQMD as might be expected since the materials project is also based on mostly materials from the ICSD. The errors for the unary and binary subsets are somewhat larger for the materials project database. This might be due to the fact that the machine trained on OQMD benefits from the larger number of systematically generated unary and binary systems in that database.

D. RMSE vs MAE

The root-mean-square errors are shown in Table III. In all cases, the values are considerably higher than the MAE. This is an indication that the distribution of the errors have heavier tails than a Gaussian, and as we shall see in the following examples that a significant number of outliers exist. The outliers might be due to limitations of the model, but could also appear because of problematic entries in the database as also discussed by Ward *et al.* [15].

TABLE III. RMSE in meV/atom of test set energy predictions obtained through fivefold cross validation. The ICSD results are for the model trained on OQMD and tested only on the ICSD part of OQMD.

Data set	Dist.	Sym.	No sym.	V-RF
OQMD all	54	74	80	173
OQMD unary	184	269	342	190
OQMD binary	89	113	138	162
OQMD ternary	52	70	71	131
ICSD all	81	107	111	188
ICSD unary	262	227	353	180
ICSD binary	73	116	129	202
ICSD ternary	88	112	102	182
Matproj all	72	121	122	172
Matproj unary	246	341	467	289
Matproj binary	120	190	192	203
Matproj ternary	65	119	111	181

E. Solid-angle cutoff of Voronoi facets

The above results are all calculated using a cutoff of the Voronoi facet solid angle of $\Omega_{\text{cut}} = 0.2$. However, the results are almost independent of the value as shown in Fig. 8, where the MAE on all of OQMD is shown for the model where symmetry but no distance information is included. We see that the error decreases slightly when small facets are removed with $\Omega_{\text{cut}} = 0.2$, and increases only slowly when Ω_{cut} is further increased. We take this as an indication that the connectivity of the material is well described even when the graph is reduced to essentially include only nearest-neighbor bonds.

F. ABO_3 materials in OQMD

We now consider the subset of all oxides in the OQMD with the composition ABO_3. We shall investigate to which extent the model is able to predict the right ground-state structure for a given composition. We first show the overall prediction for the 12 935 materials of this type in OQMD in Fig. 9. We again use the model with symmetry-labeled graphs

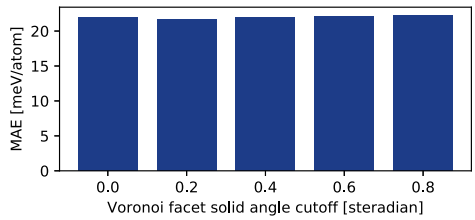


FIG. 8. Prediction error on OQMD test set vs Voronoi facet solid angle cutoff Ω_{cut} for the model using symmetry labels. The error decreases slightly when removing small facets and increases only slowly when Ω_{cut} is further increased.

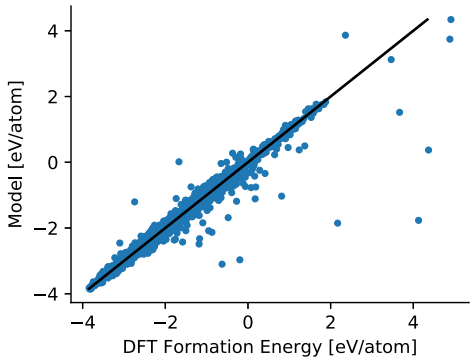


FIG. 9. Test set predictions on 12 395 ABO_3 structures of OQMD (MAE=36, RMSE = 112 meV/atom) using fivefold cross validation, i.e., the plot is a collection of predictions from five different models, each trained on $\frac{4}{5}$ of the data and tested on the remaining $\frac{1}{5}$.

without distance information. The MAE is 36 meV, which is about the same value as the one for the subset of ternaries in ICSD (35 meV). The RMSE is again significantly higher (112 meV) because of severe outliers as can be seen in the plot.

We now ask the following question: given a composition (A, B) the model predicts a ground-state structure G_{ML}. If we are going to investigate this structure and other low-energy structures with DFT, how high up in energy (as predicted by the model) do we have to go before we find the DFT ground-state structure G_{DFT}? We only include entries for which there is more than one structure (12 329/12 395) and the average number of structures per composition is 4.7. The energy difference $\Delta E = E^{\mathrm{ML}}(G_{\mathrm{DFT}}) - E^{\mathrm{ML}}(G_{\mathrm{ML}})$ of course varies from system to system, and the distribution is shown in Fig. 10. The mean absolute difference (MAD) of this

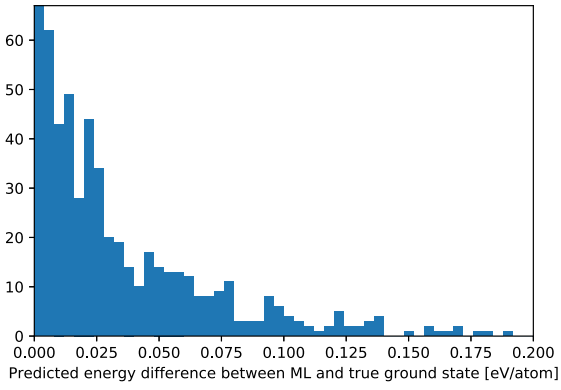


FIG. 10. Predicted energy difference between the DFT ground state and the ML ground state: $\Delta E = E^{\mathrm{ML}}(G_{\mathrm{DFT}}) - E^{\mathrm{ML}}(G_{\mathrm{ML}})$ for the ABO_3 materials in OQMD. The total number of compositions is 2646. The peak at zero is much higher than shown in the graph. It corresponds to the 2097 compositions, where the right ground state is predicted. For the remaining 549 compositions, the mean absolute difference is 44 meV/atom.

distribution is very small, only 9 meV, and the maximum error is a clear outlier at 0.92 eV. The reason for the small MAD is that for 2097 out of the 2646 compositions the correct ground state is predicted, however, in many cases because only two structures exist in the database for a given composition. For comparison, the expected number of correctly predicted ground-state structures with random guessing is 843. If we only look at the 549 compositions for which the ML model predicts the wrong ground state, the MAD is 44 meV/atom. For comparison, the energy prediction for the ground-state structures has an MAE of 29 meV/atom. The low MAD value of 44 meV is promising for applications to computational screening. It sets an energy scale for how many structures have to be investigated by DFT to identify the DFT ground state after the model predictions have been generated.

G. $ABSe_3$ selenides

The last data set we shall consider consists of selenides with the $ABSe_3$ composition as discussed in the section about the data sets. This data set is considerably more challenging for two reasons. First, there is very little overlap between this data set and the training data set OQMD. Only six materials are shared between the two data sets, and the test predictions for these are shown in Fig. 11(a). The MAE is 24 meV, and the RMSE is also low, only 38 meV. The second challenge is that we shall now use the model to make predictions based on the initial graph before relaxations. The six different prototypes in the data set each have a graph in the original material giving rise to the naming of the prototype. For example, one of the types is hexagonal $P6_3/mmc$ structure of $BaNiO_3$, so for predictions in this structure we shall use the graph of $BaNiO_3$. Some of the prototype structures have a fair number of atoms in the unit cell (up to 20) and a low symmetry (monoclinic), which means that there are many free atomic coordinates that are optimized during relaxation. This leads to frequent modifications of the graph during relaxation.

Figure 11(b) shows the model predictions based on the initial prototype graphs versus the DFT energies of the resulting optimized structures. The MAE is 176 meV, which is considerably higher than the value for the oxides. Particularly large deviations are seen for large and positive heats of formation. In a computational screening setting this might not be an issue because the high-energy materials are going to be excluded anyway. The RMSE is only a factor 236/176 = 1.34 larger than the MAE, which is due to the small number of outliers compared to, for example, the oxides (Fig. 9).

The prediction quality can be significantly improved by additional training on the selenide data set. Even a limited number of data points have a considerable effect. This is to be expected since the overlap between the selenide data set and the OQMD is only six materials as mentioned above. Figure 11(c) shows the model-DFT comparison if the model is first trained on the OQMD data set and then subsequently trained on 100 materials out of the 5976 selenides in the database. The MAE is reduced from 176 to 95 meV bringing the error down to a value comparable to the error between DFT and experiment [34].

The effect of additional training on the selenide data set is shown as a function of training set size on a logarithmic scale

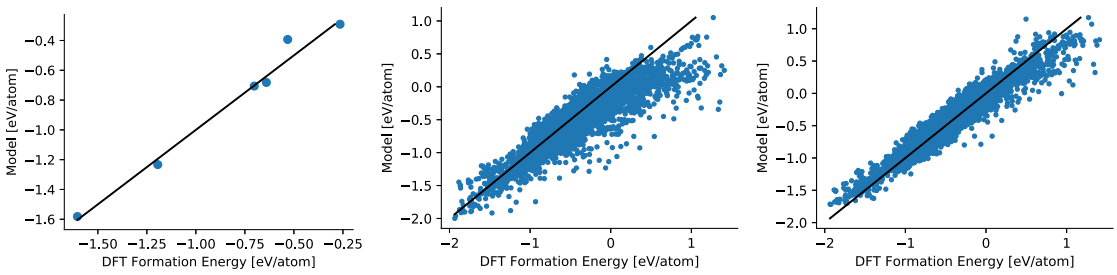


FIG. 11. Predictions on $AB\mathrm{Se}_3$ test set with model pretrained on OQMD. (a) Predicted energy for selenides in OQMD (MAE=24 RMSE=38 meV/atom). (b) Predicted energy for selenides (initial structures) using model trained on OQMD (MAE=176 RMSE=236 meV/atom). (c) Predicted energy for selenides (initial structures) using model pretrained on OQMD and then on 100 selenides (MAE=95 RMSE=129 meV/atom.

in Fig. 12. The points on the y axis correspond to the situation without any additional training. As can be seen, a small amount of additional training leads to significant reduction of the prediction error. The solid curve with square markers corresponds to the situation discussed above where the model is first trained on OQMD, and then further trained on the initial graphs (but relaxed energies) for part of the selenides. For comparison, the solid curve with diamond markers shows the prediction error, when the training and prediction is based on the final graph. Using the initial graphs instead of the final graphs gives rise to only a slightly higher MAE. This is encouraging for the potential use of the approach in computational screening studies, where predictions have to be based on the initial prototype structures to avoid the computationally costly DFT calculations.

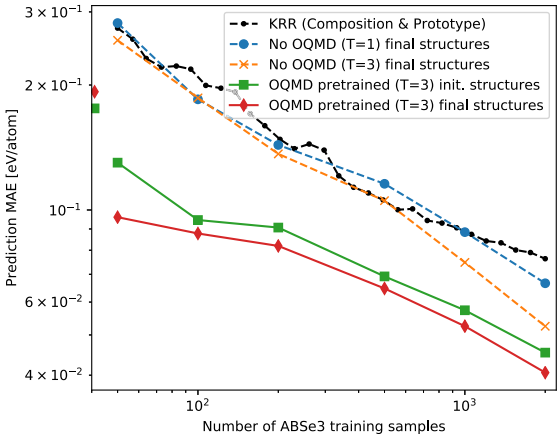


FIG. 12. Predictions on $AB\mathrm{Se}_3$ structures with increasing number of $AB\mathrm{Se}_3$ training samples. The solid lines correspond to models that have been pretrained on OQMD and then on the $AB\mathrm{Se}_3$ data set. The unconnected points correspond to the model only trained on OQMD final structures, i.e., the pretrained model. The parameter T denotes the number of interaction steps and initial/final structures refer to whether the model input is the graph derived from the prototype structure or the DFT-relaxed structure. KRR denotes a kernel ridge regression baseline model using only composition and prototype as input.

As a baseline, we also show the results of the model if it is trained exclusively on the selenide data set (dashed curve with cross markers). As expected, the MAE is much larger than for the pretrained model for small amounts of data. For larger training sets, the MAE drops gradually and with a data set size of about 500 materials, the prediction error is comparable to the one for the OQMD-pretrained model, which is trained on an additional 50 selenides. We ascribe the rather successful performance of the model without pretraining at large training set sizes to the systematic character of the data set: only six different crystal structures are represented and they are systematically decorated with a particular subset of atoms. The last model (dashed curve with circle markers) is again only trained on the selenide data set, but now only one interaction step ($T = 1$) is performed in the message passing neural network in contrast to the three iterations used otherwise. The performance is seen to be rather similar to the model with $T = 3$ up to a training data set size of 300. With only one iteration in the network information about the identity of neighboring atoms is exchanged, and this is apparently sufficient to roughly characterize the six crystal structures. At larger training set sizes, where the prediction error is smaller, the network with three iterations outperforms the one with only one iteration.

We also include an even simpler baseline model that uses only the composition and the prototype as input and is only trained on the $AB\mathrm{Se}_3$ data set. For this baseline the input vector representation consists of a 1-hot-encoding for the atom type of the A atom, a 1-hot-encoding for the B atom, and a 1-hot-encoding for the prototype. We use kernel ridge regression (KRR) as implemented by SCIKIT-LEARN using the RBF kernel and using 10-fold cross validation to choose the hyperparameters α ($\ell 2$-penalty weight) and γ (kernel length scale) on the grid $\alpha \in [1, 0.1, 0.01, 0.001]$ and $\gamma \in [0.01, 0.1, 1.0, 10.0, 100.0]$. The prediction error is similar to the other baseline model that uses only one interaction step.

Figure 13 shows the distribution of the predicted energy difference between the DFT ground-state structure and the ML-predicted ground-state structure $\Delta E = E^{\mathrm{ML}}(G_{\mathrm{DFT}}) - E^{\mathrm{ML}}(G_{\mathrm{ML}})$ for the selenide data set. Only in 104 out of

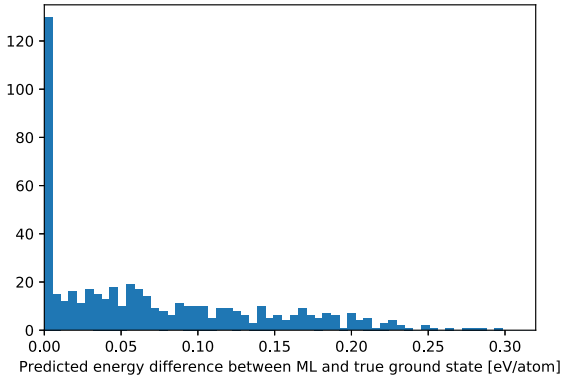


FIG. 13. Energy difference between the ML-predicted ground state and the true ground state $\Delta E = E^{\mathrm{ML}}(G_{\mathrm{DFT}}) - E^{\mathrm{ML}}(G_{\mathrm{ML}})$ for the selenide data set. The mean absolute difference is 70 meV/atom.

the 512 compositions, the model predicts the DFT ground state. This is not particularly impressive since random prediction of a structure would give roughly $512/12 \approx 43$ correct predictions. However, the data set has many low-lying energy structures, where even full DFT calculations cannot be expected to necessarily predict the correct ground-state structure. This was investigated in more detail in a similarly generated data set of ABS_3 sulfides used for computational screening of water-splitting materials [3]. The mean absolute difference is only 70 meV/atom with a maximum error of 0.3 eV/atom. The low mean value is clearly promising for future applications to computational materials screening.

VII. CONCLUSIONS

In summary, we have proposed a ML model for the prediction of the formation energy of crystalline materials based on Voronoi quotient graphs and a local symmetry description. It uses a message passing neural network with edge updates. The model is independent of the detailed atomic positions and can therefore be used to predict the formation energy of new materials, where the detailed structure is unknown.

The model test MAE is very small (22 meV) on the OQMD data set, and a factor of 2 larger (40 meV) on the ICSD subset of OQMD. To test the model in a realistic materials screening setting, we created a data set of 6000 selenides with very small overlap with the OQMD. The model pretrained on OQMD and applied to the selenides shows an MAE of 176 meV. This value can be lowered to 95 meV with an additional training on 100 selenides. Further training can lower the MAE to below 50 meV.

Based on the results, we conclude that is possible to develop ML models with position independent descriptors, which are useful for realistic materials screening studies. However, extrapolation from OQMD to other data sets is challenging. One reason for this may be, as pointed out before, that the OQMD is composed of materials of two types: some are generated systematically in rather few predefined crystal structures while others come from ICSD. (There is of course a significant overlap between the two types.) The first type is characterized by a large variation in stability, but low variation in crystal structures, while the second type is the opposite: the experimentally observed materials in ICSD exhibit a large variation in crystal structures, but they are all (except for some high-pressure entries) stable low-energy materials. This bias might limit the extrapolation to data sets which contain structures weakly represented in OQMD and with element combinations, which are far from stable. One way forward could be to create data sets with less bias so that unstable materials are represented in a greater variety of structures.

We see a number of potential improvements of the proposed model. More symmetry information could be included using, for example, Wyckoff positions [22] or additional graph edges describing symmetry relations. Furthermore, it is possible to label the quotient graphs with crystal translation information so that the infinite graph can be reconstructed [41]. This would make the crystal description more unique.

Perhaps the model could also learn the atomic positions from the graph representation. The latest developments in generative models have succeeded in generating small molecules in three-dimensional (3D) space [42]. By combining this kind of model with the restrictions imposed by the connectivity and symmetries described by the quotient graph (see, for example, [43,44]), it might be possible to directly predict the atomic positions without running DFT relaxations.

Another useful extension would be to model uncertainties in the prediction. Even though the data sets used here have a relatively high number of entries, they only contain a tiny fraction of the chemical space. If the model could learn what it does not know, it would be very useful in an active learning setting where DFT calculations could be launched by the model to explore areas of the chemical space with high uncertainty. A promising direction for uncertainty modeling is to use ensembles of neural networks where different techniques can be considered to ensure diversity between ensemble members [45–48].

ACKNOWLEDGMENTS

We would like to thank P. Mahler Larsen for helpful discussions. We acknowledge support from the VILLUM Center for Science of Sustainable Fuels and Chemicals which is funded by the VILLUM Fonden research Grant No. 9455 and thanks to Nvidia for the donation of one Titan X GPU.

[1] I. E. Castelli, T. Olsen, S. Datta, D. D. Landis, S. Dahl, K. S. Thygesen, and K. W. Jacobsen, Energy Environ. Sci. **5**, 5814 (2012).

[2] Y. Wu, P. Lazic, G. Hautier, K. Persson, and G. Ceder, Energy Environ. Sci. **6**, 157 (2012).

[3] K. Kuhar, A. Crovetto, M. Pandey, K. S. Thygesen, B. Seger, P. C. K. Vesborg, O. Hansen, I. Chorkendorff, and K. W. Jacobsen, Energy Environ. Sci. **10**, 2579 (2017).

[4] A. Urban, D.-H. Seo, and G. Ceder, npj Comput. Mater. **2**, 16002 (2016).

[5] M. Aykol, S. Kim, V. I. Hegde, D. Snydacker, Z. Lu, S. Hao, S. Kirklin, D. Morgan, and C. Wolverton, Nat. Commun. **7**, 13779 (2016).

[6] M. Andersson, T. Bligaard, A. Kustov, K. Larsen, J. Greeley, T. Johannessen, C. Christensen, and J. K. Nørskov, J. Catal. **239**, 501 (2006).

[7] J. K. Nørskov, T. Bligaard, J. Rossmeisl, and C. H. Christensen, Nat. Chem. **1**, 37 (2009).

[8] N. Mounet, M. Gibertini, P. Schwaller, D. Campi, A. Merkys, A. Marrazzo, T. Sohier, I. E. Castelli, A. Cepellotti, G. Pizzi, and N. Marzari, Nat. Nanotechnol. **13**, 246 (2018).

[9] S. Curtarolo, G. L. W. Hart, M. B. Nardelli, N. Mingo, S. Sanvito, and O. Levy, Nat. Mater. **12**, 191 (2013).

[10] K. Alberi, M. B. Nardelli, A. Zakutayev, L. Mitas, S. Curtarolo, A. Jain, M. Fornari, N. Marzari, I. Takeuchi, M. L. Green, M. Kanatzidis, M. F. Toney, S. Butenko, B. Meredig, S. Lany, U. Kattner, A. Davydov, E. S. Toberer, V. Stevanovic, A. Walsh, N.-G. Park *et al.*, J. Phys. D: Appl. Phys. **52**, 013001 (2018).

[11] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).

[12] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).

[13] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller, and E. K. U. Gross, Phys. Rev. B **89**, 205118 (2014).

[14] F. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armiento, Int. J. Quantum Chem. **115**, 1094 (2015).

[15] L. Ward, R. Liu, A. Krishna, V. I. Hegde, A. Agrawal, A. Choudhary, and C. Wolverton, Phys. Rev. B **96**, 024104 (2017).

[16] C. Oses, C. Toher, E. Gossett, A. Tropsha, O. Isayev, and S. Curtarolo, Nat. Commun. **8**, 1 (2017).

[17] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, Nat. Commun. **8**, 13890 (2017).

[18] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, J. Chem. Phys. **148**, 241722 (2018).

[19] T. Xie and J. C. Grossman, Phys. Rev. Lett. **120**, 145301 (2018).

[20] P. Zaspel, B. Huang, H. Harbrecht, and O. A. von Lilienfeld, J. Chem. Theory Comput. **15** 1546 (2019).

[21] B. Meredig, A. Agrawal, S. Kirklin, J. E. Saal, J. W. Doak, A. Thompson, K. Zhang, A. Choudhary, and C. Wolverton, Phys. Rev. B **89**, 094104 (2014).

[22] A. Jain and T. Bligaard, Phys. Rev. B **98**, 214112 (2018).

[23] S. J. Chung, T. Hahn, and W. E. Klee, Acta Crystallogr. Sec. A **40**, 42 (1984).

[24] E. A. Lazar, J. Han, and D. J. Srolovitz, Proc. Natl. Acad. Sci. USA **112**, E5769 (2015).

[25] D. Reem, *Proceedings of the Twenty-seventh Annual Symposium on Computational Geometry, SoCG '11, Paris, France* (ACM, New York, NY, 2011), pp. 254–263.

[26] A. Malins, S. R. Williams, J. Eggers, and C. P. Royall, J. Chem. Phys. **139**, 234506 (2013).

[27] H. J. A. M. Heijmans and A. V. Tuzikov, IEEE Trans. Pattern Anal. Machine Intell. **20**, 980 (1998).

[28] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, *International Conference on Machine Learning* (IEEE, Piscataway, NJ, 2017), pp. 1263–1272.

[29] P. B. Jørgensen, K. W. Jacobsen, and M. N. Schmidt, arXiv:1806.03146.

[30] K. He, X. Zhang, S. Ren, and J. Sun, arXiv:1512.03385.

[31] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, APL Mater. **1**, 011002 (2013).

[32] G. Bergerhoff, R. Hundt, R. Sievers, and I. D. Brown, J. Chem. Inf. Comput. Sci. **23**, 66 (1983).

[33] J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton, JOM **65**, 1501 (2013).

[34] S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl, and C. Wolverton, npj Comput. Mater. **1**, 15010 (2015).

[35] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson *et al.*, J. Phys.: Condens. Matter **29**, 273002 (2017).

[36] J. Enkovaara, C. Rostgaard, J. J. Mortensen, J. Chen, M. Dułak, L. Ferrighi, J. Gavnholt, C. Glinsvad, V. Haikola, H. A. Hansen, H. H. Kristoffersen, M. Kuisma, A. H. Larsen, L. Lehtovaara, M. Ljungberg, O. Lopez-Acevedo, P. G. Moses, J. Ojanen, T. Olsen, V. Petzold *et al.*, J. Phys.: Condens. Matter **22**, 253202 (2010).

[37] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, O. A. Vydrov, G. E. Scuseria, L. A. Constantin, X. Zhou, and K. Burke, Phys. Rev. Lett. **100**, 136406 (2008).

[38] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[39] H. J. Monkhorst and J. D. Pack, Phys. Rev. B **13**, 5188 (1976).

[40] D. P. Kingma and J. Ba, arXiv:1412.6980.

[41] W. E. Klee, Cryst. Res. Technol. **39**, 959 (2004).

[42] N. W. A. Gebauer, M. Gastegger, and K. T. Schütt, arXiv:1810.11347.

[43] G. Thimm, Acta Crystallogr., Sect. A **65**, 213 (2009).

[44] J.-G. Eon, Acta Crystallogr., Sect. A **67**, 68 (2011).

[45] A. A. Peterson, R. Christensen, and A. Khorshidi, Phys. Chem. Chem. Phys. **19**, 10978 (2017).

[46] B. Lakshminarayanan, A. Pritzel, and C. Blundell, in *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Red Hook, NY, 2017), pp. 6402–6413.

[47] T. Pearce, N. Anastassacos, M. Zaki, and A. Neely, arXiv:1805.11324.

[48] I. Osband, J. Aslanides, and A. Cassirer, arXiv:1806.03335.

Paper II

Local Bayesian optimizer for atomic structures

Estefanía Garijo del Río, Jens Jørgen Mortensen, and Karsten Wedel Jacobsen

Phys. Rev. B **100**, 104103 – Published 5 September 2019

Local Bayesian optimizer for atomic structures

Estefanía Garijo del Río, Jens Jørgen Mortensen, and Karsten Wedel Jacobsen
CAMD, Department of Physics, Technical University of Denmark, 2800 Kongens Lyngby, Denmark

(Received 11 July 2019; published 5 September 2019)

A local optimization method based on Bayesian Gaussian processes is developed and applied to atomic structures. The method is applied to a variety of systems including molecules, clusters, bulk materials, and molecules at surfaces. The approach is seen to compare favorably to standard optimization algorithms like the conjugate gradient or Broyden-Fletcher-Goldfarb-Shanno in all cases. The method relies on prediction of surrogate potential energy surfaces, which are fast to optimize, and which are gradually improved as the calculation proceeds. The method includes a few hyperparameters, the optimization of which may lead to further improvements of the computational speed.

DOI: 10.1103/PhysRevB.100.104103

I. INTRODUCTION

One of the great successes of density functional theory (DFT) [1,2] is its ability to predict ground-state atomic structures. By minimizing the total energy, the atomic positions in solids or molecules at low temperatures can be obtained. However, the optimization of atomic structures with density functional theory or higher-level quantum chemistry methods require substantial computer resources. It is therefore important to develop new methods to perform the optimization efficiently.

It is of key interest here that for a given atomic structure a DFT calculation provides not only the total electronic energy but also, at almost no additional computational cost, the forces on the atoms, i.e., the derivatives of the energy with respect to the atomic coordinates. This means that for a system with N atoms in a particular configuration only a single energy value is obtained while $3N$ derivatives are also calculated. It is therefore essential to include the gradient information in an efficient optimization.

A number of well-known function optimizers exploring gradient information exist [3] and several are implemented in standard libraries like the SciPy library [4] for use in Python. Two much-used examples are the conjugate gradient (CG) method and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. Both of these rely on line minimizations and perform particularly well for a nearly harmonic potential energy surface (PES). In the CG method, a series of conjugated search directions are calculated, while the BFGS method gradually builds up information about the Hessian, i.e., the second derivatives of the energy, to find appropriate search directions.

The Gaussian process (GP) method that we are going to present has the benefit that it produces smooth surrogate potential energy surfaces (SPESs) even in regions of space where the potential is nonharmonic. This leads to a generally improved convergence. The number of algebraic operations that has to be carried out in order to move from one atomic structure to the next is much higher for the GP method than for the CG or BFGS methods; however, this is not of concern for optimizing atomic structures with DFT, because the electronic structure calculations themselves are so time consuming. For more general optimization problems where the function evaluations are fast, the situation may be different.

Machine learning for PES modeling has recently attracted the attention of the materials modeling community [5–18]. In particular, several methods have focused on fitting the energies of electronic structure calculations to expressions of the form

$$E(\rho) = \sum_{i=1}^{n} \alpha_i \, k(\rho^{(i)}, \rho). \tag{1}$$

Here, $\{\rho^{(i)}\}_{i=1}^{n}$ are some descriptors of the n atomic configurations sampled, $k(\rho^{(i)}, \rho)$ is known as a kernel function, and $\{\alpha_i\}_{i=1}^{n}$ are the coefficients to be determined in the fit. Since there are n coefficients and n free parameters, the SPES determined by this expression has the values of the calculations at the configurations on the training set.

Here we note that expression (1) can easily be extended to

$$E(\rho) = \sum_{i=1}^{n} \alpha_i \, k(\rho^{(i)}, \rho) + \sum_{i=1}^{n} \sum_{j=1}^{3N} \beta_{ij} \frac{\partial k(\rho^{(i)}, \rho)}{\partial r_j^{(i)}}, \tag{2}$$

where $\{r_j^{(i)}\}_{j=1}^{3N}$ represent the coordinates of the N atoms in the ith configuration. The new set of parameters β_{ij} together with α_i can be adjusted so that not only the right energy of a given configuration $\rho^{(i)}$ is predicted, but also the right forces. This approach has two advantages with respect to the previous one: (i) the information included in the model scales with the dimensionality; (ii) the new model is smooth and has the right gradients.

In the case of systems with many identical atoms or similar local atomic structures it becomes advantageous to construct SPESs based on descriptors or fingerprints characterizing the local environment [5–11]. The descriptors can then be constructed to obey basic principles as rotational and translational symmetries and invariance under exchange of identical atoms. Here we shall develop an approach based on Gaussian processes which works directly with the atomic

coordinates and effectively produces a surrogate PES of the type Eq. (2) aimed at relaxing atomic structures. We note that Gaussian processes with derivatives for PES modeling are a field that is developing fast, with recent applications in local optimization [19] and path determination in elastic band calculations [13,20,21].

II. GAUSSIAN PROCESS REGRESSION

We use Gaussian process regression with derivative information to produce a combined model for the energy E and the forces $\mathbf{f}$ of a configuration with atomic positions $\mathbf{x} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)$:

$$\mathbf{U}(\mathbf{x}) = (E(\mathbf{x}), -\mathbf{f}(\mathbf{x})) \sim \mathcal{GP}(\mathbf{U}_p(\mathbf{x}), K(\mathbf{x}, \mathbf{x}')), \tag{3}$$

where $\mathbf{U}_p(\mathbf{x}) = (E_p(\mathbf{x}), \nabla E_p(\mathbf{x}))$ is a vector-valued function which constitutes the prior model for the PES and $K(\mathbf{x}, \mathbf{x}')$ is a matrix-valued kernel function that models the correlation between pairs of energy and force values as a function of the configuration space.

In this work, we choose the constant function $\mathbf{U}_p(\mathbf{x}) = (E_p, \mathbf{0})$ as the prior function. For the kernel, we use the squared-exponential covariance function to model the correlation between the energy of different configurations:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 e^{-\|\mathbf{x}-\mathbf{x}'\|^2/2l^2}, \tag{4}$$

where l is a typical scale of the problem and σ_f is a parameter describing the prior variance at any configuration $\mathbf{x}$. The full kernel K can be obtained by noting that [22,23]

$$\operatorname{cov}(E(\mathbf{x}), E(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}'), \tag{5}$$

$$\operatorname{cov}\left(E(\mathbf{x}), \frac{\partial E(\mathbf{x}')}{\partial x_i'}\right) = \frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial x_i'} \equiv J_i(\mathbf{x}, \mathbf{x}'), \tag{6}$$

$$\operatorname{cov}\left(\frac{\partial E(\mathbf{x})}{\partial x_i}, \frac{\partial E(\mathbf{x}')}{\partial x_j'}\right) = \frac{\partial^2 k(\mathbf{x}, \mathbf{x}')}{\partial x_i \partial x_j'} \equiv H_{ij}(\mathbf{x}, \mathbf{x}'), \tag{7}$$

and assembling these covariance functions in a matrix form:

$$K(\mathbf{x}, \mathbf{x}') = \begin{pmatrix} k(\mathbf{x}, \mathbf{x}') & \mathbf{J}(\mathbf{x}, \mathbf{x}') \\ \mathbf{J}(\mathbf{x}', \mathbf{x})^T & H(\mathbf{x}, \mathbf{x}') \end{pmatrix}. \tag{8}$$

The expressions for the mean and the variance for the posterior distribution follow the usual definitions incorporating the additional matrix structure. Let $X = \{\mathbf{x}^{(i)}\}_{i=1}^n$ denote the matrix containing n training inputs and let $Y = \{\mathbf{y}^{(i)}\}_{i=1}^n = \{(E(\mathbf{x}^{(i)}), -\mathbf{f}(\mathbf{x}^{(i)}))\}_{i=1}^n$ be the matrix containing the corresponding training targets. By defining

$$K(\mathbf{x}, X) = (K(\mathbf{x}, \mathbf{x}^{(1)}), K(\mathbf{x}, \mathbf{x}^{(2)}), \ldots, K(\mathbf{x}, \mathbf{x}^{(n)})) \tag{9}$$

and

$$(K(X, X))_{ij} = K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}), \tag{10}$$

we get the following expressions for the mean,

$$\begin{aligned} \bar{\mathbf{U}}(\mathbf{x}) &= (\bar{E}(\mathbf{x}), -\bar{\mathbf{f}}(\mathbf{x})) \\ &= \mathbf{U}_p(\mathbf{x}) + K(\mathbf{x}, X)\mathbb{K}_X^{-1}(Y - \mathbf{U}_p(X)), \end{aligned} \tag{11}$$

and the variance,

$$\boldsymbol{\sigma}^2(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}) - K(\mathbf{x}, X)\mathbb{K}_X^{-1}K(X, \mathbf{x}), \tag{12}$$

of the prediction, where $\mathbb{K}_X = K(X, X) + \Sigma_n^2$. Here, we have assumed an additive Gaussian noise term with covariance matrix Σ_n [22]. This term corrects only for the self-covariance of the points in the training set, and thus, it is a diagonal matrix that models the self-correlation of forces with a hyperparameter σ_n^2 and the self-correlation of energies with $\sigma_n^2 \times l^2$. We note that even for computational frameworks where the energy and forces can be computed with very limited numerical noise, small nonzero values of σ_n are advantageous since they prevent the inversion of the covariance matrix $K(X, X)$ from being numerically ill conditioned [13].

In the following, we will refer to $\bar{E}(\mathbf{x})$ as defined in Eq. (11) as the surrogate potential energy surface (SPES) and distinguish it from the first-principles PES, $E(\mathbf{x})$.

III. GAUSSIAN PROCESS MINIMIZER: GPMin

The GP framework can be used to build an optimization algorithm. In this section, we introduce the main ideas behind the proposed Gaussian process minimizer (denoted GPMin from hereon). A more detailed description of the algorithm can be found in the Appendix in the form of a pseudocode.

The GP regression provides a SPES that can be minimized using a gradient-based local optimizer. For this purpose, we have used the L-BFGS-B algorithm as implemented in SciPy [24]. The prior value for the energy is initially set as the energy of the initial configuration and then the expression (11) is used to produce a SPES from that data point alone. This model is then minimized, and the evaluation at the new local minimum generates new data that is then fed into the model to produce a new SPES that will have a different local minimum. Before generating each new SPES the prior value for the energy is updated to the maximum value of the energies previously sampled. This step is important because it makes the algorithm more stable. If a high-energy configuration is sampled, the forces may be very large leading to a too large new step. The increase of the prior value tends to dampen this by effectively reducing the step size. The whole process is then iterated until convergence is reached.

It is illustrative to consider in more detail the first step of the algorithm. It is straightforward to show using Eqs. (4)–(11) that if only a single data point $\mathbf{x}^{(1)}$ is known the SPES is given by

$$\bar{E}(\mathbf{x}) = E^{(1)} - \mathbf{f}^{(1)} \cdot (\mathbf{x} - \mathbf{x}^{(1)}) e^{-\|\mathbf{x}-\mathbf{x}^{(1)}\|^2/2l^2}, \tag{13}$$

where $E^{(1)}$ and $\mathbf{f}^{(1)}$ are the energy and forces of the SPES at the point $\mathbf{x}^{(1)}$, respectively. We have here used that the prior energy is set to the energy of the first configuration $E^{(1)}$. One can confirm that this is the prior energy by noting that points far away from $\mathbf{x}^{(1)}$, where no information is available, take on this value for the energy. It is seen that the initial force $\mathbf{f}^{(1)}$ gives rise to a Gaussian depletion of the SPES. The first step of the GPMin algorithm minimizes the SPES leading to a new configuration,

$$\mathbf{x} = \mathbf{x}^{(1)} + l\frac{\mathbf{f}^{(1)}}{\|\mathbf{f}^{(1)}\|}. \tag{14}$$

The first step is thus in the direction of the force with a step length of l. Considering the information available this is a very natural choice.

GPMin depends on a number of parameters: the length scale l, the prior value of the energy E_p, the energy width σ_f, and the noise or regularization parameter σ_n. It can be seen from expressions (4) and (11) that the prediction of the SPES depends only on the ratio of σ_f and σ_n and not their individual values.

The prior energy E_p is, as explained above, taken initially as the energy of the first configuration and then updated if larger energies are encountered. It is important that the prior value is not too low to avoid large steps, since the prior energy is the value of the SPES for all configurations far away (on the scale of l) from previously investigated structures.

The scale l is very important as it sets the distance over which the SPES relaxes back to the prior value E_p when moving away from the region of already explored configurations. It therefore also effectively determines a step length in the algorithm.

One interesting advantage of the Bayesian approach is that it allows for update of parameters (usually termed hyperparameters) based on existing data. We investigate this option by allowing the value of the length scale l to change. Since the update procedure also depends on the width parameter σ_f, we update this as well. The updated hyperparameters, $\boldsymbol{\theta} = (l, \sigma_f)$, are determined by maximizing the marginal likelihood:

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\vartheta}} P(Y|X, \boldsymbol{\vartheta}). \tag{15}$$

The optimization may fail, for example if there is not enough evidence and the marginal likelihood is very flat, and if that happens, the previous scale is kept. The update procedure allows the algorithm to find its own scale as it collects more information, producing a model that self-adapts to the problem at hand. In Sec. VI we shall consider in more depth the adequate choices for the values of the hyperparameters and the different strategies for the update of hyperparameters when the optimizers are applied to DFT calculations.

IV. COMPUTATIONAL DETAILS

We illustrate and test the method on a variety of different systems using two different calculation methods: An interatomic effective medium theory potential (EMT) [25,26] as implemented in ASE [27,28] and DFT. The DFT tests have been performed using GPAW [29] with the local density approximation (LDA) exchange-correlation functional and a plane wave basis set with an energy cutoff at 340 eV. The Brillouin zone has been sampled using the Monkhorst-Pack scheme with a k-point density of $2.0/(\text{Å}^{-1})$ in all three directions. The PAW setup with one valence electron has been used for the sodium cluster for simplicity. In addition to the default convergence criteria for GPAW, we specify that the maximum change in magnitude of the difference in force for each atom should be smaller than 10^{-4} eV Å^{-1} for the self-consistent field iteration to terminate. This improves the convergence of the forces. All systems have been relaxed until the maximum force of the atoms was below 0.01 eV Å^{-1}.

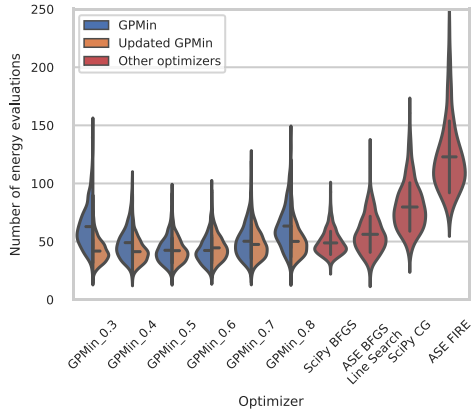


FIG. 1. Statistics of the number of energy evaluations for 1000 relaxations of a 10-atom gold cluster. The initial conditions have been randomly generated. The left-hand side of the plot shows the distribution of the number of energy evaluations for GPMin in its two variants for scales ranging from 0.3 to 0.8 Å: keeping the scale fixed or allowing it to be updated. The right-hand side shows the performance of other widely used optimizers, which have been sorted according to the average number of function evaluations.

V. EXAMPLE: GOLD CLUSTERS DESCRIBED IN EFFECTIVE MEDIUM THEORY

In the first example GPMin is used to find the structure of 10-atom gold clusters as described by the EMT potential, and the efficiency is compared with other common optimizers. For this purpose, we generate 1000 random configurations of a 10-atom gold cluster. The configurations are constructed by sequentially applying three uniform displacements for each atom in a cubic box with side length 4.8 Å and only keeping those that lie farther than 1.7 times the atomic radius of gold away from any of the other atoms already present in the cluster. Each configuration is then optimized with different choices of parameters for GPMin, and, for comparison, the same structures are optimized with the ASE implementations of FIRE [30] and BFGS Line Search, and the SciPy implementations of BFGS and the CG.

For the gold clusters, we have investigated the effect of updating σ_f and l for six different initial scales between 0.3 and 0.8 Å and initial $\sigma_f = 1.0$ eV. Since the EMT potential has very small numerical noise, we choose a small value of $\sigma_n/\sigma_f = 5 \times 10^{-4}$ eV Å^{-1} for the regularization. In the update version of the optimizer, we update the scale every fifth iteration.

The statistics of the number of energy evaluations are shown in Fig. 1. The GP optimizers are seen to be the fastest on average, with the appropriate choice of the hyperparameters. For the initial scale of 0.5 Å, for example, the updated version of GPMin had relaxed the clusters after 42.1 ± 0.3 energy evaluations and the nonupdated one after 42.5 ± 0.3, as compared to 48.8 ± 0.3 and 56.2 ± 0.5 for the BFGS implementations in SciPy and ASE, respectively. CG exhibits an average number of steps of 79.7 ± 0.7, and FIRE, 122.9 ± 1.0.

Figure 1 shows the trend in the performance as the scale is varied. For this system, $l = 0.5$ Å has the lowest average and variance for GPMin. The performance depends rather sensitively on the scale parameter: reducing the scale results in a more conservative algorithm where more but smaller steps are needed. Increasing the scale leads to a more explorative algorithm with longer steps that may fail to reduce the energy. In the algorithm with updates, the scale is automatically modified to compensate for a nonoptimal initial scale. The update is particularly efficient for small scales where the local environment is sufficiently explored. For larger scales the sampling is less informative and it takes longer for the algorithm to reduce the scale.

We note that under the appropriate choice of scale, both GPMin with and without update are among the fastest for the best-case scenario, with 18 evaluations for the regular GPMin optimizer and 19 for the updated version with scale $l = 0.5$ Å, compared to 19 for ASE BFGS, 27 and 34 for the SciPy implementations of BFGS and CG, respectively, and 70 for FIRE. We further note that the updated version has by far the best worst-case performance.

Of the total of 18 000 relaxations, only 17 failed to find a local minimum. These 17 relaxations were all run with the GPMin optimizer with $l = 0.8$ Å without the updates. An optimizer with a too long scale fails to build a successful SPES: the minimum of the SPES often has a higher energy than the previously evaluated point. Thus, we consider that the optimization has failed if after 30 such catastrophic attempts, the optimizer has still not been able to identify a point that reduces the energy or if SciPy's BFGS cannot successfully optimize the predicted SPES.

VI. DETERMINATION OF THE HYPERPARAMETERS

We now continue by considering the use of the GP optimizers more generally for systems with PESs described by DFT. Default values of the hyperparameters should be chosen such that the algorithm performs well for a variety of atomic systems. For this purpose, we have chosen a training set consisting of two different structures: (i) a 10-atom sodium cluster with random atomic positions and (ii) a carbon dioxide molecule on a (111) surface with two layers of gold and a 2×2 unit cell. We have generated 10 slightly different initial configurations for each of the training systems by adding random numbers generated from a Gaussian distribution with standard deviation 0.1 Å. The training configurations are then relaxed using DFT energies and forces.

For each pair of the hyperparameters $(l, \sigma_n/\sigma_f)$, we relax the training systems and average over the number of DFT evaluations the optimizer needs to find a local minimum. The results are shown in Fig. 2. The plot shows that the metallic cluster benefits from relatively large scales, while the CO on gold system with a tight CO bond requires a shorter scale. A too long scale might even imply that the optimizer does not converge. The set of hyperparameters $l = 0.4$ Å, $\sigma_n = 1$ meV Å^{-1}, and $\sigma_f = 1$ eV seems to be a good compromise between the two cases and these are the default values we shall use in the following.

A similar procedure has been used to determine the default values of the hyperparameters and their initial values in the updated versions of GPMin. Here, the hyperparameter σ_n/σ_f is kept fixed during the optimization, whereas l and σ_f are determined using expression (15). The value of σ_n/σ_f and the initial values of the other hyperparameters are then determined from the analysis of the performance of the optimizer on the two systems in the training set. The evolution of the hyperparameters depends on the details of the optimization of the marginal likelihood together with the frequency at which the hyperparameters are optimized. Here, we explore three different strategies: Unconstrained maximization of the marginal log-likelihood every 5 energy evaluations ("GPMin-5"), and two constrained optimization strategies, where the outcome of the optimization is constrained to vary in the range $\pm 10\%$ and $\pm 20\%$ of the value of the hyperparameter in the previous step ("GPMin-10%" and "GPMin-20%," respectively). In the latter two cases we let the optimization take place whenever new information is added to the sample. The algorithm used to maximize the marginal log-likelihood is L-BFGS-B [24] for all strategies.

We have relaxed the same 10 slightly different copies of the two training set systems described before using these three strategies for three different initial values of the scale (0.2,

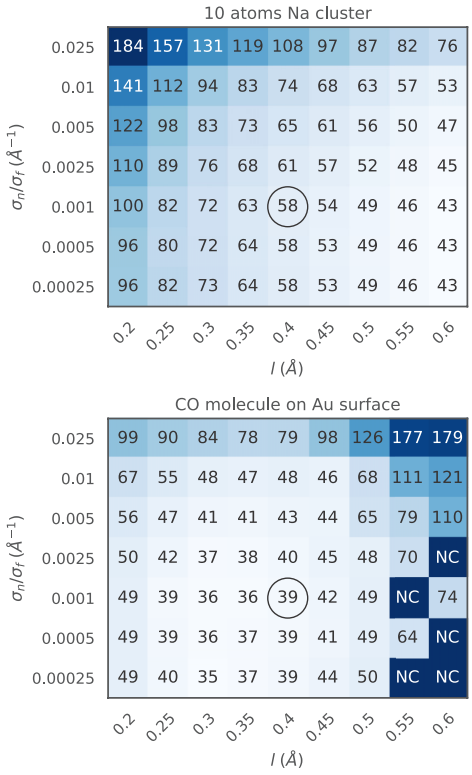


FIG. 2. Average number of potential energy evaluations needed to relax 10 atomic structures as a function of the two hyperparameters: the length scale l, and the regularization parameter σ_n. The label NC (not converged) indicates that at least one of the relaxations did not converge. The default choices for the hyperparameters are indicated by circles.

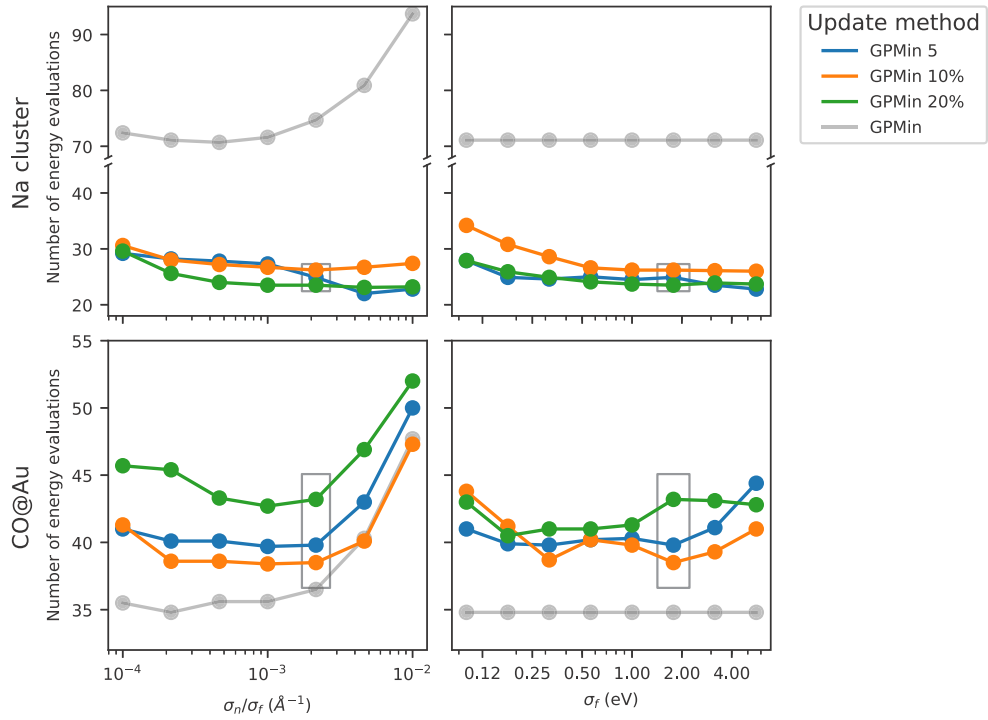


FIG. 3. Average number of energy evaluations needed to relax the two training set systems as a function of the hyperparameter σ_n/σ_f or of the initial value of σ_f, while the other one is kept fixed. The results are shown for the three different updating strategies and compared with the result of running GPMin without update with the same choice of hyperparameters. The rectangles show the values of the hyperparameter that have been chosen as default values. The value of σ_f chosen in the right panels has been used in the relaxations shown in the left panels, and similarly, the value of σ_n/σ_f that has been found optimal in the left panel is the one that has been used in the relaxations in the right panel.

0.3, and 0.4 Å), eight different initial values of σ_f, and seven different values of the regularization parameter σ_n/σ_f. An overview of the full results can be found in the Supplemental Material [31].

The average numbers of energy evaluations needed to relax the training set for the different strategies and hyperparameters are shown in Fig. 3. The initial value of the scale is chosen as 0.3 Å. The plot shows the variation of the average number of energy evaluations with σ_n/σ_f when the initial value of $\sigma_f = 1.8$ eV and the variation with σ_f when the value of $\sigma_n/\sigma_f = 2 \times 10^{-3}$ Å^{-1}. The performance of the optimizers is seen to depend rather weakly on the parameter values in particular for the sodium cluster. We shall therefore in the following use the values $\sigma_f = 1.8$ eV and $\sigma_n/\sigma_f = 2 \times 10^{-3}$ Å^{-1}.

From the figure it can also be seen that the versions of the optimizer with updates perform considerably better than GPMin without updates for the sodium cluster, while for the CO molecule on gold, the version without update works slightly better than the three optimizers with updates.

To understand this behavior further we consider in Fig. 4 the evolution of the length scale l as it is being updated. The scale is initially set at three different values $l = 0.2, 0.3, 0.4$ Å. For the sodium cluster the update procedure quickly leads to a much longer length scale around 1.5 Å. For GPMin-5 the length scale is raised dramatically already at the first update after five energy evaluations, while for GPMin-10% and GPMin-20% the length scale increases gradually because of the constraint built into the methods. The advantage of a longer length scale is in agreement with the results above for the gold cluster described with the EMT interatomic interactions, where a long length scale also led to faster convergence. The situation is different for the CO/Au system, where the update leads first to a significant decrease in the scale and later to an increase saturating at a value around 0.3 Å. This result was to be expected from the one shown in Fig. 2 for the performance of GPMin without the hyperparameter update. We interpret the variation of the scale for the CO/Au system as being due to the different length scales present in the system, where the CO bond is short and strong while the metallic bonds are much longer. In the first part of the optimization the CO configuration is modified requiring a short scale, while the later stages involve the CO-metal and metal-metal distances. Overall the update of the scale does not provide an advantage over the GPMin without updates where the scale is kept fixed at $l = 0.4$ Å. It can be seen that the final scales obtained, for example in the case of the sodium cluster optimized with GPMin-10%, vary by about 30%, where the variation depends on the particular system being optimized and not on the initial value for the length scale.

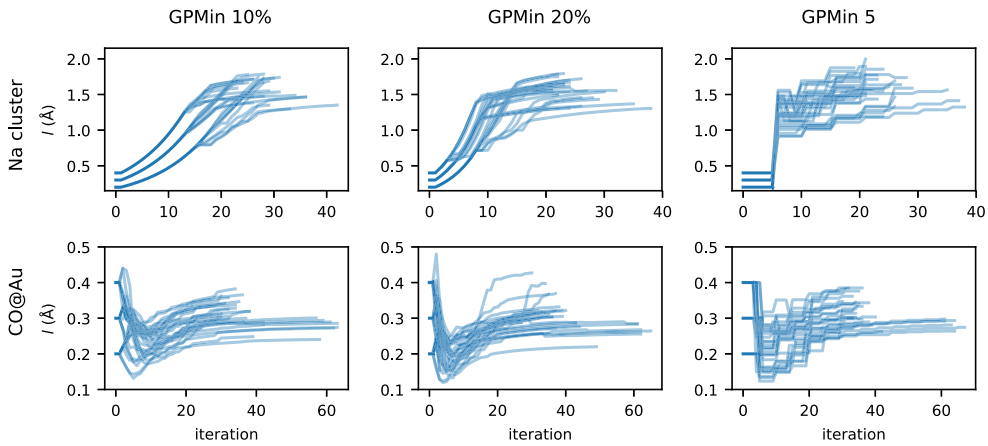


FIG. 4. Evolution of the length scale l with iteration for the three optimizers with update GPMin-5, GPMin-10%, and GPMin-20%. The upper panel in each case shows the results for the sodium cluster, while the lower panel shows the evolution for the CO/Au system. In all cases three different values $l = 0.2, 0.3, 0.4$ Å for the initial scale have been considered. For the sodium cluster the length scale is seen to increase significantly, while in the case of the CO/Au system, the length scale first decreases and then subsequently increases. The final length scale varies by about 30% dependent on the particular initial structure of the systems.

In the following we shall use $l = 0.3$ Å as the initial scale for the optimizers with updates. As shown in Figs. S1, S2, and S3 in the Supplemental Material [31], the results do not depend very much on the initial scale in the range 0.2–0.4 Å. Furthermore, the results for the EMT gold cluster indicate that long length scales should be avoided: it is easier for the algorithm to increase the length scale than to decrease it.

To summarize, we select the following default (initial) values of the hyperparameters for the updated versions of GPMin: $l = 0.3$ Å, $\sigma_f = 2.0$ eV, and $\sigma_n = 0.004$ eV Å^{-1} ($\sigma_n/\sigma_f = 0.002$ Å^{-1}). These values are used in the rest of this paper.

VII. RESULTS

To test the Bayesian optimizers we have investigated their performance for seven different systems with DFT: a CO molecule on a Ag(111) surface, a C adsorbate on a Cu(100) surface, a distorted Cu(111) surface, bulk copper with random displacements of the atoms with Gaussian distribution and width 0.1 Å, an aluminum cluster with 13 atoms in a configuration close to fcc, the H_2 molecule, and the pentane molecule. All surfaces are represented by two-layer slabs with a 2×2 unit cell and periodic boundary conditions along the slab. The bulk structure is represented by a $2 \times 2 \times 2$ supercell with periodic boundary conditions along the three unit cell vectors. For each of the systems we have generated ten slightly different initial configurations by rattling the atoms by 0.1 Å. The resulting configurations are then relaxed using the ASE and SciPy optimizers, together with the different GPMin optimizers.

It should be noted that in a few cases an optimizer fails to find a local minimum: an atomic configuration is suggested for which GPAW raises an error when it attempts to compute the energy, because two atoms are too close. This happens for SciPy's BFGS for one of the CO/Ag configurations and for SciPy's conjugate gradient method for one of the hydrogen molecule configurations.

The results are collected in Fig. 5. For the sake of clarity, ASE FIRE has been excluded from the plot, since it takes about a factor of three more steps than the fastest optimizer for all systems. The average number of DFT evaluations for the relaxation of the systems in the test set with the implementation of FIRE in ASE is 122 ± 4 for CO/Ag, 91 ± 5 for the pentane molecule, 58 ± 4 for C/Cu, 85 ± 3 for the aluminum cluster, 62 ± 2 for the Cu slab, 53 ± 1 for Cu bulk, and 30 ± 3 for the H_2 molecule.

The GP optimizers are seen to compare favorably or on par with the best one of the other optimizers in all cases. GPMin without update is on average faster than the other optimizers for 6 of the 7 systems. For the bulk Cu system, it is only slightly slower than the ASE-BFGS algorithm. The updated GP optimizers perform even better with one exception: GPMin-5 is clearly worse than the other GP optimizers and ASE BFGS for the copper bulk system. Since the atomic displacements from the perfect crystal structure are quite small ($\sim$0.1 Å), this system is probably within the harmonic regime and requires only a few ($\sim$10) iterations to converge. The ASE BFGS can therefore be expected to perform well, which is also what is observed in Fig. 5. GPMin-5 does not update the scale for the first 5 iterations, and when it does so, the new scale does not lead to immediate convergence. The plain GPMin and the two other optimizers with updates perform on par with ASE BFGS.

Generally, the updated optimizers perform better than GPMin without updates, and both GPMin-10% and GPMin-20% with constrained update perform consistently very well. The updated optimizers are clearly better than the plain GPMin for the Al cluster, similarly to the behavior for the Na cluster used in the determination of hyperparameters. For the other training system, the CO/Au system, GPMin was seen to perform

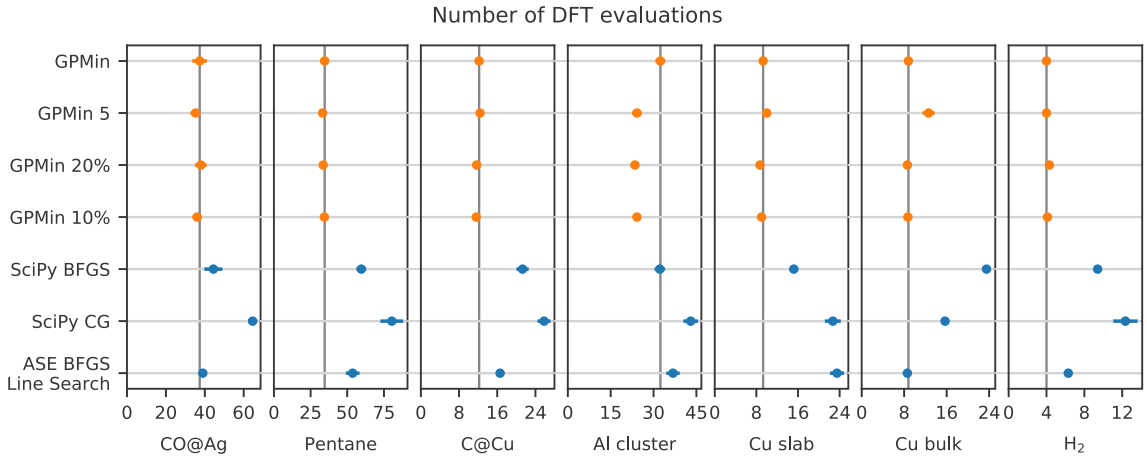


FIG. 5. Number of DFT evaluations required to optimize a given structure. For each structure 10 different initial configurations are generated and optimized. The vertical line represents the average number of steps of GPMin without parameter updates. The error bar represents the error on the average. A different color has been used to highlight the optimizers of the GPMin family.

better than all the updated optimizers. However, in Fig. 3 the scale was chosen to be $l = 0.3$ Å, which is superior for that particular system. This behavior does not appear for any of the test systems including the CO/Ag system, which otherwise could be expected to be somewhat similar.

VIII. DISCUSSION

We ascribe the overall good performance of the GP optimizers to their ability to predict smooth potential energy surfaces covering both harmonic and anharmonic regions of the energy landscape. Since the Gaussian functions applied in the construction of the SPES all have the scale l, the SPES will be harmonic at scales much smaller than this around the minimum configuration. If the initial configuration is in this regime the performance of the optimizer can be expected to be comparable to BFGS, which is optimal for a harmonic PES, and this is what is for example observed for the Cu bulk system. We believe that the relatively worse performance of the SciPy implementation of BFGS can be attributed to an initial guess of the Hessian that is too far from the correct one.

Given the performance on both the training and test sets, GPMin-10% seems to be a good choice. It should be noted that updating the hyperparameters require iteration over the marginal log-likelihood leading to an increased computational cost. However, this is not a problem at least for systems comparable in size to the ones considered here.

The current version of the algorithm still has room for improvement. For example, different strategies for the update of hyperparameters may be introduced. Another, maybe even more interesting, possibility is to use more advanced prior models of the PES than just a constant. The prior model to the PES could for example be obtained from fast lower-quality methods. Somewhat along these lines there have been recent attempts to use previously known semiempirical potentials for preconditioning more traditional gradient-based optimizers [32,33]. This approach might be combined with the GP framework suggested here.

We also note that the choice of the Gaussian kernel, even though encouraged by the characteristics of the resulting potential [22] and its previously reported success for similar problems [13], is to some extent arbitrary. It would be worthwhile to test its performance against other kernel functions, for example the Matérn kernel, which has been reported to achieve better performance in different contexts [19,34,35]. The kernels used in the work here are also limited to considering only one length scale. More flexible kernels allowing for different length scales for different types of bonds would be interesting to explore.

The probabilistic aspect, including the uncertainty as expressed in Eq. (12), is presently used only in the update of the hyperparameters. It could potentially lead to a further reduction of the number of function evaluations [13]. The uncertainty provides a measure of how much a region of configuration space has been explored and can thereby guide the search also in global optimization problems [16,34,36].

Finally, a note on the limitations of the present version of the optimizer. The construction of the SPES involves the inversion of a matrix [Eq. (11)] which is a square matrix, where the number of columns is equal to $n = N_c(3N + 1)$, where N is the number of atoms in the system and N_c the number of previously visited configurations. This is not a problem for moderately sized systems, but for large systems, where the optimization also requires many steps, the matrix inversion can be very computationally time consuming, and the current version of the method will only be efficient if this time is still short compared to the time to perform the DFT calculations. In addition, this can also result in a memory issue for large systems where the relaxation takes many steps. These issues may be addressed by considering only a subset of the data points or other sparsification techniques. Recently, Wang

et al. [37] showed that by using the black-box matrix-matrix multiplication algorithm it is possible to reduce the cost of training from $O(n^3)$ to $O(n^2)$ and then by using distributed memory and 8 GPUs they were able to train a Gaussian process of $n \sim 4 \times 10^4$ (this would correspond to about 100 steps for 150 atoms with no constraints) in 50 seconds. This time is negligible compared to the time for DFT calculations of systems of this size.

The GPMin optimizers are implemented in Python and available in ASE [27].

ACKNOWLEDGMENTS

We appreciate fruitful conversations with P. Bjørn Jørgensen. This work was supported by Grant No. 9455 from VILLUM FONDEN.

APPENDIX

The optimization algorithm can be represented in pseudocode as follows:

Input:
Initial structure: $\mathbf{x}^{(0)} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)$
Hyperparameters: l, σ_n,
Tolerance: $f_{\max}$

$E^{(0)}, \mathbf{f}^{(0)} \leftarrow$ CALCULATOR($\mathbf{x}^{(0)}$)
$E_p \leftarrow E^{(0)}$
while $\max_i |\mathbf{f}_i^{(0)}| > f_{\max}$ **do**
  $X, Y \leftarrow$ UPDATE($\mathbf{x}^{(0)}, E^{(0)}, \mathbf{f}^{(0)}$)
  $E_p \leftarrow \max Y_E$
  $\mathbf{x}^{(1)} \leftarrow$ L-BFGS-B(GP(X, Y), start_from $= \mathbf{x}^{(0)}$)
  $E^{(1)}, \mathbf{f}^{(1)} \leftarrow$CALCULATOR($\mathbf{x}^{(1)}$)
  while $E^{(1)} > E^{(0)}$ **do**
    $X, Y \leftarrow$ UPDATE($\mathbf{x}^{(1)}, E^{(1)}, \mathbf{f}^{(1)}$)
    $E_p \leftarrow \max Y_E$
    $\mathbf{x}^{(1)} \leftarrow$ L-BFGS-B(GP(X, Y), start_from $= \mathbf{x}^{(0)}$)
    $E^{(1)}, \mathbf{f}^{(1)} \leftarrow$CALCULATOR($\mathbf{x}^{(1)}$)
    if $\max_i |\mathbf{f}_i^{(1)}| > f_{\max}$ **then** break
    end if
  end while
  $\mathbf{x}^{(0)}, E^{(0)}, \mathbf{f}^{(0)} \leftarrow \mathbf{x}^{(1)}, E^{(1)}, \mathbf{f}^{(1)}$
end while
Output: $\mathbf{x}^{(0)}, E^{(0)}$

[1] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).
[2] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[3] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, New York, 2007).
[4] E. Jones, T. Oliphant, and P. Peterson, SciPy: Open source scientific tools for Python, http://www.scipy.org.
[5] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, Sci. Adv. **3**, e1701816 (2017).
[6] B. Huang and O. A. von Lilienfeld, arXiv:1707.04146.
[7] A. Glielmo, C. Zeni, and A. De Vita, Phys. Rev. B **97**, 184307 (2018).
[8] J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007).
[9] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, Phys. Rev. Lett. **108**, 058301 (2012).
[10] G. Csányi, T. Albaret, M. C. Payne, and A. De Vita, Phys. Rev. Lett. **93**, 175503 (2004).
[11] A. Khorshidi and A. A. Peterson, Comput. Phys. Commun. **207**, 310 (2016).
[12] T. Yamashita, N. Sato, H. Kino, T. Miyake, K. Tsuda, and T. Oguchi, Phys. Rev. Mater. **2**, 013803 (2018).
[13] O.-P. Koistinen, F. B. Dagbjartsdóttir, V. Ásgeirsson, A. Vehtari, and H. Jónsson, J. Chem. Phys. **147**, 152720 (2017).
[14] T. L. Jacobsen, M. S. Jørgensen, and B. Hammer, Phys. Rev. Lett. **120**, 026102 (2018).
[15] M. Todorović, M. U. Gutmann, J. Corander, and P. Rinke, npj Comput. Mater. **5**, 35 (2019).
[16] M. S. Jørgensen, U. F. Larsen, K. W. Jacobsen, and B. Hammer, J. Phys. Chem. A **122**, 1504 (2018).
[17] E. V. Podryabinkin and A. V. Shapeev, Comput. Mater. Sci. **140**, 171 (2017).
[18] K. Gubaev, E. V. Podryabinkin, G. L. Hart, and A. V. Shapeev, Comput. Mater. Sci. **156**, 148 (2019).
[19] A. Denzel and J. Kästner, J. Chem. Phys. **148**, 094114 (2018).
[20] O.-P. Koistinen, V. Ásgeirsson, A. Vehtari, and H. Jónsson, ChemRxiv (2019), doi:10.26434/chemrxiv.8850440.v1.
[21] J. A. Garrido Torres, P. C. Jennings, M. H. Hansen, J. R. Boes, and T. Bligaard, Phys. Rev. Lett. **122**, 156001 (2019).
[22] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning* (MIT, Cambridge, Massachusetts, 2006).
[23] J. Wu, M. Poloczek, A. G. Wilson, and P. Frazier, in *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017), pp. 5267–5278.
[24] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, SIAM J. Sci. Comput. **16**, 1190 (1995).
[25] K. W. Jacobsen, J. K. Nørskov, and M. J. Puska, Phys. Rev. B **35**, 7423 (1987).
[26] K. W. Jacobsen, P. Stoltze, and J. K. Nørskov, Surf. Sci. **366**, 394 (1996).
[27] Atomic Simulation Environment (ASE), https://wiki.fysik.dtu.dk/ase.
[28] A. H. Larsen, J. J. Mortensen *et al.*, J. Phys.: Condens. Matter **29**, 273002 (2017).
[29] J. Enkovaara, C. Rostgaard, J. J. Mortensen *et al.*, J. Phys.: Condens. Matter **22**, 253202 (2010).
[30] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch, Phys. Rev. Lett. **97**, 170201 (2006).
[31] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevB.100.104103 for a full overview of the average number of DFT calculations needed to relax the structures in the training set with different hyperparameters and strategies for GPMin with updates.

[32] J. O. B. Tempkin, B. Qi, M. G. Saunders, B. Roux, A. R. Dinner, and J. Weare, J. Chem. Phys. **140**, 184114 (2014).
[33] L. Mones, C. Ortner, and G. Csányi, Sci. Rep. **8**, 13991 (2018).
[34] D. J. Lizotte, Practical Bayesian optimization, Ph.D. thesis, University of Alberta, 2008.
[35] G. Schmitz and O. Christiansen, J. Chem. Phys. **148**, 241704 (2018).
[36] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, Proc. IEEE **104**, 148 (2016).
[37] K. A. Wang, G. Pleiss, J. R. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson, arXiv:1903.08114.

Paper III

An artificial intelligence-driven approach for the exploration of potential energy surfaces

José A. Garrido Torres, **Estefania Garijo del Río**, Verena Streibel, Martin H. Hansen, Tej S. Choksi, Ask H. Larsen, Jens J. Mortensen, Alexander Urban, Michal Bajdich, Frank Abild-Pedersen, Karsten W. Jacobsen and Thomas Bligaard

Submitted

An artificial intelligence-driven approach for the exploration of potential energy surfaces

José A. Garrido Torres[1, 2, 3, †], **Estefanía Garijo del Río**[4, †], **Verena Streibel**[1,2], **Martin H. Hansen**[1,2], **Tej S. Choksi**[1,2], **Ask H. Larsen**[4], **Jens J. Mortensen**[4], **Alexander Urban**[3], **Michal Bajdich**[2], **Frank Abild-Pedersen**[2], **Karsten W. Jacobsen**[4], **and Thomas Bligaard**[1, 2, *]

[1]SUNCAT Center for Interface Science and Catalysis, Department of Chemical Engineering, Stanford University, Stanford, CA 94305, USA
[2]SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA
[3]Columbia Electrochemical Energy Center, Department of Chemical Engineering, Columbia University, New York, NY 10027, USA
[4]Department of Physics, Technical University of Denmark, Fysikvej, 2800, Kgs. Lyngby
[*]corresponding author: Thomas Bligaard (bligaard@slac.stanford.edu)
[†]these authors contributed equally to this work

ABSTRACT

Despite the continuing advances in software and hardware, simulating high accuracy atomic-scale phenomena is still challenging since they usually require performing expensive first principles calculations. Studying chemical reactions of even small molecules involves initializing optimizations with several atomic configurations to find minima and transition states of the energy function, and each optimization usually requires tens to hundreds first-principles function evaluations. Typically, optimization algorithms perform their task without communicating to each other and end up inefficiently exploring the same configuration space. Here, we demonstrate that the use of artificial intelligence can serve to advance the efficiency of screening atomic-related hypersurfaces by coupling optimization algorithms through an active learning framework. Our workflow combines different machine learning algorithms that can collectively operate in an active-learning manner to reduce computational effort of atomic-related optimizations. Using these principles we are able to reduce the number of costly function queries by more than one order of magnitude with respect to the workhorse methods for optimizing atomic structures.

Introduction

In the field of computational chemistry, simulations are used to improve success rates and reduce development time and cost of experiments. Exploring the chemical space using *ab initio* simulations is crucial to study atomic-scale mechanisms of chemical reactions[1–4]. One of the major drawbacks of simulating atomic-related phenomena from first-principles is the high computational cost of solving quantum-mechanical equations. Typical atomic-scale simulations ultimately rely on the construction of potential energy surfaces (PES). These surfaces describe the potential energy of the system for a given nuclear configuration (e.g. atom positions, charge or magnetic ordering). However, predicting the morphology of PES is non-trivial, especially when dealing with multi-element systems involving a large number of atoms, due to the increasing combinatorial explosion of particle interactions. A computationally efficient manner of navigating these hypersurfaces is to perform geometry optimizations of atomic structures, searching for particular atomic arrangements representing equilibrium geometries and transition states. These stationary points on potential energy surfaces are especially important since they dictate the stability of atomic structures and activation barriers of reactions involved in chemical and physical processes.

A large variety of optimization algorithms are available to determine stationary points of potential energy surfaces, searching for energy minima (such as gradient descent[5], *quasi*-Newton[6] and molecular dynamics-based[7] methods), transition states (such as the Dimer[8] and the Lanczos[9] methods) or minimum energy pathways (such as the Nudged Elastic Band[10,11] and string methods[12,13]). One of the shortfalls of the aforementioned methods is that their decisions are strictly limited by the information of the observations acquired by evaluating expensive functions, for instance energies and forces. No statistical theory is used to assess the probability of success in their future decisions or guide their choices. Information obtained about the potential energy surface is either discarded or reused only to a very limited extent.

Machine learning has partially addressed the expensive computational cost of optimizing energy landscapes of atomic

structures by introducing surrogate models[14–22]. Statistical models of the potential energy surfaces can be built upon a few first-principles observations, and then, the structures can be optimized on the cheaper predicted potential. Neural networks[14,23] and Gaussian processes[16,17,20,21] regression have been used to build predictive models of potential energy surfaces using first-principles observations. These methods have served to construct supervised learning algorithms that can be exploited to accelerate minima searches locally[20] and globally[18] along with minimum energy pathways[16,21] and transition states[17].

Despite the number of methods available, one important aspect, which is key to increase the computational efficiency, has been overlooked: allowing communication between different algorithms can be used to substantially reduce the number of expensive function evaluations. Several initial geometries are typically optimized in order to increase the probability of finding the global minimum of complex potential energy surfaces. A similar scenario is found when searching for reaction barriers, i.e. different geometries must be optimized to obtain minimum energy pathways. When optimizing complex hypersurfaces a large number of optimization are required and many of them end up exploring locations of the PES that have already been visited during previous optimizations. Revisiting regions of the PES ends up being extremely inefficient since expensive calculations are performed redundantly.

Our goal is to accelerate the search of multiple atomic structure optimizations by implementing regression routines that inherently hold information of previously explored potential energy regions. We propose a framework that combines ideas from previous gradient-based[8,11] and machine learning algorithms[16,17,20,21] to create surrogate models of the PES. The framework relies on using probabilistic estimates from machine learning models as a surrogate of the PES to efficiently guide classical optimization algorithms towards finding minima, transition states and minimum energy pathways with strictly the same accuracy as the traditional methods. The communication between the different algorithms is achieved via a shared training set, such that the machine learning estimates can provide a holistic view of the explored configuration space. This comprehensive view allows the algorithms to concentrate their efforts on performing expensive calculations in the regions of interest (e.g. local minima and saddle points). In this work, we include examples of optimizations on the effective medium theory (EMT) and density functional theory (DFT) to highlight the importance of moving towards a new generation of AI-driven algorithms. We show that our AI-driven (AID) framework can accelerate the search of minima and transition states by an order of magnitude with respect to the workhorse methods for optimizing atomic structures.

Results

Active learning architecture.

Our framework interconnects different algorithms and allows us to create machine learning surrogate models. These surrogates ultimately dictate the geometries to compute next, following an active learning approach (Fig. 1). The workflow starts by performing first-principles calculations of the initial guesses if no previous data is available. The observations are then gathered in a training set and stored in the machine learning calculator object. This calculator object is the core component of our framework since it is in charge of building the machine learning model, training with first-principles observations and then offering predictions of the potential energy surface. In this framework, we train the machine learning model for atomic positions, energies and forces (see Methods for details of the models). Once the model calculator is trained, we can make queries to it, asking for the predicted energies and forces of any unseen configuration along with the corresponding uncertainty estimates. Training the model and querying for predictions of a large number of configurations is substantially cheaper than evaluating the energy and forces of a single structure using first principles. Therefore, we implemented gradient-based optimization methods in our framework to optimize structures in the predicted potential (these serve as surrogates for our optimizations). Within these methods, we included surrogate models based on quasi-Newton theory for energy minimization and transition state search along with a surrogate model that uses Nudged Elastic Band (NEB) theory to find minimum energy pathways (details about the implementation of these algorithms are included in Methods). The outcome of the optimizations performed in the predicted machine learning potential serve as the inputs for the next generation of first-principles calculations (observations). Finally, these observations are fed into the machine learning calculator, improving the predictions of the model and closing the feedback loop in the active learning cycle. This routine is stopped when the suggested structures are evaluated by first-principles calculations and they satisfy the optimization convergence criteria (e.g. forces of the atoms below a certain user-defined threshold). Therefore, we can certify that the accuracy of the results is not compromised when using our workflow.

Visualizing the workflow of the active learning framework through a toy model.

We built a toy model to explain the different steps composing our active learning framework on a test case that involves the optimization of atomic structures. Here we focus on applying our method to probe a simple reaction network using the Effective Medium Theory (EMT) to describe interatomic interactions. In particular, we study the diffusion of an Au adatom on a highly disordered Al/Au surface, such as the one represented in Fig. 2. This toy model contains a small number of degrees of freedom to allow for a simple visual representation of the targeted potential, but it is challenging enough to require optimization methods to search for the stationary points (e.g. minima and transition states) of the reaction network. The surface that we used for

this study is illustrated in Fig. 2b. The relationship between the position of the Au adatom adsorbed on this surface and the energy of the system can be depicted by the contour plot in Fig. 2c. This potential energy surface is constructed by calculating the energies resulting from optimizing the distance between the surface atoms and the Au adatom at different constrained (x, y) positions and serves to provide a general overview of the stability of the Au adatom upon adsorption at different positions of the surface. Note that scanning hypersurfaces with many degrees of freedom is not a common practice since it cannot be done with a reasonable amount of computer power, so the intention of scanning this PES with a grid is only to produce a visual representation of the target potential (Fig. 2c).

The goal of this test case is to obtain the energy profile of the reaction network connecting three non-degenerate initial configurations (S1, S2, S3), which correspond to three different adsorption sites of the Au adatom on the Al/Au surface (marked by the white dashed circles in Fig. 2b-c). The challenge is to reduce the number of queries to the analytical potential to probe reaction networks connecting these configurations.

We start the active learning workflow by optimizing three initial guesses around the S1−S3 positions using our machine learning surrogate model for structure energy minimization (details of the models are included in Methods). The predicted potential after performing the optimizations towards finding the S1 (blue circles), S2 (green squares) and S3 (red diamonds) structures along with their corresponding energy profiles are shown in Fig. 3a and Fig. 3b, respectively. After that, the optimized structures are used to initialize NEB calculations to find minimum energy pathways connecting the S1, S2 and S3 configurations. The resulting predicted potential after converging the NEB calculations are shown in Fig. 3c. The potential energy profiles for each transition are represented in Fig. 3d. During the optimization of the NEBs, we unraveled the location of three minima (I1−I3) and six transitions-states (TS1−TS6). Due to the limitations of the climbing-image NEB method, we can only ensure that the saddle point with higher energy is accurately converged for each NEB. However, the other structures composing the set of NEB images (including other minima and transition states) might not be converged with the same accuracy. Therefore, a last set of optimizations is required to refine the location of the minima and saddle points involved in the aforementioned reaction pathways. We used our surrogate machine learning algorithms based on quasi-Newton algorithms to perform optimizations of the minima and saddle points to converge the structures to the same level of accuracy (details included in Methods). The predicted potential after training with the observations collected by the different surrogate machine learning algorithms is shown in Fig. 3e.

The surrogate models in our workflow update the model when a new observation to the "*exact*" potential energy function is made and, hence, the overall description of the predicted potential energy becomes more accurate at each step. The improvement of the predictions with increasing data becomes visually evident by the evolution of the predicted potential energy after each optimization (see Figs. 3a,c,e). The improvement of the model with each observation allows to explore the regions of interest (near the stationary-points) in a very efficient manner. For instance, the collected energy and forces during the optimizations of the S1, S2 and S3 structures serve to accelerate the convergence of the NEB calculations by including information of the regions surrounding the initial and final end-points of the elastic band. Another situation, in which this workflow accelerates the convergence of the optimizations, is found when different NEB calculations end up connecting reactants and products through similar potential energy regions. In that case, the machine learning algorithm decides (using the uncertainty estimates of the model) that no further evaluations are required at these locations. For example, the data collected during the optimization of the S1→S2 NEB speeds up the convergence of the S1→S3 optimization, since both reaction pathways contain regions with similar structures, e.g. the transition state TS1 and intermediate I1 (see Fig. 3e). In this case, the second NEB optimization is accelerated by avoiding the evaluation of structures in previously explored regions of the PES. The last example of acceleration is observed when combining different algorithms that work in synergy to converge minima and saddle points. For instance, refining the geometries of the new minima (I1−I3) and transition states (TS1−TS6) using the energy minimization and transition state search machine learning algorithms requires only a few extra calculations since the model already contains data in the surroundings of these stationary points, which was collected during previous NEB calculations.

The exploration of multiple pathways is crucial to estimate reaction barriers and thermodynamic relations of chemical and physical phenomena. In our example, we observe that the Au adatom can migrate from an initial state S1 to a final state S3 through two different pathways (the energy profiles for these transitions are represented in Fig. 3f). The first route contains two intermediates (I1 and S2) and three transition states connecting these minima (TS1, TS2 and TS6) whilst the second one involves a transition through three intermediates (I1, I2 and I3) and four transition states (TS1, TS3, TS4 and TS5). The limiting reaction step for the diffusion of the Au along the first route is limited by TS6 and is energetically less favourable than the one for the second Au diffusion pathway (limited by TS3). Noticeably, it was essential to explore a longer route (containing a larger number of intermediate states) to find an energetically more favourable process for the migration of the Au adatom (see Fig. 3f).

In order to quantify the acceleration of our workflow, we perform energy minimization, NEB and transition state search calculations using methods that do not rely on machine learning surrogate models and are the workhorse of atomic-related optimizations. In particular we used the BFGS[24], FIRE[7], and MDMin[25] algorithms implemented in the Atomic Simulation Environment (ASE)[25]. To make a fair comparison between the methods, we tried to keep the parameters of these algorithms

consistent with the ones implemented in our workflow and we use the same initial configurations for all methods. In Fig. 3g, we compare the number of function evaluations required for each method to complete the set of optimizations required to converge the minima and saddle points connecting the S1, S2 and S3 states. In this example, we show that our artificial intelligence-driven workflow (AID) reduces the number of function evaluations of each individual optimization method, but even more significantly, the combination of active learning surrogate models serves to accelerate the overall set of optimizations by more than an order of magnitude with respect to the methods that are not assisted by machine learning models (see Fig. 3g).

Active learning to explore reaction networks of density functional theory potential energy surfaces.

We use our active learning surrogate models to probe a reaction network for the dissociation of a CH* molecule on a stepped copper (211) surface. The simulation model is illustrated in Fig. 4a. We start our workflow by enumerating the symmetrically non-equivalent adsorption sites for CH* along with the configurations describing its dissociative adsorption (C···H). A total of 159 starting configurations were used to initialize the DFT optimizations (details of the enumeration and simulations are included in Methods). The optimizations end up converging into only 5 geometrically different CH* configurations (bonded to the surface through the C atom) and 21 dissociated configurations (C and H adsorbed on different adsorption sites). We label these configurations as A−E (for adsorbed CH*) and 0−20 (dissociated C···H), "A" and "0" being the most stable configurations for each group. The atoms' positions in the unit cell for these optimized structures are marked by the black (carbon positions) and white circles (hydrogen positions) in Fig. 4b.

We compare the performance of the aforementioned optimizations when using our surrogate machine learning approach and when we carry out the same optimizations using a *quasi*-Newton (BFGS) energy minimization method. The *hexbin* plot in Fig. 4b is built by counting and binning the carbon (blue heatmaps) and hydrogen (red heatmaps) positions where DFT force evaluations were performed during the optimizations. The darker the color the larger the number of DFT calculations that were performed in a given region. The three algorithms tested in this work (BFGS, FIRE and MDMin) required over 10,000 DFT function queries to perform the energy minimizations. For a large number of bins the *quasi*-Newton method already requires an order of magnitude more DFT forces queries than our active learning algorithm (note the logarithmic color scale in Fig. 4b).

The stability of the different molecular CH* (adsorbed) and C···H (dissociated) optimized configurations is shown in Fig. 4c. Using the geometries of the optimized structures we built a reaction network connecting the CH* with the C···H configurations (see Fig. 4c). Building this network requires performing 105 different NEB calculations. Again, we compare our active learning approach with the other methods using the same optimization parameters and initial guesses. We stopped the calculations after exceeding 12,000 DFT evaluations to avoid unnecessarily expending computer resources. The BFGS, FIRE and MDMin algorithms were not able to converge the first reaction pathway of this network (A→0) before exceeding this threshold. Remarkably, using the active learning surrogates we were able to exhaustively probe the reaction network with 8,758 DFT calculations (including the DFT calculations performed during the 159 energy minimizations and 105 NEB optimizations).

The activation energy barriers for the CH dissociation obtained after optimizing the NEBs are shown in Fig. 4d. The colors of each box represent the activation energy values for the dissociation of CH using different combinations of CH* and C···H configurations. The boxes with dark red color imply that the energy required to overcome the activation barrier for dissociating CH* into C···H is large. In contrast, dark blue colors represent low activation barriers. Interestingly, the dissociation reaction through the most stable adsorption configurations, i.e. configurations "A" and "0", do not reveal the lowest energy reaction pathway. Instead, we found that the NEB calculation that is initialized using the "D" and "3" configurations offers a lower energy pathway. The structures of the initial, final and transition states for the A→0 and D→3 paths are included in Fig. 4e. The energy barriers for these transitions differ by more than 0.4 eV and consider the dissociation of C−H through geometrically different pathways, i.e. dissociating the molecule on the step−edge (A→0) or on the Cu terrace (D→3).

Note that in Fig. 4d we include only the energies required to dissociate CH* into C and H, however, the optimized NEBs can go through pathways that include other processes, such as diffusion of molecular CH* or the diffusion of the C and H atoms after dissociation. For instance, the NEB path from "D" to "3" starts with the diffusion of CH* from "D" to the most stable "A" configuration without breaking the CH* bond. After that, the path follows the dissociation process from "A" to "3" represented in Fig. 4e (before dissociation snapshot). The energy for D→3 represented in Fig. 4d contains only the barrier for CH* dissociation. However, we have to include in our calculation set different starting positions for C and H in order to consider the possibility that the reaction undergoes through a concerted mechanism. For instance, the reaction involving a simultaneous migration of C and H atoms to different adsorption sites when breaking the C−H bond. Here, we note that it is essential to consider multiple candidates for obtaining meaningful minimum energy pathways. Again, our framework takes advantage of storing and reusing the information of previous calculations to significantly accelerate the convergence rate of these optimizations.

Conclusions

To conclude, we have presented an AI workflow that combines multiple surrogate machine learning models to reduce the number of first principles calculations required to explore potential energy surfaces (PES). This framework uses an active learning approach to improve the predictive capabilities of the surrogate models. The first-principles calculations data is stored at each optimization step so the surrogate models can simultaneously access the training sets for constructing the machine learning models. The surrogate models become more accurate with increasing data size and the model utilizes the predictive estimates to prioritize the atomic structures that need to be calculated using first-principles. The fact that the different ML surrogates can simultaneously access the stored data helps to avoid performing calculations of atomic structures in regions of the PES that have been already explored.

Selecting relevant reaction pathways to explore becomes a real challenge when exploring high−dimensional PESs, since guessing the energetic profile and geometries involved in reaction pathways is non-trivial. Computationally expensive optimizations are typically required to unravel reaction mechanisms when using first-principles, which results in a necessity to reduce the number of pathways explored. However, initializing multiple optimizations from different initial guesses is crucial to maximize the probability of sampling parts of the PES that might be relevant to understand the reaction mechanism. Unfortunately, the optimization of many initial guesses end up exploring multiple common structures. For instance, in this work, the dissociation mechanism of CH* on Cu($2\times1\times1$) was studied using 159 initial guesses and these optimizations ended up converging into only 26 geometrically non-equivalent structures. We also found that multiple converged NEBs contain regions of the minimum energy pathway (MEP) that had been previously calculated, such as the migration of CH* from one adsorption site to another before dissociation.

Here, we stress the need for implementing data driven approaches, for an optimal exploration of PESs since there is a high probability of performing redundant calculations when optimizing high−dimensional hypersurfaces. The use of the active learning algorithms implemented in our workflow allows us to reduce the number of first-principles function evaluations by more than an order of magnitude with respect to the workhorse methods for optimizing atomic structures. In particular, we have probed a reaction network for the dissociate adsorption of C−H on Cu(211), which required optimization of 159 structures and the performance of 105 NEB calculations. The active learning framework requires less than 9,000 DFT force evaluations to perform these optimizations. Remarkably, none of the other algorithms were able to optimize the 159 initial atomic structures with the same amount of DFT calculations, and typically, the NEB calculations are orders of magnitude more expensive than the geometry optimization of atomic structures.

Currently, our framework relies on the construction of machine learning models using Gaussian Processes Regression (GPR). GPR is a non-parametric method and provides accurate predictions of the local environment of the training data and intrinsically offers uncertainty estimates of the model, which has served us to build efficient surrogate models for local optimizations. However, the field of artificial intelligence is advancing very rapidly, and other approaches may quick become relevant. By design, the machine−learning model calculator in our framework is detached from the algorithms that are in charge of building the surrogates. Therefore, varying the fingerprint or updating the framework with new models can be done without affecting the structure of the framework. For instance, the models could potentially include features with molecular symmetry information or extra atomic properties (such as magnetic ordering). Furthermore, our modular framework eases the implementation of other types of optimization algorithms that have not been introduced in this work, for instance, methods for global optimization. We believe that our work provides a generalizable method to facilitate machine learning model building to advance computational experiments beyond user's chemical intuition to increase autonomy and efficiency for exploration of chemical space.

Methods

Machine learning models and parameters

Our algorithms use Gaussian Process Regression (GPR) to build predictive potential energy surface models using energy and force observations of the targeted potential following the same implementation than in Ref.[20]. We use the squared exponential kernel to model the correlation between the energy of the different structures:

$$k = \sigma_f \, e^{-||\mathbf{x}-\mathbf{x'}||^2/2\ell^2}, \tag{1}$$

where $||\mathbf{x}-\mathbf{x'}||^2$ defines the squared Euclidean distance between two configurations ($\mathbf{x}$) and a constant function to model the prior mean of the process. The length-scale of the kernel (ℓ) is kept fixed to 0.4 Å. The prior variance (σ_f) and the constant of the prior mean are updated each time a new observation is made to maximize the marginal likelihood of the process. We include first derivative observations in the covariance following the same mathematical procedure as in Ref.[26].

The core of this framework is the active learning calculator, which serves to interconnect the algorithms building machine learning models using a common training set. In our framework, the surrogate models are trained each time a data point is

added to the training list improving their predictive capabilities. We used the BFGS, NEB and Dimer methods (as implemented in ASE[25]) to guide different optimizations towards finding local minima (AIDMin), minimum energy pathways (AIDNEB) and performing transition state searches (AIDTSS).

The computational cost of training and predicting a GPR model including first-derivative observations scales $\mathcal{O}(n^3d^3)$ with increasing training data size (n) and the dimensionality of the manifold (d), i.e. the number of relevant components of the forces. Our framework stores the information of all the calculated structures, however, to reduce the computational overhead, we only train the models with the information that is close to the initial guesses for each optimization step. For instance, the AIDMin algorithm takes only the 5 nearest training points to build a predictive model and the initial guess is iteratively optimized in the predicted potential using a *quasi*-Newton method. We add the 5 nearest training points in each cycle, until the surrogate positions do not change more than 0.001 Å. Then, we consider that the surrogate cycle has converged and the geometry of the surrogate is calculated using first-principles. The AIDNEB follows a similar approach than the algorithms in Ref.[21], and utilizes NEB theory as a surrogate model. In this case, the AIDNEB algorithm always considers the 25 nearest structures to each image along the NEB for building the predicted potentials. The NEB is optimized in the predicted potential and the image with the maximum uncertainty is evaluated using first-principles, until the uncertainty of all the images lie below 0.025 eV. After that, a climbing-image is optimized and evaluated using first-principles. If the forces of the relaxed atoms are below the convergence criteria (e.g. 0.05 eV/Å) the calculation is stopped, otherwise, the active learning cycle continues until satisfying the NEB uncertainty and forces criteria.

Density functional theory calculations.

DFT calculations were performed using the Vienna *Ab initio* Simulation Package (VASP)[27,28] and the revised Perdew−Burke−Ernzerhof functional (RPBE)[29], with a mesh of (4×4×1) $k-$points, a 400 eV energy cutoff and the default pseudopotentials included in the VASP version 5.4. The convergence criterion for the electronic self-consistent cycle was fixed at 10^{-5} eV whilst the structures were optimized until the forces of the relaxed atoms were below 0.03 eV/Å.

The Cu(211)−(3×3) periodic slabs are composed of 4 layers with the atoms fixed to their bulk positions. We include 20 Åof vacuum to separate the periodic images along the z coordinate. The enumeration of the different initial structures was performed using graph theory as implemented in the Catalysis Kit (CatKit)[30] and the NEB initial guesses were generated using the Image Dependent Pair Potential (IDPP) interpolation[31]. The number of moving images along the path is chosen by measuring the distance between the geometries of the initial and final end−points, including a moving image every 0.5 Å along the path. The NEB spring constants are calculated as:

$$k = \frac{2\sqrt{N-1}}{d_{A-B}}, \tag{2}$$

where N is the number of NEB images and d_{A-B} is the Euclidean distance between the initial and final structures.

Code implementation and availability

The surrogate machine learning models along with the framework are implemented in the Atomic Simulation Environment (ASE). The code containing the active learning routines are hosted in the "AID_framework" branch at the repository https://gitlab.com/ase.

References

1. Nichols, J., Taylor, H., Schmidt, P. & Simons, J. Walking on potential energy surfaces. *The J. chemical physics* **92**, 340–346 (1990).
2. Ball, K. D. *et al.* From topographies to dynamics on multidimensional potential energy surfaces of atomic clusters. *Science* **271**, 963–966 (1996).
3. Chmiela, S. *et al.* Machine learning of accurate energy-conserving molecular force fields. *Sci. advances* **3**, e1603015 (2017).
4. Gromski, P. S., Henson, A. B., Granda, J. M. & Cronin, L. How to explore chemical space using algorithms and automation. *Nat. Rev. Chem.* **3**, 119–128 (2019).
5. Chatzieleftheriou, S., Adendorff, M. R. & Lagaros, N. D. Generalized potential energy finite elements for modeling molecular nanostructures. *J. chemical information modeling* **56**, 1963–1978 (2016).
6. Burger, S. K. & Ayers, P. W. Quasi-newton parallel geometry optimization methods. *The J. chemical physics* **133**, 034116 (2010).

7. Bitzek, E., Koskinen, P., Gähler, F., Moseler, M. & Gumbsch, P. Structural relaxation made simple. *Phys. review letters* **97**, 170201 (2006).

8. Henkelman, G. & Jónsson, H. A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives. *The J. chemical physics* **111**, 7010–7022 (1999).

9. Olsen, R., Kroes, G., Henkelman, G., Arnaldsson, A. & Jónsson, H. Comparison of methods for finding saddle points without knowledge of the final states. *The J. chemical physics* **121**, 9776–9792 (2004).

10. Pratt, L. R. A statistical method for identifying transition states in high dimensional problems. *The J. chemical physics* **85**, 5045–5048 (1986).

11. Henkelman, G., Uberuaga, B. P. & Jónsson, H. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The J. chemical physics* **113**, 9901–9904 (2000).

12. Peters, B., Heyden, A., Bell, A. T. & Chakraborty, A. A growing string method for determining transition states: Comparison to the nudged elastic band and string methods. *The J. chemical physics* **120**, 7877–7886 (2004).

13. Weinan, E., Ren, W. & Vanden-Eijnden, E. Simplified and improved string method for computing the minimum energy paths in barrier-crossing events. *J. Chem. Phys.* **126**, 164103 (2007).

14. Khorshidi, A. & Peterson, A. A. Amp: A modular approach to machine learning in atomistic simulations. *Comput. Phys. Commun.* **207**, 310–324 (2016).

15. Peterson, A. A. Acceleration of saddle-point searches with machine learning. *The J. chemical physics* **145**, 074106 (2016).

16. Koistinen, O.-P., Dagbjartsdóttir, F. B., Ásgeirsson, V., Vehtari, A. & Jónsson, H. Nudged elastic band calculations accelerated with gaussian process regression. *The J. Chem. Phys.* **147**, 152720 (2017).

17. Koistinen, O.-P., Ásgeirsson, V., Vehtari, A. & Jónsson, H. Minimum mode saddle point searches using gaussian process regression with inverse-distance covariance function. *J. Chem. Theory Comput.* **16**, 499–509 (2019).

18. Jørgensen, M. S., Larsen, U. F., Jacobsen, K. W. & Hammer, B. Exploration versus exploitation in global atomistic structure optimization. *The J. Phys. Chem. A* **122**, 1504–1509 (2018).

19. Chen, X., Jørgensen, M. S., Li, J. & Hammer, B. Atomic energies from a convolutional neural network. *J. chemical theory computation* (2018).

20. del Río, E. G., Mortensen, J. J. & Jacobsen, K. W. Local bayesian optimizer for atomic structures. *Phys. Rev. B* **100**, 104103 (2019).

21. Torres, J. A. G., Jennings, P. C., Hansen, M. H., Boes, J. R. & Bligaard, T. Low-scaling algorithm for nudged elastic band calculations using a surrogate machine learning model. *Phys. review letters* **122**, 156001 (2019).

22. Garijo del Río, E., Kaappa, S., Garrido Torres, J. A., Bligaard, T. & Wedel Jacobsen, K. Machine learning with bond information for local structure optimizations in surface science. *arXiv e-prints* arXiv–2010 (2020).

23. Artrith, N. & Urban, A. An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for tio2. *Comput. Mater. Sci.* **114**, 135–150 (2016).

24. Liu, D. C. & Nocedal, J. On the limited memory bfgs method for large scale optimization. *Math. programming* **45**, 503–528 (1989).

25. Larsen, A. H. *et al.* The atomic simulation environment—a python library for working with atoms. *J. Physics: Condens. Matter* **29**, 273002 (2017).

26. Wu, J., Poloczek, M., Wilson, A. G. & Frazier, P. Bayesian optimization with gradients. In Guyon, I. *et al.* (eds.) *Advances in Neural Information Processing Systems 30*, 5267–5278 (Curran Associates, Inc., 2017).

27. Kresse, G. & Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. review B* **54**, 11169 (1996).

28. Kresse, G. & Joubert, D. From ultrasoft pseudopotentials to the projector augmented-wave method. *Phys. Rev. B* **59**, 1758 (1999).

29. Hammer, B., Hansen, L. B. & Nørskov, J. K. Improved adsorption energetics within density-functional theory using revised perdew-burke-ernzerhof functionals. *Phys. review B* **59**, 7413 (1999).

30. Boes, J. R., Mamun, O., Winther, K. & Bligaard, T. Graph theory approach to high-throughput surface adsorption structure generation. *The J. Phys. Chem. A* **123**, 2281–2285 (2019).

31. Smidstrup, S., Pedersen, A., Stokbro, K. & Jónsson, H. Improved initial guess for minimum energy path calculations. *The J. chemical physics* **140**, 214106 (2014).

Acknowledgements

This work was supported by the U.S. Department of Energy, Chemical Sciences, Geosciences, and Biosciences (CSGB) Division of the Office of Basic Energy Sciences, via Grant DE-AC02-76SF00515 to the SUNCAT Center for Interface Science and Catalysis and the VILLUM FONDEN via Grant No. 9455. The authors would like to acknowledge the use of the computer time allocation for the 2997 allocation at the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. V.S. acknowledges financial support from the Alexander von Humboldt Foundation. We thank Prof. Alan C. Luntz and Dr. Johannes Voss for fruitful scientific discussions.

Author contributions statement

J.A.G.T and E.G. del R. conceived the study, ran the calculations, implemented the methods and contributed to the writing, V.S., M.H.H., T.S.C., M.B. and F.A.P. designed the theoretical studies and contributed to the writing., A.H.L. and J.J.M. and A.U. were involved in the implementation of ML methods and K.W.J. and T.B. supervised research, method development and contributed to the writing. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Figures

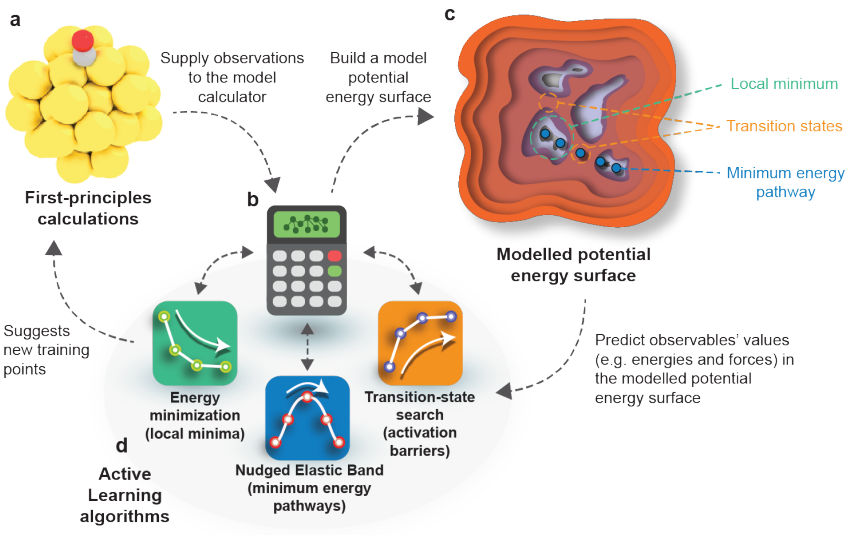


Figure 1. Workflow of the artificial intelligence-driven (AID) framework. The screening process of the potential energy surface (PES) is initialized by (a) performing first-principles calculations of the atomic structure. The observations are added to (b) a machine learning calculator which is in charge of building a (c) predictive model of the PES and (d) optimizing the atomic structures in the predictive potential to suggest the configurations that should be calculated next. The model is updated in each iteration with new observations, improving the predictions of the PES until the user-defined convergence criteria is satisfied.

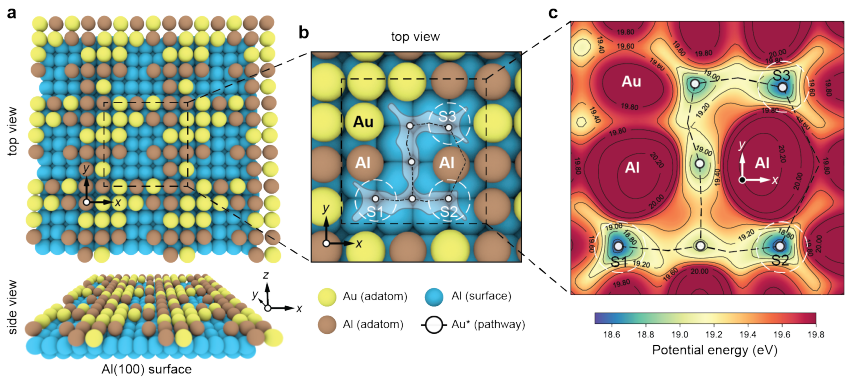


Figure 2. Example model used to describe the behaviour of the machine learning workflow. (a) Top and side views of the sphere model. Al(100) surface atoms in blue, Au and Al adatoms in yellow and brown respectively. (b) Magnification of the spheres model highlighting the area in which the diffusion energy of a Cu adatom is probed. (c) Potential energy surface resulting from varying the x and y coordinates (surface plane) and relaxing the z coordinate (normal to the surface) of the Au adatom. The white circles mark the positions of six connected minima whilst the dashed lines represents the connecting path between them.

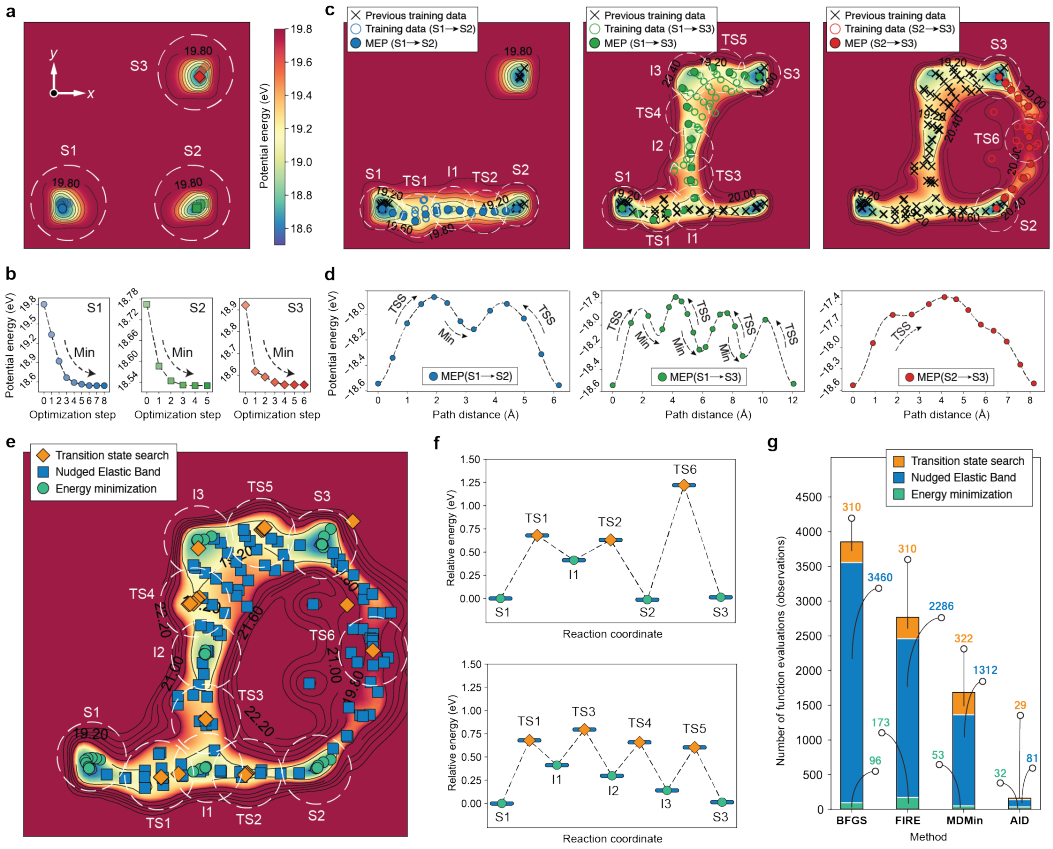


Figure 3. Exploration of the toy model potential energy surface. (a) Machine learning model potential after optimizing the S1, S2 and S3 structures along with (b) the corresponding energy profiles for each optimization. (c) Evolution of the predicted potential when including the observations collected after optimizing the NEBs connecting S1→S2, S1→S3 and S2→S3 consecutively and (d) the minimum energy pathways for each reaction. (e) Predicted potential after refining the intermediate and transition states found during the NEB optimizations. "S" and "I" labels refer the starting and intermediate configurations (minima) whilst the "TS" labels refer the transition states (saddle points). (f) Energy profile for two different pathways connecting the S1 and S3 configurations. (g) Number of function evaluations required for the each method to probe the reaction pathways showed in (f).

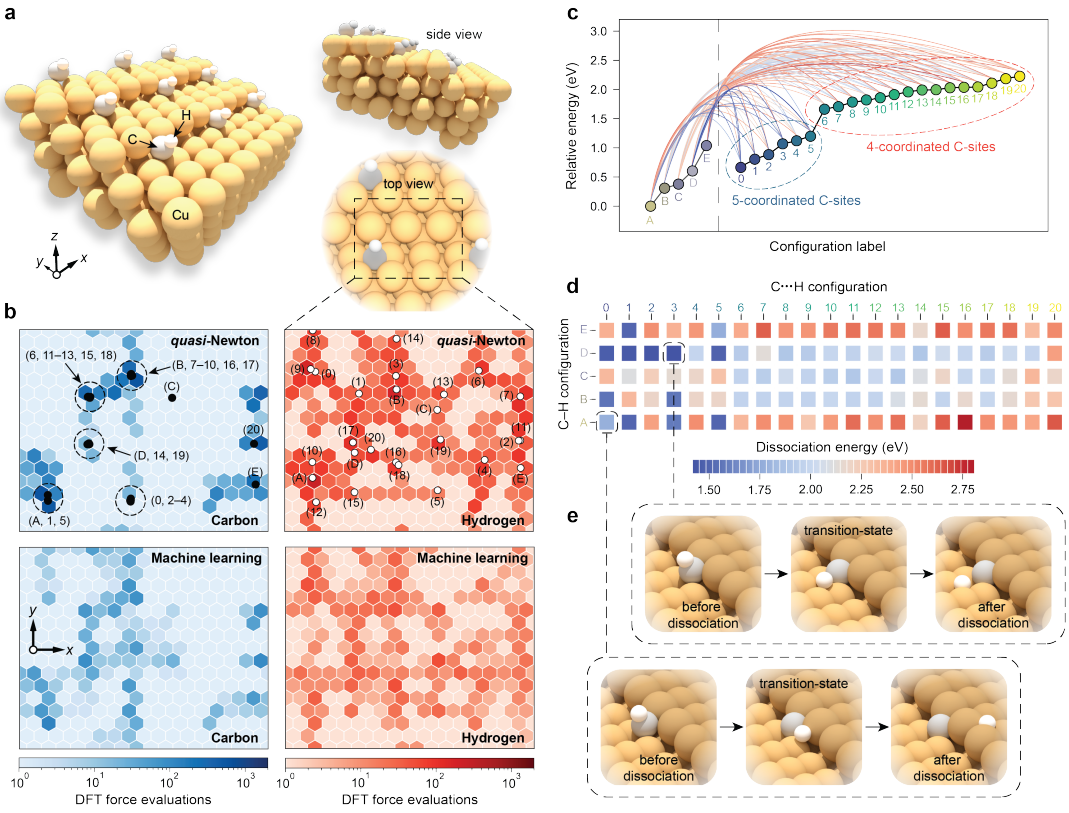


Figure 4. Dissociative adsorption of C−H on a stepped copper (211) surface. (a) Atomic model representing the adsorption of CH* on a Cu(211) surface. Dashed lines mark off the periodic boundaries of the simulation cell. (b) Visualization of the DFT force evaluations (carbon and hydrogen counts in blue and red color scales, respectively) required by the active learning and *quasi*−Newton approaches to optimize a total of 159 symmetrically non-equivalent molecular CH* and dissociated C···H configurations. Optimized positions for the CH* adsorption (A−E) and when dissociated into C and H (0−20) are highlighted by the black (carbon) and white circles (hydrogen), respectively. (c) Energy profile for the aforementioned (A−E and 0−20) configurations. (d) Visualization of the dissociation energies for the reaction network between the CH* and dissociated C···H configurations represented in (b). (e) Geometries of the stationary-points involved in the A→0 (upper panels) and D→E (lower panels) dissociation pathways.

Paper IV

Machine Learning with bond information for local structure optimizations in surface science

Estefanía Garijo del Río, Sami Kaappa, José A. Garrido Torres, Thomas Bligaard, Karsten Wedel Jacobsen

Journal of Chemical Physics, **153**, 234116 - Published 21 December 2020

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

Machine learning with bond information for local structure optimizations in surface science

Cite as: J. Chem. Phys. 153, 234116 (2020); doi: 10.1063/5.0033778
Submitted: 19 October 2020 • Accepted: 26 November 2020 •
Published Online: 21 December 2020

Estefanía Garijo del Río,[1] Sami Kaappa,[1] José A. Garrido Torres,[2,3,4] Thomas Bligaard,[2,5] and Karsten Wedel Jacobsen[1,a)]

AFFILIATIONS

[1]Department of Physics, Technical University of Denmark, Kgs. Lyngby, Denmark
[2]SUNCAT Center for Interface Science and Catalysis, Department of Chemical Engineering, Stanford University, Stanford, California 94305, USA
[3]SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, California 94025, USA
[4]Columbia Electrochemical Energy Center, Department of Chemical Engineering, Columbia University, New York, New York 10027, USA
[5]Department of Energy Conversion and Storage, Technical University of Denmark, Kgs. Lyngby, Denmark

[a)]**Author to whom correspondence should be addressed:** kwj@fysik.dtu.dk

ABSTRACT

Local optimization of adsorption systems inherently involves different scales: within the substrate, within the molecule, and between the molecule and the substrate. In this work, we show how the explicit modeling of different characteristics of the bonds in these systems improves the performance of machine learning methods for optimization. We introduce an anisotropic kernel in the Gaussian process regression framework that guides the search for the local minimum, and we show its overall good performance across different types of atomic systems. The method shows a speed-up of up to a factor of two compared with the fastest standard optimization methods on adsorption systems. Additionally, we show that a limited memory approach is not only beneficial in terms of overall computational resources but can also result in a further reduction of energy and force calculations.

I. INTRODUCTION

One of the most common tasks in computational heterogeneous catalysis is finding local minima in a potential energy surface (PES). Such equilibrium atomic configurations are of great interest since they are often the first step from which more complicated studies of reaction rates are carried out. A number of well-established methods exist for this task,[1–4] which rely on iteratively computing energy-force pairs for a set of atomic configurations. A common and successful choice for the computation of energies and forces is density functional theory (DFT).[5,6] Even though this approach carries a good trade-off between computational cost and accuracy, the structure determination can be very computationally demanding if the optimization method requires many energy-force evaluations.

Recently, the field of efficient local optimization of atomic structures has attracted considerable attention. An interesting approach is that of *preconditioning*:[7–9] in atomic systems where bonds of very different stiffness are present, changes of certain atomic positions produce much more rapid changes in energy than others, and this can result in a slowdown of traditional methods. If the differences in stiffness are very large, the forces may not point in the direction toward the minimum, which is known as a poorly scaled optimization problem.[1] Preconditioning then consists in finding a linear transformation of the problem that will lead to a Hessian with a better condition number, which corrects for the difference in

stiffness in the PES landscape and results in better guidance for the search. For atomic systems, preconditioners based on the adjacency matrix of atoms and their interatomic distances[7] or on the Hessian of semi-empirical potentials[8] have shown a significant reduction in the number of steps necessary to relax atomic structures, as well as to guide transition state searches.[9]

The use of machine learning techniques to build surrogate models of the PES that are then used to guide the search of critical points has also recently attracted increasing attention. Successful examples of methods that have achieved a significant reduction in the number of electronic structure evaluations are abundant for local optimization,[10–15] as well as transition state search[16–21] and global optimization of atomic structures.[22–28]

In particular, Gaussian process regression (GPR)[29] has proved itself a particularly successful technique to build the surrogate PES that guides the critical point search since it has the ability to generalize, given just a few training points.

The computation of forces comes with little additional computational overhead to the energy computation in DFT, and training on both energies and forces has become a well-established technique in the field.[10,12,13,17–19,21] Along these lines, there has been a recent attempt to also incorporate higher derivatives.[20] A recent study by Christensen and von Lilienfeld[30] has confirmed that the inclusion of forces along with energies of the configurations as targets in the training set results in a significant increase in the precision of the surrogate model of the PES of a single atomic system.

A less well-established choice is that of the correlation model between two atomic structures or the kernel in the case of GPR. After the initial success in the use of stationary covariance functions of Cartesian coordinates (squared exponential and Matérn covariance functions),[10,12,17,19,21] there have been some studies attempting to extend these covariance functions in order to further reduce the number of DFT calculations needed to find the critical point. Koistinen *et al.*[18] proposed a non-stationary kernel based on the difference between the inverse of interatomic distances in each configuration for each pair of atoms. Meyer and Hauser[13] instead proposed the use of the squared exponential and Matérn kernels in internal coordinates, instead of Cartesian. Both approaches have led to a further reduction in the number of steps. We note that outside the subfield of gradient-based GPR modeling for PES critical point identification, both internal coordinates[11,28,31] and fingerprints[26,30,32–37] have been used to incorporate knowledge of the PES topology into the covariance function of kernel methods.

In this paper, we introduce a preconditioning scheme of the usual squared exponential kernel in Cartesian coordinates. The resulting expression for the kernel we propose can be reinterpreted in terms of chemical bonds and covalent radii, making it easy for the method to account for differences in the stiffness of each interaction and easy for the user to interpret the results. In this way, the method relies on a model of bond stiffness that can be provided by the user, but we prove that an educated guess can work even better if the method is allowed to self-update and find the bond constants itself. In addition, the structure of the kernel naturally incorporates the translation invariance of the PES.

We have incorporated the new surrogate model into a machine learning optimization method that we have named BondMin, and we have tested its performance in local relaxation problems with DFT. For this method, we have obtained speed-ups of up to a factor of 2 for problems that involve molecules on surfaces as compared to the quasi-Newton method BFGSLineSearch while retaining the good performance of the not preconditioned squared exponential kernel on general atomic systems.

II. METHODS

A. Gaussian process regression

Let $\mathbf{r}_i$ stand for the position vector of the ith atom. For each atomic configuration $\mathbf{x} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{N_{\text{atoms}}})$, we describe the surrogate potential energy surface (sPES) $E(\mathbf{x})$ and the associated force field $\mathbf{f}(\mathbf{x})$ using Gaussian process regression (GPR),[29]

$$(E(\mathbf{x}), -\mathbf{f}(\mathbf{x})) \sim \mathcal{N}\big(\mathbf{m}(\mathbf{x}), K(\mathbf{x}, \mathbf{x}')\big), \tag{1}$$

where $\mathbf{m}(\mathbf{x})$ is the prior mean for each variable and $K(\mathbf{x}, \mathbf{x}')$ is the prior covariance matrix. This matrix can be written in terms of the kernel function $k(\mathbf{x}, \mathbf{x}')$ as[38]

$$K(\mathbf{x}, \mathbf{x}') = \begin{pmatrix} k(\mathbf{x}, \mathbf{x}') & \left(\nabla_{\mathbf{x}'} k(\mathbf{x}, \mathbf{x}')\right)^T \\ \nabla_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}') & \nabla_{\mathbf{x}}\left(\nabla_{\mathbf{x}'} k(\mathbf{x}, \mathbf{x}')\right)^T \end{pmatrix}. \tag{2}$$

The GPR is trained on density functional theory (DFT) energies $\{E^{(i)}\}_{i=1}^N$ and forces $\{\mathbf{f}^{(i)}\}_{i=1}^N$ corresponding to a set of N atomic configurations $\{\mathbf{x}^{(i)}\}_{i=1}^N$. We arrange the inputs into the $3N_{\text{atoms}} \times N$ design matrix X and the targets $\{(E^{(i)}, -\mathbf{f}^{(i)})\}_{i=1}^N$ into the $(3N_{\text{atoms}} + 1) \times N$ matrix Y. By denoting the Gram matrix as $K(X, X)$, which is given by the block matrices $(K(X,X))_{ij} = K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, and defining the matrix $K(\mathbf{x}, X) = (K(\mathbf{x}, \mathbf{x}^{(1)}), \ldots, K(\mathbf{x}, \mathbf{x}^{(N)}))$, the prediction can be written as

$$(E(\mathbf{x}), -\mathbf{f}(\mathbf{x})) = \mathbf{m}(\mathbf{x}) + K(\mathbf{x}, X)\, C_X^{-1}(Y - \mathbf{m}(X)), \tag{3}$$

where $C_X = K(X, X) + \Sigma$ is the regularized Gram matrix and the diagonal matrix Σ is the regularization.

The GPR framework also includes an analytical expression for the marginal likelihood $p(Y|X)$.

$$\log p(Y|X) = -\frac{1}{2}(Y - \mathbf{m}(X))^T C_X^{-1}(Y - \mathbf{m}(X)) - \frac{1}{2}\log|C_X| + \mathfrak{N}, \tag{4}$$

which depends on a number of hyperparameters θ that parametrize the regularized kernel $C_X(X; \theta)$ and the prior $\mathbf{m}(X; \theta)$. The logarithm of the marginal likelihood can be maximized using a gradient-based optimizer to find the most likely hyperparameters, given the inputs and the targets. $\mathfrak{N}$ stands for the normalization factor, which does not depend on X, Y, or θ.

In this work, we introduce a new kernel that uses the difference between the positions between every pair of atoms in the system to define a distance measure $d(\mathbf{x}, \mathbf{x}')$ between configurations,

$$d^2(\mathbf{x},\mathbf{x}') = \frac{1}{N_{\text{atoms}}}\sum_{i,j}^{N_{\text{atoms}}}\frac{\|(\mathbf{r}_i-\mathbf{r}_j)-(\mathbf{r}'_i-\mathbf{r}'_j)\|^2}{\ell^2_{X_iX_j}}, \tag{5}$$

where X_i stands for the atomic symbol of the ith atom and $\mathbf{r}_i$ stands for its position. The scales $\ell_{X_iX_j}$ for each pair of atoms here have length dimensions and have the role of re-scaling the weight of each interatomic distance according to the atomic type.

We note that Eq. (5) can be rewritten into the matrix form as follows:

$$d^2(\mathbf{x},\mathbf{x}') = \frac{1}{N_{\text{atoms}}}(\mathbf{x}-\mathbf{x}')^T G(\mathbf{x}-\mathbf{x}'). \tag{6}$$

It is easy to show that the metric matrix is given by $G = P^T\text{diag}(g, g, g)P$, where P is the permutation matrix mapping $P\mathbf{x} = (r_1^{(x)}, r_2^{(x)}, \ldots, r_N^{(x)}, r_1^{(y)}, \ldots, r_N^{(z)})$ and diag(g, g, g) is the diagonal block matrix composed by three copies of

$$g_{ij} = \begin{cases}\sum_{k\neq i}\ell^{-2}_{X_iX_k} & \text{if } i = j\\ -\ell^{-2}_{X_iX_j} & \text{if } i \neq j.\end{cases} \tag{7}$$

We note that matrix g is the Laplacian matrix of a fully connected undirected graph where the nodes are the atoms in the unit cell and the weights on the edges depend on the chemical species of the atoms that they connect as $1/\ell^2_{X_iX_j}$. A distance measure in the form of a Laplacian matrix has also been used by Packwood *et al.*[7]

This distance measure has then been incorporated into the usual squared exponential kernel, replacing the Euclidean distance between Cartesian coordinates,

$$k(\mathbf{x},\mathbf{x}') = k_0^2\exp(-d^2(\mathbf{x},\mathbf{x}')/2\ell^2), \tag{8}$$

where k_0 and ℓ are hyperparameters: the prefactor of the kernel and the dimensionless global scale.

One could define the vector $\mathbf{b}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ in Eq. (5) as the bond vector defining the distance and the orientation of the bond between atoms i and j. In this picture, the distance between two configurations is then the weighted sum of Euclidean distances between all bonds, with $\ell_{X_iX_j}$ being the weight. However, note that in this conception, every atom is bonded to every other atom in the atomic structure so that the distance measure is not biased toward the initial structure. We note that the inclusion of the interatomic distance of every pair of atoms in the structure is frequently used in fingerprints (such as the Coulomb matrix[35] and other Coulomb-based definitions[34,39] or the bag of bonds[36]) and kernels[18,31] by the machine learning for materials and molecules community. We illustrate this concept in Fig. 1.

Equations (6) and (8) [and noting from Eq. (5) that the matrix G is positive semi-definite for any value of the bond scales $\ell_{X_iX_j}$] reveal that the kernel in terms of bonds is nothing but an anisotropic version of the stationary squared exponential kernel.[29] G has three and only three zero eigenvalues, corresponding to translation along the three axis, making the method translationally invariant. Along the ideas in the work by Packwood *et al.*,[7] we note that G can be factorized as $G = Q^TQ$ and that by defining the fingerprint $\mathbf{u}(\mathbf{x}) = Q\mathbf{x}$, one can regard G as a preconditioner since the energy becomes less anisotropic and hence a better conditioned function in the fingerprint space than in coordinate space.

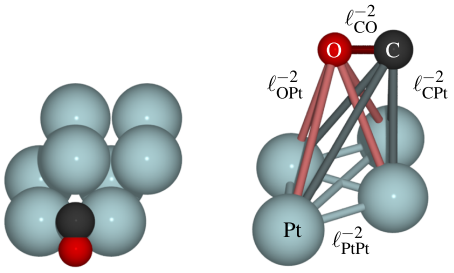


FIG. 1. (a) Top view of the atomic structure of CO on a 2 × 2 × 2 fcc (100) platinum slab. Only the atoms in the unit cell are shown. (b) Representation of the weighted fully connected graph corresponding to the CO molecule and the platinum atoms in the top layer of the structure displayed in (a).

For the particular case of a unary material, there is only one bond scale, ℓ_{XX}. It can be shown that matrix g as defined in Eq. (7) has all eigenvalues equal to N_{atoms}, except for the one associated with translation symmetry (for example, by realizing, g becomes a circulant matrix for unaries). Then, if $\mathbf{x}$ and $\mathbf{x}'$ do not differ in a translation, the distance in Eq. (6) becomes $d^2(\mathbf{x},\mathbf{x}') = \|\mathbf{x}-\mathbf{x}'\|/\ell^2_{XX}$ and the kernel in (8) becomes the isotropic squared exponential kernel $k(\mathbf{x},\mathbf{x}') = k_0^2\exp\|\mathbf{x}-\mathbf{x}'\|^2/2\ell^2\ell^2_{XX}$. Consequently, if the sampling method used to generate the training set does not generate global translations of the atomic structure (i.e., the optimizer does not translate the system), an active learning method using it would behave as its isomorphic squared exponential kernel counterpart with scale $\ell\ell_{XX}$. We then note that in this case, the splitting provides a natural way of systematically providing different scales for different systems, by, for example, making $\ell_{X_iX_j}$ a function of the covalent radii. Additionally, we note other active learning methods using the squared exponential kernel to guide PES exploration[10,12,17,21,40] that could benefit of using kernel (8) with all $\ell_{X_iX_j} = 1$ with no additional retraining since they would obtain similar performance and enforce translation symmetry.

As in previous work,[12,40] we have used the constant function $m(\mathbf{x}) = m_0$ as the prior function. We choose to call the diagonal terms in the matrix Σ for σ corresponding to the regularization in the forces, and we use $\sigma\ell$ as the regularization of the energies.

B. Optimization method

We use the energies and forces from the prediction of the Gaussian process in Eq. (3) to guide the searches for the DFT local minimum of the PES. The optimization method we follow is the one used by GPMin[12] with some variations.

Starting with the initial atomic configuration, the method computes its DFT energy and forces. This information is used to determine the prior constant m_0 and to build a tentative surrogate model of the PES. The method then finds a local minimum of the surrogate PES, computes its DFT energy and forces, and includes that point

in the training set. The surrogate model is then updated with the new information, leading to a new location of the minimum, which is subsequently sampled. The iteration terminates when the DFT maximum force on any of the atoms in the system is smaller than a user-defined tolerance, as is usual for local optimizers in the ASE package.[41] The optimization of the surrogate model always takes the structure with the lowest energy in the training set as the starting point and then uses the L-BFGS-B optimizer[42] as implemented in SciPy[43] to find a neighboring local minimum.

In each iteration, the update of the model may include the update of some of its hyperparameters. In Sec. II A, we have introduced the hyperparameters m_0, k_0, σ, ℓ, and $\ell_{X_iX_j}$ for every pair of atomic species X_i and X_j in the atomic structure, but not all of them may play an independent role in the prediction of the Gaussian process model (3), and thus, not all of them may be updated.[12]

The global scale ℓ and the bond scales $\ell_{X_iX_j}$ are not independent of each other, but ℓ is rather a global dimensionless prefactor to the bond scales. For this reason, the optimizer regards ℓ as a fixed quantity during the optimization of the other hyperparameters.

k_0 and σ only enter Eq. (3) in the form of the quotient σ/k_0, being effectively the same hyperparameter as far as prediction is concerned [note that this does not hold for Eq. (4)]. In addition, we note that the quotient σ/k_0 is the effective regularization of the Gram matrix $K(X, X)$, and even in the absence of numerical noise in the electronic structure model, it needs to be fixed to a small but non-zero value to enable the inversion of the sometimes numerically ill-conditioned Gram matrix, which increases the robustness of the method. In Sec. II C, we determine a value of σ/k_0, which is appropriate for all systems, and the parameter is not updated any further during the optimizations.

Interestingly, the marginal likelihood (4) depends on both k_0 and σ/k_0 in a non-trivial way, making it necessary to optimize k_0 along with the other hyperparameters to obtain sensible results. In fact, the maximization of the marginal log likelihood (4) provides with an analytical expression for the prefactor k_0

$$k_0 = \sqrt{\frac{(Y - \mathbf{m}(X))^T C^{-1}_{X,k_0=1}(Y - \mathbf{m}(X))}{N}} \tag{9}$$

if the quotient σ/k_0 and the scales are kept fixed. A similar expression can be obtained for the prior constant m_0,

$$m_0 = \frac{U^T C_X^{-1} Y}{U^T C_X^{-1} U}, \tag{10}$$

where U is the prior matrix $\mathbf{m}(X)$ with prior constant $m_0 = 1$.

In this work, we present various flavors of the optimization method and we compare their performances. The plain version without updates (termed "BondMin" in the following) only differs with default GPMin in the choice of the kernel. It chooses the prior constant to be the maximum of the energies included in the training set and does not update any other hyperparameter.

We also introduce a method capable of optimizing its own hyperparameters ("BondMin update"). At each step, k_0 and m_0 are updated using expressions (9) and (10). The bond scales, together with k_0, are then further updated by numerically maximizing the marginal log likelihood with optimal m_0. Here, we follow the strategy used by GPMin in the sense that the values of the hyperparameters are found using SciPy's L-BFGS-B with the constraints of not letting any hyperparameter vary more than 10% at each step.

A frequently mentioned limitation of Gaussian process regression is the poor scaling of the computational time and memory requirements with the number of points in the training set.[29] In particular, the use of Cholesky factorization to solve Eqs. (3) and (4) results in $O(n^2)$ scaling for the memory and $O(n^3)$ for the computational time [where the scaling factor n is defined as $n = N(3N_{\text{atoms}} + 1)$] in atomic systems training on energies and forces.[12,17] Here, we have followed the ideas presented by Garrido Torres *et al.*[40] as a way to leverage the computational requirements for systems with large numbers of atoms in the unit cell:

1. We note that for most molecule-on-surface systems (and more generally, in most systems with a large number of atoms), a significant number of atoms have their positions fixed. Thus, there is no need to train on the forces of the constrained atoms, which can also be masked in the kernel, leading to a scaling factor of $n = N(3N_{\text{dyn\_atoms}} + 1)$, where $N_{\text{dyn\_atoms}}$ is the number of dynamical atoms.
2. The problem of predicting the PES for a minimum and its basin with a kernel in the form of Eq. (8) mainly depends on the points close to the minimum. In fact, not including distant points may not dramatically decrease accuracy, while it may increase the robustness of the method.[40,44] This observation allows us to include only the N_0 closest points to the current atomic configuration in Euclidean space in the training set. After the relaxation on the surrogate model has been completed, the method checks if there are points that have not been included in the training that are closer than the N_0 points used, adds them to the training set, and relaxes the new resulting surrogate model.

Altogether, the two strategies give a new scaling with $n = N_0(3N_{\text{dyn\_atoms}} + 1)$. This still yields a quadratic scaling for the memory and a cubic one for the computational time, but it is a big improvement in the scaling of the method. Since N_0 is now a user-defined fixed number, the computational requirements remain constant instead of growing as the optimization progresses. Additionally, the computational cost remains cubic in time, as for the DFT, but on a smaller variable.

We have named the method presented in this paper as BondMin when all the sampled points are included in the training set and LBondMin (light memory BondMin) to the version with the two memory restrictions aforementioned.

C. Computational details

We have described the PES using Density Functional Theory (DFT) as implemented in ASE[41,45] and GPAW.[46] All the calculations presented in this work use RPBE[47] as the exchange-correlation functional, a plane wave basis-set, and an energy cutoff of 600 eV, unless otherwise stated. The Brillouin zone has been sampled with a density of 2.0 k-points per inverse Å in each direction. We have used the projector augmented wave (PAW) formalism,[48] using the setup

with one valence electron for sodium and the default one in GPAW otherwise. We impose the convergence criterium that the maximum change in the magnitude of the force over each atom should fall below 10^{-4} eV/Å to exit the self-consistent field iteration in addition to the default thresholds on the energy, the density, and the Kohn–Sham eigenstates.

Throughout this paper, we consider a structure relaxed when the maximum force on any atom is below 0.01 eV/Å.

D. Datasets

The determination of the parameters to the optimizer and the testing of its performance have been done on different sets of atomic structures. The hyperparameters of the method have been determined by training on and validating systems inspired by the training and the test set used to train GPMin, where the elements of group 11 of the periodic table have been substituted by their counterparts in group 10 for the surfaces with adsorbed CO, since CO does not bind to the original Ag and Au surfaces (see the supplementary material for more details on this matter). The inclusion of clusters, molecules, bulk structures, and surfaces both with and without adsorbates ensures a good overall performance of the method for a large class of systems, preventing overfitting. The method has been tested on two sets of atomic structures containing molecules adsorbed on surfaces.

All the systems considered in this work have been studied in 10 slightly different initial configurations. The 9 rattled copies of each system are generated by adding white noise to the atomic positions of the initial one. The value of the standard deviation of the white noise is specified in the description of each dataset.

1. Hyperparameter training set

The hyperparameter training set consists of two different atomic systems: a randomly generated sodium cluster and a CO molecule on a fcc (100) platinum slab. The two original systems have been perturbed with a noise following a Gaussian distribution with a standard deviation of 0.1 Å.

2. ASE/GPAW test set

The ASE/GPAW test set is the same as the test set presented in Ref. 12 and that available in the GPAW webpage,[49] with the exception that silver has been substituted by palladium in the CO on a surface test. The set consists of two molecules, hydrogen molecule and pentane molecule; the bulk structure of copper fcc in a 2 × 2 × 2 supercell, shaken; a two-layer distorted copper fcc (111) slab; a 13 atom aluminum cluster; and two adsorbates, a carbon atom on a 2 × 2 × 2 fcc (100) copper slab and the aforementioned CO molecule on a 2 × 2 × 2 fcc (111) palladium slab.

3. MS5: Small molecules on surfaces dataset

This dataset is made up of five adsorbates with up to 5 atoms at three different adsorption locations. It consists of two molecules, water and nitrogen dioxide on palladium fcc (100) surfaces and hydroxyl, hydroxymethyl, and methyl radicals on copper fcc (100) surfaces. We have used 2 × 2 × 2 slabs to represent the surfaces and constrained the movement of the atoms in the bottom layer. For each system, we have studied three initial highly symmetric bonding sites of the molecule or radical, those termed "on top," "hollow," and "bridge," as implemented in ASE. The G2 set has been used to obtain the initial structures of the molecules.[50] Each of the original systems has been perturbed with a noise following a Gaussian distribution with a standard deviation of 0.07 Å.

4. C3–4S: 3 and 4 carbon organic molecules on surfaces dataset

The C3–4S set contains two different molecules: acrylic acid[51] (CH_2=CHCOOH) molecule on a fcc (111) 4 × 3 palladium surface in the "on top" position and butanethiolate[52] radical ($C_4H_9S^*$) on a fcc (111) 3 × 3 gold surface in the "hollow" position. Both surfaces are modeled by a 3 layer slab, with the atoms in the bottom layer being kept fixed during the relaxation. Thus, the acrylic acid system has 33 dynamical atoms (45 in total) and the butanethiolate one has 32 dynamical atoms (41 in total). We have solved the electronic structure problem with increased convergence for these two systems: in addition to raising the plane wave energy cutoff to 800 eV, we have added the additional threshold for the termination to the self-consistent field iteration such that the change in the energy in the last 3 iterations should be less than 10^{-6} eV per valence electron.

Each of the original systems has been perturbed with a noise following a Gaussian distribution with a standard deviation of 0.07 Å.

E. Selection of the hyperparameters

In this section, we present the values of those hyperparameters that should remain fixed during the optimization as well as the initial values of the remaining ones. The selection of these values has been done in a two step process: First, we investigate the performance of the relaxation method on the hyperparameter training set with different sets of hyperparameters. In a second step, we have chosen the values of the hyperparameters whose performance is more consistent across the more diverse ASE/GPAW test set among those that performed the best on the hyperparameter training set. Thus, the ASE/GPAW test set acts as a validation set here, preventing overfitting and ensuring increased robustness of the method.

For the method with fixed hyperparameters, we have modeled the bond scales as the average of the covalent radius r^c as tabulated by Cordero *et al.*[53] of the species,

$$\ell_{X_iX_j} = \frac{r^c_{X_i} + r^c_{X_j}}{2}, \tag{11}$$

but the method allows for a user-defined model. In particular, we have further tested the product of covalent radii $\ell_{X_iX_j} = \sqrt{r^c_{X_i} r^c_{X_j}}$, which, under the appropriate choice of the rest of the hyperparameters, did not produce a qualitative improvement when it was tested on the validation set.

The values of the other hyperparameters σ/k_0 and ℓ have been chosen such that they minimize the average number of DFT calculations necessary to relax the structures in the training set. The results are shown in Fig. 2.

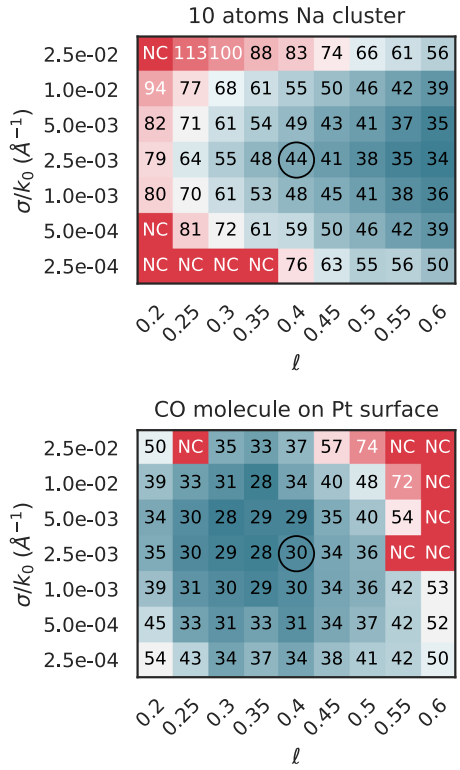


FIG. 2. Average number of DFT calculations required to relax the structures in the training set for the method with fixed bond scales as a function of the effective regularization σ/k_0 and the global dimensionless scale ℓ. The label "NC" stands for at least one relaxation failed with those sets of hyperparameters. The circle marks the values of the hyperparameters that have been used in the subsequent calculations in this paper with fixed bond scales. Both the colors and the figure in the heat map show the average number of steps.

Even with the anisotropy introduced with the inclusion of a model for the bonds in the Gaussian process regression, Fig. 2 shows that the metallic cluster still prefers a longer value of the global scale, ℓ, while for the molecule on the surface, it is more favorable to choose a shorter value. Both systems prefer a value for the regularization $\sigma/k_0 \sim 10^{-3}$ Å^{-1}. Expecting a value for the prefactor of the order $k_0 \sim 1$ eV, this leads to $\sigma \sim 10^{-3}$ eV/Å. This is a reasonable value since an order of magnitude higher would conflict with the convergence threshold of the optimizer (0.01 eV/Å) and the forces are converged with precision 10^{-4} eV/Å. We have chosen the values $\ell = 0.4$, $\sigma = 2.5 \cdot 10^{-3}$ eV/Å, and $k_0 = 1$ eV.

We have also used the average of the covalent radii in Eq. (11) as the initial value for the bond scales when the model is set to update them (and tested that the square root of the product does not produce a significant improvement on the validation set when the other hyperparameters are trained accordingly). For the initial value of the prefactor of the kernel, we have chosen the value that was found for GPMin during training, $k_0 = 2$ eV.

As for the version of the method with updated bond scales, the values of ℓ and σ/k_0, which remain fixed, have been determined by analyzing the performance of the method over the training set, as shown in Fig. 3.

Figure 3 shows that it is easier to find the optimal performance by maximizing the marginal log-likelihood if the initial scales in the GPR underestimate their value, as compared to overestimates, which had already been observed in previous work.[12] This results in an almost flat number of steps as a function of ℓ for the sodium cluster, which prefers overall long scales, but a sharp minimum for the CO on platinum. Our investigations show that the values

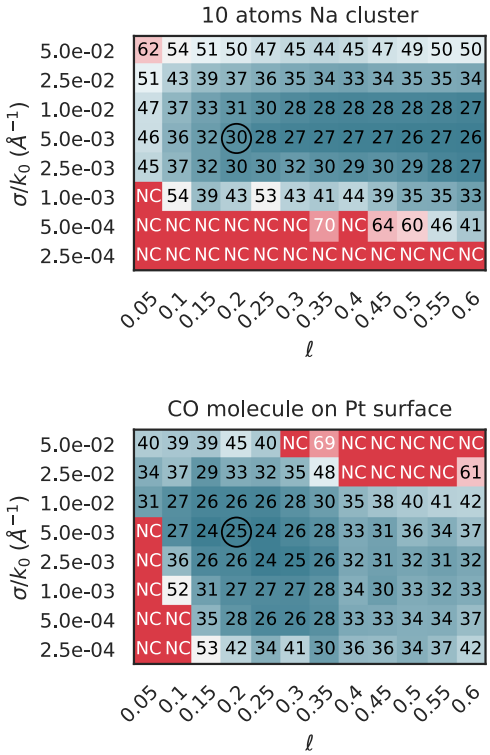


FIG. 3. Average number of DFT calculations required to relax the structures in the training set for the method with updated bond scales as a function of the effective regularization σ/k_0 and the global dimensionless scale ℓ. The label "NC" stands for at least one relaxation failed with those sets of hyperparameters. The circle marks the values of the hyperparameters that have been used in the subsequent calculations in this paper with updated bond scales. Both the colors and the figure in the heat map show the average number of steps.

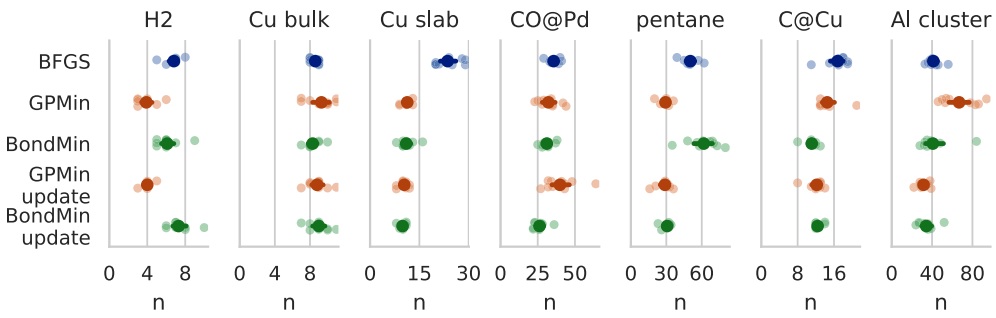


FIG. 4. Number of steps needed to relax the systems in the validation set for different optimizers. Lighter markers represent individual runs, while the darker ones mark the average number of runs and its error bar.

$\sigma = 2.5 \cdot 10^{-3}$ eV/Å and $\ell = 0.2$ result in a good overall performance in both the training and the validation sets.

We conclude Sec. II E by commenting on the performance of the method with and without hyperparameter updates on the ASE/GPAW test set. The performance of each method, along with BFGS with line search as implemented in ASE, and both the default version of GPMin and GPMin with hyperparameter updates are shown in Fig. 4. We compare with BFGS with line search because in a previous study this was clearly the best ASE or SciPy optimizer on this test set.[12]

From Fig. 4, we observe that the performance of the BondMin method, especially with parameter updates, is in general similar to that of GPMin for this class of systems. As noted in Sec. II A, BondMin reduces to GPMin with a different set of hyperparameters for unary systems. We believe this to be the cause for the bad performance on the H_2 system: a scale of 0.12 Å for the method without updates and an initial scale of 0.6 Å for the method with updates force the method to take (at least, initially) very short steps in the configuration space, which makes it difficult to compete with 4 steps in average GPMin. However, we attribute the increased performance in the aluminum cluster to the same effect, where we believe that the new corrected global scale is initially close to optimal.

In contrast, we note the worsening of the performance for the pentane molecule, but we also note that it is improved when hyperparameters are allowed to update. In addition, it seems that the new kernel improves the results of molecules on surfaces the most, especially CO on palladium, which seems to be a difficult problem for GPMin.

We see that the performance of the updated BondMin is rather similar to that of the updated GPmin. In the cases with only one type of interatomic bonds, the two methods should behave similarly. However, in the case of CO/Pd, where metallic, molecular, and molecule–metal bonds are present, BondMin seems to be superior.

III. RESULTS

In the tests on the validation set above, the BondMin optimizer shows an improved performance on a particular subclass of systems: molecules on surfaces. To illustrate this further, we have studied the performance of BondMin with and without bond scale updates as compared to other optimizers for datasets MS5 and C3–4S involving molecules and radicals on surfaces.

The results of MS5 are shown in Fig. 5.

We note that the optimizers of the GPMin family do not show a consistent improvement on this test set as compared to BFGS. In particular, GPmin shows a relatively poor performance on the structures in the bridge initial positions and the NO_2 molecule for all initial positions. In fact, in three of the runs for GPMin with hyperparameter updates on NO_2@Pd with the starting bridge position, the relaxation was terminated and marked as failed after more than 210 steps had been taken without finding the minimum. An additional NO_2@Pd relaxation has failed with the GPMin optimizer and hollow initial position in this case because the model was not able to predict a low energy configuration in 30 consecutive steps. These failed optimizations are marked with inverted triangles in Fig. 5.

In contrast, the BondMin family of optimizers seems to perform well in this test set. The BondMin version without hyperparameter updates consistently shows a similar or lower number of steps than BFGS, and it is also competitive compared to GPMin with hyperparameter updates on the number of steps but with a smaller computational cost.

The BondMin with hyperparameter update optimizer exhibits the lowest number of steps needed to relax all the systems in this test set. As compared with BFGS, it shows an average reduction of over 40% in the number of steps necessary to relax a molecule on a surface. The relative reduction in the number of steps seems to be more pronounced on those systems where the number of steps required by BFGS is large, reaching a factor of 2 reduction for H_2O and $HOCH_2$ on the hollow initial and OH radical in the on-top position and a factor of 2.15 reduction for the CH_3 radical in the "hollow" position.

Furthermore, the BondMin optimizers also show a reduction in the spread in the number of steps among the 10 slightly rattled initial conditions as compared to the other methods. BondMin with hyperparameter updates also shows an average reduction in the standard deviation of the number of steps of over 40% as

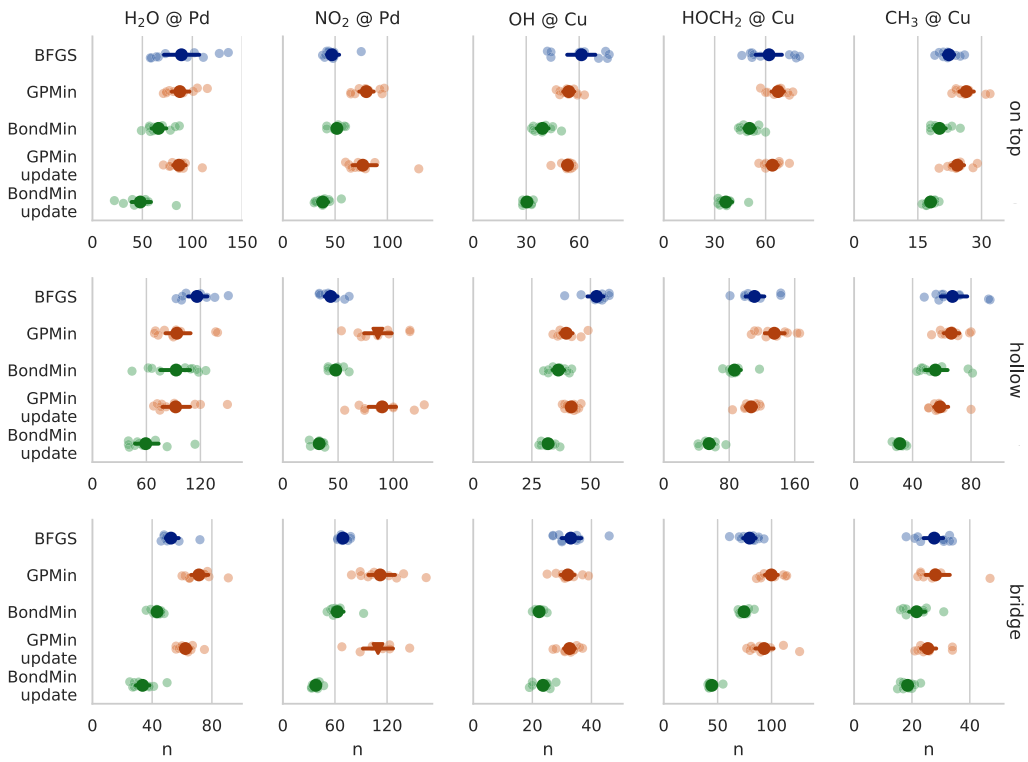


FIG. 5. Number of steps needed to relax different molecules and radicals on fcc (100) slabs for different optimizers. Lighter markers represent individual runs, while the darker ones mark the average number of runs and its error bar. All ten runs converged for systems marked with circles, while the averages for systems where at least one relaxation failed are marked with inverted triangles.

compared with BFGS. Even though the standard deviation is comparable with the one of BFGS for the water molecule, we note that for some of the systems, it can reach up to a factor of 5 reduction, with standard deviations of only 2 or 3 steps in a large fraction of the tests.

We now turn to the results of the limited memory approach (LBondMin). The performance of LBondMin has been studied on the hyperparameter training set systems (where the optimal hyperparameters are known) as a function of the training set size and further tested on two large systems as illustrations.

The results for the hyperparameter training set (a sodium cluster and the CO on Pt) are shown in Fig. 6. The training set size for the PES is now limited to the last N_0 configurations in the light memory version, and the figure shows the number of required minimization steps for different values of N_0. The full memory version is included for comparison.

In the relaxations without hyperparameter updates, the performance seems to saturate to the value of the full memory method relatively fast. In particular, the performance for the sodium cluster is qualitatively indistinguishable between the full memory and low memory versions for the range of training set sizes studied. We note that the performance of the full memory method could have been achieved with a fourth of the training images for the sodium cluster and with half as many training images for the CO on Pt.

Including the update of the hyperparameters, a different picture emerges: the reduction in the number of points in the training set can lead to a reduction in the number of steps needed to relax the structure in the systems studied. This is particularly significant for the sodium cluster, where the average number of steps is reduced by 10%–17% with respect to the full memory method in all of the investigated range of training set sizes. Again, for both systems, one could have used about half of the number of points in the training set, with a modest boost in performance as an effect.

We illustrate the application of the light memory approach on the two systems of larger size in the C3–4S dataset. The results of the optimization can be found in Fig. 7.

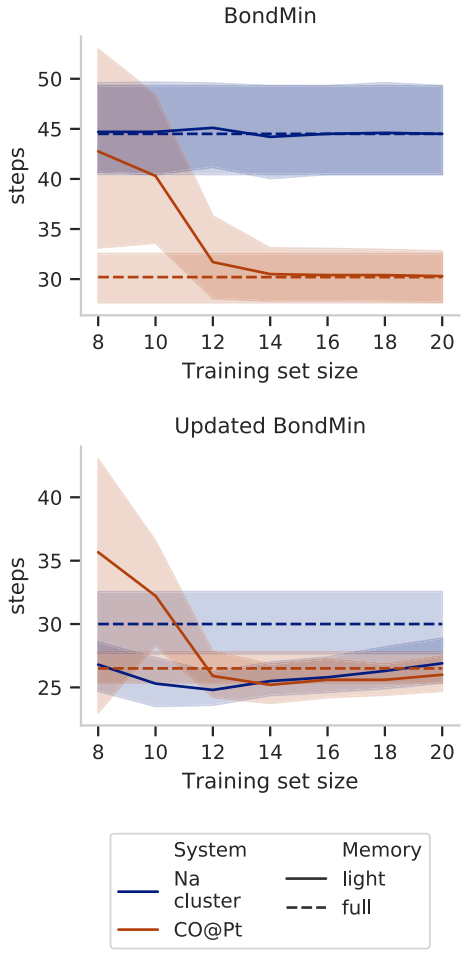


FIG. 6. Average number of steps of the light memory BondMin optimizer as a function of the size of the training set for the two training systems. The shaded areas indicate 95% confidence interval of the mean estimated with bootstrapping. The dashed line shows the results of BondMin with no memory restrictions for the same systems.

For these systems, we have limited the size of the training set to 20 points. The results obtained are consistent with those shown in Figs. 5 and 6: The BondMin optimizer results in a significant reduction in the number of steps needed to relax the system as compared to BFGS even by reducing the number of points in the training set by a factor of 5 (in the case of the butanethiolate radical). Thus, it shows that the reduction in the training set size results in a reduction in computational cost while retaining the performance.

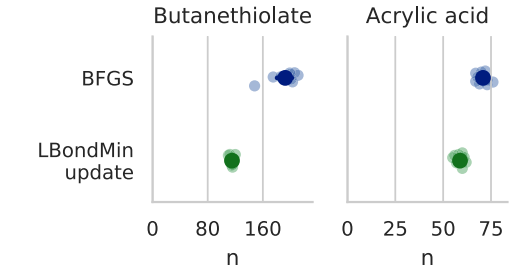


FIG. 7. Number of steps needed for different optimizers to relax the molecules on surfaces in dataset C3–4S. Lighter markers represent individual runs, while the darker ones mark the average number of runs and its error bar.

IV. DISCUSSION

The BondMin optimizer, especially with hyperparameter updates, shows superior performance to the GPMin and BFGS line search optimizers for molecules on surfaces and at the same time comparable performance on a broader range of atomic systems. In particular, it shows speed-ups of up to more than a factor of two on the moderate size adsorbates as compared with BFGS. Moreover, the speed-up seems to increase in those systems where the performance of BFGS is poor and shows robust behavior with small differences in performance for different initial configurations. Thus, the performance of the BondMin method over different systems and initial conditions is not only superior but also more consistent and reliable, as compared with the other methods presented in this article.

We ascribe these improvements to two factors. First, we believe that the ability of the Gaussian process regression to capture both harmonic and anharmonic regimes improves the description of the PES in regions where the PES is not convex (for example, in the vicinity of saddle points), reducing the number of sample points needed to get out of them. This characteristic is shared with other GPR methods such as GPMin, as a contrast to the convex quadratic model in BFGS.

Second, BondMin is able to adapt to anisotropic potential energy surface landscapes with fewer points, as compared with isotropic kernels, which would need a large number of points to describe an anisotropic landscape. We believe that this capacity is the key to success for problems involving molecules on surfaces since they typically involve a combination of stiff and soft bonds that are difficult to capture for GPMin. We also suggest that the reduced number of parameters needed to model the anisotropy as compared to BFGS (i.e., a few bond scales vs the full Hessian) may contribute to the improved performance of BondMin as compared with BFGS line search. We further illustrate this point in Figs. 8 and 9.

The initial bond scales obtained from the covalent radii of the atoms in the system provide a reasonable preconditioning for the different scales involved in the problem, as illustrated in Fig. 5 for the BondMin optimizer without updates of the hyperparameters. However, the ability to update the hyperparameters results

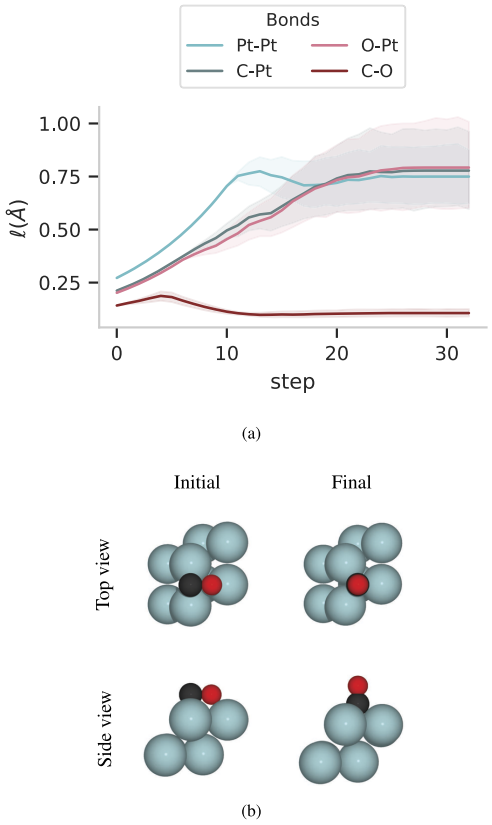


FIG. 8. CO on platinum during the optimization. (a) Evolution of the bond scales as a function of the optimization step. The hyperparameters used are those chosen in Fig. 3. The solid line represents the average over 10 runs, and the shaded area represents the 95% confidence interval of the mean estimated with bootstrapping. (b) Atomic structures of the initial and final atomic configurations. Only the atoms in the unit cell are shown.

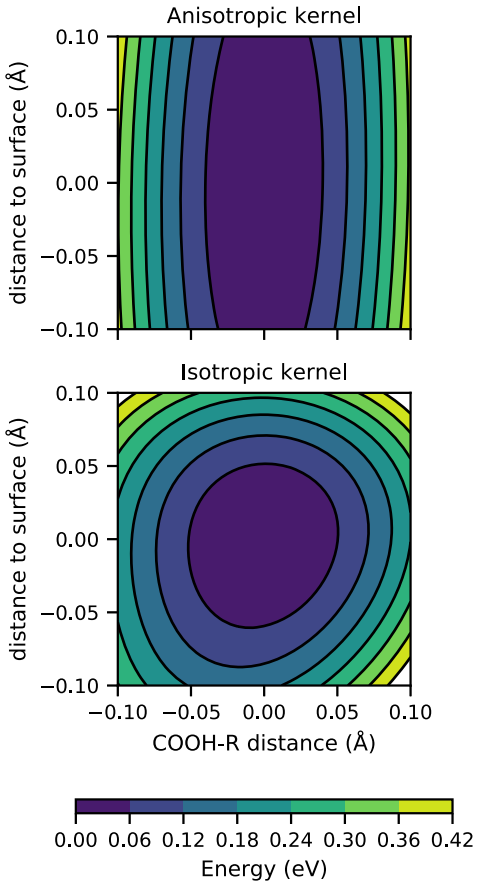


FIG. 9. Surrogate models of the potential energy surface of acrylic acid on palladium around the minimum for two different kernels. The x axis represents the dissociation of the molecule into the carboxyl and vinyl radicals, while the y axis accounts for the variation of the adsorption distance between the molecule and the surface. The anisotropic kernel in the top panel corresponds to the one used by BondMin, while the isotropic kernel is the one used by GPMin.

in a better model of the potential energy surface. Figure 8(a) shows the evolution of the scales $\ell \cdot \ell_{X_i X_j}$ for the different bonds in CO on platinum, with the hyperparameters found during training. The scales start at 0.2 times the average of the covalent radii of the two species and evolve a maximum of a 10% every step to maximize the log marginal likelihood. As evidenced in Fig. 8(b), the main task of the optimizer is to rigidly rotate the CO molecule from being parallel to perpendicular to the surface of the slab.

The carbon–oxygen scale starts at 0.14 Å and ends at 0.11 Å, on average. These are low values compared to the other bonds in Fig. 8(a). The CO scale also presents a comparably low variation between the initial and final scales and low spread, compared to the other bonds. We attribute this to the fact that there is only one CO bond in the atomic structure and to the stiffness of the CO bond. We conclude from this that the machine learning algorithm is able to correctly learn to separate the stiff covalent bonds in molecules (i.e., those exhibiting relatively fast variations in energy as the bond distance is changed) from softer molecule–surface and metallic bonds.

The other bonds, O–Pt, C–Pt, and Pt–Pt, start at 0.20 Å, 0.22 Å, and 0.27 Å, respectively. These seem to be underestimates since by the end of the run, the average value over 10 runs ends up being

around 0.77 Å for these three bond scales. Initially, the Pt–Pt bond scale grows faster than the O–Pt and C–Pt scales, reaching the final value earlier. We attribute this behavior to the fact that the Pt slab is initially very close to the relaxed configuration and it can thus relax to it very fast, while the rotation of the molecule provides an increasingly diverse training set as the optimizer gathers more data. It can be noted that these scales are shorter than those found by GPMin on the sodium cluster, which were always longer than 1 Å, which might have to do with the relatively soft bonds in the cluster.

The optimized molecule-Pt scales show a larger variation across the 10 different relaxations than the Pt–Pt ones by the end of the run. We attribute this to the ability of the method to adapt itself to fit the minimization trajectory the best: differences in the configurations sampled by the optimizer might lead to differences in the relevance of different "bonds" in different training structures, thus leading to slightly different models.

The introduction of different scales associated with the different bonds leads to models that can capture anisotropy better with fewer data points, as illustrated in Fig. 9 for acrylic acid on palladium (dataset C3–4S). The figure shows the level sets of the models of the PES underlying BondMin and GPMin, respectively, around the minimum. Both regressions have been trained on the data from the trajectory of one of the LBondMin energy minimizations for this system, and the hyperparameters have been fully optimized. The plot shows the variation in the potential energy when breaking the single C–C bond that splits the molecule into the carboxyl and vinyl radicals and translating the whole molecule along the z direction, away or toward the surface.

The scales for the bonds between the C, O, and H atoms and palladium saturate at between 3.6 Å and 3.8 Å, in contrast with the 0.14 Å of the C–C bond. This produces weights in Eq. (5) between 600 and 700 times smaller for the translation of the molecule with respect to the surface as compared with its dissociation, resulting in a more anisotropic PES, as illustrated in the top panel of Fig. 9. In contrast, the optimization of the scale in GPMin over these same training points yields a value of 0.6 Å, which is a compromise between the long scales and short scales found by BondMin. The GPMin model for the potential energy surface does not capture the different nature of the two bonds with this amount of information, even when the training set contains the minimum and neighboring points. The Gaussian process model underlying GPMin is less reliable at extrapolating, and as a result, the GPMin optimizer would in general need more points to describe the surface and find its minimum.

The surrogate model of the potential energy surface presented in this work is invariant under rigid translations of the whole system, which turns out to be both a blessing and a curse. Even though the underlying physics is translationally invariant, we note that the numerical solution to the Kohn–Sham equations does not need to fully obey this. For instance, if the grid in real space is too coarse, this might lead to an egg-box effect,[54] resulting in a small translation-dependent spurious potential.

As a result, combining a DFT method whose parameters are not finely tuned together with a very tight threshold for the optimization step can result in the optimizer failing to converge. To our experience, this feature becomes particularly relevant for systems where some of the atoms are constrained to stay fixed: the local minimum might be in a direction that would represent a translation, and the surrogate model would never be able to capture that.

We suggest that, in most cases, the solution to this situation is to reconsider the accuracy that is needed for the particular application and either increase the convergence of the DFT method or relax the tolerance for the convergence of the optimization method. Notwithstanding, we have considered some possible alternative solutions when the above is not possible.

One possibility would be to explicitly break the translational invariance of the model, by adding a soft-mode for rigid translations. This can be done by defining a new matrix $\tilde{G}$,

$$\tilde{G} = G + \ell_T^{-2}(\mathbf{t}_x\mathbf{t}_x^T + \mathbf{t}_y\mathbf{t}_y^T + \mathbf{t}_z\mathbf{t}_z^T), \tag{12}$$

where ℓ_T is the scale for the translation mode and $\mathbf{t}_x$, $\mathbf{t}_y$, and $\mathbf{t}_z$ are the unitary vectors generating the translations of the system along the three axis.

Another possibility is to redefine convergence: by defining the corrected forces on the atoms to be the DFT forces minus the average force over all atoms such that the sum of all forces is zero, one can redefine the convergence criterium of the optimizer as having the maximum corrected force among the atoms to fall under a certain threshold. This approach results in an approximate best structure given the circumstances, which could then be finely tuned with a more precise DFT method. We note that when using this approach, the Gaussian process still needs to be trained on the uncorrected forces (those not being translationally invariant) since training on corrected forces would introduce an energy-force noise term in the model.

We now turn to a discussion of the approach where the number of training points is limited to a constant number. As far as we know, there is no easy rule of thumb to determine the number of neighboring points one should include in the training in order to obtain the optimal speed-up. The sodium cluster and the CO molecule on platinum in Fig. 6 have the same number of atoms in the system and show different optimal numbers of points in the training set. Moreover, the potential gain (if any) compared to using the full dataset also seems to vary from system to system.

Considering the poor scaling of the full memory approach with the number of atoms in the system, we consider that the better scaling of the light memory approach at no significant reduction in performance for a wide range of values of the training set size makes it the method of choice for large systems. In such cases, the number of points to include in the training set should be chosen by the user under a consideration of the computational resources available for the problem at hand.

Let us finally note that all the molecules on surfaces discussed in this article bind to the surfaces using RPBE for the exchange–correlation energy. We have chosen not to show systems that do not bind since, for such a system, the PES does not have a clear and well-defined minimum. This makes step-counts as a measure of performance difficult to interpret. We have included some tests on non-bonding systems in the supplementary material, where we show that BondMin still performs well in finding the minimum for such systems.

V. CONCLUSION

We have presented three versions of a local optimization method based on a model composed of preconditioned radial functions: a full memory version with hyperparameter updates, the same method without updates, and a light version with less memory requirement. We have shown that the full memory version with hyperparameter updates reduces the average number of steps needed to relax molecules on surfaces in a robust manner, with potential speed-ups of up to a factor of 2 compared to BFGS, depending on the system. The light memory version works with a reduced training set, which might be necessary for large systems in the present implementation. Surprisingly, the limitation of the training set might in some cases lead to superior performance. A reimplementation of the method using parallelization and distributed techniques[55,56] would make the method benefit from the kind of speed-ups most DFT implementations are already taking advantage of when executed at large supercomputing facilities.

The method presented obtains comparable improvements over standard optimizers to those presented in other works using Gaussian processes[10,13] on other classes of materials while retaining a good overall performance on a wide class of systems. In several references, the boost in performance has been attributed to the use of non-stationary kernels[13,18] as a way to include relevant information relative to chemical bonding. In contrast, we show that the preconditioning of an isotropic stationary kernel can in fact include a crucial fraction of the bond information in an unbiased way. This is particularly useful for adsorption systems, where the bonds inside the molecule, inside the substrate, and between the two might involve different length scales. As proposed by Meyer and Hauser,[13] the combination of preconditioning with non-stationary kernels might bring a further reduction in the computational time needed in local explorations of the potential energy surface.

As discussed by Garrido Torres *et al.*,[40] a further gain compared to traditional methods can be achieved by using the method for several calculations on the same system, for example, when relaxing the same molecule on different sites. The first relaxation would exhibit a speed-up comparable to the one discussed in this paper, and the subsequent ones would benefit of better initial estimates of the hyperparameters as well as of the energies and forces from previous relaxations.

Finally, we note that the choice of the initial preconditioning as the average of the covalent radii is physically reasonable but also somewhat arbitrary. We have shown that this choice improves the performance in systems where the PES is anisotropic and the number of steps is large, but we have also reported that it severely underestimates the ratio between molecular bond scales and the scales of molecule-surface bonds. Along this line, we note that the implementation of the method is flexible enough to allow for other user-defined choices of the initial scales, for example, van der Waals radii,[57] results from previous similar calculations, or parameters extracted from semi-empirical models.[58]

SUPPLEMENTARY MATERIAL

See the supplementary material for the binding energies and number of steps in the relaxations with different optimizers for additional systems.

ACKNOWLEDGMENTS

We acknowledge support from the VILLUM Center for the Science of Sustainable Fuels and Chemicals, which is funded by the VILLUM Fonden research grant (No. 9455). We also acknowledge support from the U.S. Department of Energy, Chemical Sciences, Geosciences, and Biosciences (CSGB) Division of the Office of Basic Energy Sciences, via Grant No. DE-AC02-76SF00515, to the SUNCAT Center for Interface Science and Catalysis.

DATA AVAILABILITY

The data that support the findings of this study are openly available at https://cmr.fysik.dtu.dk/bondmin/bondmin.html#bondmin. The code for the optimizer is available at https://gitlab.com/egarijo/bondmin.

REFERENCES

[1] J. Nocedal and S. Wright, *Numerical Optimization* (Springer Science & Business Media, 2006).
[2] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch, "Structural relaxation made simple," Phys. Rev. Lett. **97**, 170201 (2006).
[3] R. Lindh, A. Bernhardsson, G. Karlström, and P.-Å. Malmqvist, "On the use of a hessian model function in molecular geometry optimizations," Chem. Phys. Lett. **241**, 423–428 (1995).
[4] S. K. Burger and P. W. Ayers, "Quasi-Newton parallel geometry optimization methods," J. Chem. Phys. **133**, 034116 (2010).
[5] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," Phys. Rev. **136**, 864–871 (1964).
[6] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," Phys. Rev. **140**, 1133–1138 (1965).
[7] D. Packwood, J. Kermode, L. Mones, N. Bernstein, J. Woolley, N. Gould, C. Ortner, and G. Csányi, "A universal preconditioner for simulating condensed phase materials," J. Chem. Phys. **144**, 164109 (2016).
[8] L. Mones, C. Ortner, and G. Csányi, "Preconditioners for the geometry optimisation and saddle point search of molecular systems," Sci. Rep. **8**, 13991 (2018).
[9] S. Makri, C. Ortner, and J. R. Kermode, "A preconditioning scheme for minimum energy path finding methods," J. Chem. Phys. **150**, 094109 (2019).
[10] A. Denzel and J. Kästner, "Gaussian process regression for geometry optimization," J. Chem. Phys. **148**, 094114 (2018).
[11] G. Schmitz and O. Christiansen, "Gaussian process regression to accelerate geometry optimizations relying on numerical differentiation," J. Chem. Phys. **148**, 241704 (2018).
[12] E. Garijo del Río, J. J. Mortensen, and K. W. Jacobsen, "Local Bayesian optimizer for atomic structures," Phys. Rev. B **100**, 104103 (2019).
[13] R. Meyer and A. W. Hauser, "Geometry optimization using Gaussian process regression in internal coordinate systems," J. Chem. Phys. **152**, 084112 (2020).
[14] E. V. Podryabinkin and A. V. Shapeev, "Active learning of linearly parametrized interatomic potentials," Comput. Mater. Sci. **140**, 171–180 (2017).
[15] K. Gubaev, E. V. Podryabinkin, G. L. W. Hart, and A. V. Shapeev, "Accelerating high-throughput searches for new alloys with active learning of interatomic potentials," Comput. Mater. Sci. **156**, 148–156 (2019).
[16] A. A. Peterson, "Acceleration of saddle-point searches with machine learning," J. Chem. Phys. **145**, 074106 (2016).
[17] O.-P. Koistinen, F. B. Dagbjartsdóttir, V. Ásgeirsson, A. Vehtari, and H. Jónsson, "Nudged elastic band calculations accelerated with Gaussian process regression," J. Chem. Phys. **147**, 152720 (2017).

[18]O.-P. Koistinen, V. Ásgeirsson, A. Vehtari, and H. Jónsson, "Nudged elastic band calculations accelerated with Gaussian process regression based on inverse interatomic distances," J. Chem. Theory Comput. **15**, 6738–6751 (2019).
[19]A. Denzel and J. Kästner, "Gaussian process regression for transition state search," J. Chem. Theory Comput. **14**, 5777–5786 (2018).
[20]A. Denzel and J. Kästner, "Hessian matrix update scheme for transition state search based on Gaussian process regression," J. Chem. Theory Comput. **16**, 5083–5089 (2020).
[21]J. A. Garrido Torres, P. C. Jennings, M. H. Hansen, J. R. Boes, and T. Bligaard, "Low-scaling algorithm for nudged elastic band calculations using a surrogate machine learning model," Phys. Rev. Lett. **122**, 156001 (2019).
[22]M. S. Jørgensen, U. F. Larsen, K. W. Jacobsen, and B. Hammer, "Exploration versus exploitation in global atomistic structure optimization," J. Phys. Chem. A **122**, 1504–1509 (2018).
[23]V. L. Deringer, C. J. Pickard, and G. Csányi, "Data-driven learning of total and local energies in elemental boron," Phys. Rev. Lett. **120**, 156001 (2018).
[24]T. Yamashita, N. Sato, H. Kino, T. Miyake, K. Tsuda, and T. Oguchi, "Crystal structure prediction accelerated by Bayesian optimization," Phys. Rev. Mater. **2**, 013803 (2018).
[25]M. Todorović, M. U. Gutmann, J. Corander, and P. Rinke, "Bayesian inference of atomistic structure in functional materials," npj Comput. Mater. **5**, 35 (2019).
[26]M. K. Bisbo and B. Hammer, "Efficient global structure optimization with a machine-learned surrogate model," Phys. Rev. Lett. **124**, 086102 (2020).
[27]H. L. Mortensen, S. A. Meldgaard, M. K. Bisbo, M.-P. V. Christiansen, and B. Hammer, "Atomistic structure learning algorithm with surrogate energy model relaxation," Phys. Rev. B **102**, 075427 (2020).
[28]L. Fang, E. Makkonen, M. Todorovic, P. Rinke, and X. Chen, "Efficient cysteine conformer search with Bayesian optimization," arXiv:2006.15006 [physics.comp-ph] (2020).
[29]C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning* (MIT press Cambridge, MA, 2006).
[30]A. S. Christensen and O. A. von Lilienfeld, "On the role of gradients for machine learning of molecular energies and forces," Mach. Learn.: Sci. Technol. **1**, 045018 (2020).
[31]V. L. Deringer and G. Csányi, "Machine learning based interatomic potential for amorphous carbon," Phys. Rev. B **95**, 094203 (2017).
[32]S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, "Machine learning of accurate energy-conserving molecular force fields," Sci. Adv. **3**, e1603015 (2017).
[33]A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," Phys. Rev. Lett. **104**, 136403 (2010).
[34]A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," Phys. Rev. B **87**, 184115 (2013).
[35]M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," Phys. Rev. Lett. **108**, 058301 (2012).
[36]K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. Von Lilienfeld, K.-R. Müller, and A. Tkatchenko, "Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space," J. Phys. Chem. Lett. **6**, 2326–2331 (2015).
[37]M. O. Jäger, E. V. Morooka, F. F. Canova, L. Himanen, and A. S. Foster, "Machine learning hydrogen adsorption on nanoclusters through structural descriptors," npj Comput. Mater. **4**, 37 (2018).
[38]J. Wu, M. Poloczek, A. G. Wilson, and P. Frazier, "Bayesian optimization with gradients," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2017), pp. 5267–5278, available at https://papers.nips.cc/paper/2017/hash/64a08e5f1e6c39faeb90108c430eb120-Abstract.html.
[39]L. Himanen, M. O. J. Jäger, E. V. Morooka, F. Federici Canova, Y. S. Ranawat, D. Z. Gao, P. Rinke, and A. S. Foster, "Dscribe: Library of descriptors for machine learning in materials science," Comput. Phys. Commun. **247**, 106949 (2020).
[40]J. A. Garrido Torres, E. Garijo del Río, V. Streibel, M. H. Hansen, T. S. Choksi, J. J. Mortensen, A. Urban, M. Bajdich, F. Abild-Pedersen, K. W. Jacobsen, and T. Bligaard, "An artificial intelligence approach for navigating potential energy surfaces" (unpublished) (2020).
[41]Atomic Simulation Environment (ASE), https://wiki.fysik.dtu.dk/ase/, 2020.
[42]R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," SIAM J. Sci. Comput. **16**, 1190–1208 (1995).
[43]E. Jones, T. Oliphant, and P. Peterson, SciPy: Open source scientific tools for Python, http://www.scipy.org/(2001–).
[44]D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek, "Scalable global optimization via local Bayesian optimization," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2019), pp. 5496–5507, available at https://proceedings.neurips.cc/paper/2019/hash/6c990b7aca7bc7058f5e98ea909e924b-Abstract.html.
[45]A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, "The atomic simulation environment—A python library for working with atoms," J. Phys.: Condens. Matter **29**, 273002 (2017).
[46]J. Enkovaara, C. Rostgaard, J. J. Mortensen, J. Chen, M. Dułak, L. Ferrighi, J. Gavnholt, C. Glinsvad, V. Haikola, H. A. Hansen, H. H. Kristoffersen, M. Kuisma, A. H. Larsen, L. Lehtovaara, M. Ljungberg, O. Lopez-Acevedo, P. G. Moses, J. Ojanen, T. Olsen, V. Petzold, N. A. Romero, J. Stausholm-Møller, M. Strange, G. A. Tritsaris, M. Vanin, M. Walter, B. Hammer, H. Häkkinen, G. K. H. Madsen, R. M. Nieminen, J. K. Nørskov, M. Puska, T. T. Rantala, J. Schiøtz, K. S. Thygesen, and K. W. Jacobsen, "Electronic structure calculations with GPAW: A real-space implementation of the projector augmented-wave method," J. Phys.: Condens. Matter **22**, 253202 (2010).
[47]B. Hammer, L. B. Hansen, and J. K. Nørskov, "Improved adsorption energetics within density-functional theory using revised Perdew-Burke-Ernzerhof functionals," Phys. Rev. B **59**, 7413–7421 (1999).
[48]G. Kresse and D. Joubert, "From ultrasoft pseudopotentials to the projector augmented-wave method," Phys. Rev. B **59**, 1758 (1999).
[49]Optimizer tests - GPAW, https://wiki.fysik.dtu.dk/gpaw/devel/ase_optimize/ase_optimize.html, 2020.
[50]L. A. Curtiss, K. Raghavachari, P. C. Redfern, and J. A. Pople, "Assessment of Gaussian-2 and density functional theories for the computation of enthalpies of formation," J. Chem. Phys. **106**, 1063–1079 (1997).
[51]National Center for Biotechnology Information, Pubchem compound summary for cid 6581, acrylic acid, 2020, retrieved August 25, 2020.
[52]National Center for Biotechnology Information, Pubchem compound summary for cid 8012, 1-butanethiol, 2020, retrieved August 25, 2020.
[53]B. Cordero, V. Gómez, A. E. Platero-Prats, M. Revés, J. Echeverría, E. Cremades, F. Barragán, and S. Alvarez, "Covalent radii revisited," Dalton Trans. **2008**, 2832–2838.
[54]F. Nogueira, A. Castro, and M. A. Marques, "A tutorial on density functional theory," in *A Primer in Density Functional Theory* (Springer, 2003), pp. 218–256.
[55]J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson, "Gpytorch: Blackbox matrix-matrix Gaussian process inference with gpu acceleration," in *Advances in Neural Information Processing Systems*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Inc., 2018), Vol. 31, pp. 7576–7586.
[56]K. Wang, G. Pleiss, J. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson, "Exact Gaussian processes on a million data points," in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), Vol. 32, pp. 14648–14659.
[57]S. Alvarez, "A cartography of the van der waals territories," Dalton Trans. **42**, 8617–8636 (2013).
[58]E. B. Tadmor, R. S. Elliott, J. P. Sethna, R. E. Miller, and C. A. Becker, "The potential of atomistic simulations and the knowledgebase of interatomic models," JOM **63**, 17 (2011).

Paper V

Global optimization of atomic structures with gradient-enhanced Gaussian process regression

Sami Kaappa, **Estefanía Garijo del Río**, Karsten Wedel Jacobsen

Submitted

Global optimization of atomic structures with gradient-enhanced Gaussian process regression

Sami Kaappa, Estefanía Garijo del Río, and Karsten Wedel Jacobsen
Department of Physics, Technical University of Denmark, Kongens Lyngby, Denmark
(Dated: April 21, 2021)

Determination of atomic structures is a key challenge in the fields of computational physics and materials science, as a large variety of mechanical, chemical, electronic, and optical properties depend sensitively on structure. Here, we present a global optimization scheme where energy and force information from density functional theory (DFT) calculations is transferred to a probabilistic surrogate model to estimate both the potential energy surface (PES) and the associated uncertainties. The local minima in the surrogate PES are then used to guide the search for the global minimum in the DFT potential. We find that adding the gradients in most cases improves the efficiency of the search significantly. The method is applied to global optimization of $[Ta_2O_5]_x$ clusters with $x = 1, 2, 3$, and the surface structure of oxidized ZrN.

I. INTRODUCTION

Global optimization in high-dimensional space is a long-standing challenge in numerical analysis, and also in physics, chemistry, and material science. The structure of an atomic system is at low temperature given by the global minimum point of the potential energy surface (PES), which is a function, $E(\mathbf{x})$, of all atomic coordinates $\mathbf{x}$. For atomic systems with more than a few atoms, the dimensionality therefore constitutes a challenge. Furthermore, the PES is usually determined by a quantum mechanical calculation with for example density functional theory (DFT), and these calculations are computationally demanding so the optimization should therefore be performed with as few function evaluations as possible.

Numerous algorithms for finding the structural ground state of a system are implemented for material science problems [1], such as basin hopping [2], evolutionary algorithms, [3–5], particle swarm optimization [6], and random searches [7], but the issue with these methods remains the large number of DFT evaluations required by the algorithms.

Recently, machine-learned surrogate models have been considered in order to overcome the problem of spending excessive amounts of computer resources on DFT calculations. A surrogate model for the PES is constructed based on a dataset typically obtained with DFT, and it allows for subsequent much faster evaluation of atomic energies and forces. Surrogate models have been used in local optimization [8], global optimization [4, 9–15], nudged-elastic band calculations [16, 17], searches for transition states [18], adsorption studies [19], and the design of force fields [20–22].

Many of the surrogate models are based on Bayesian inference, or Gaussian processes (GP) [23, 24], where the resulting PES is a sum of joint kernel functions, centered at the training points. In its most traditional form, the GP is only trained with the target values, i.e. the electronic ground-state energies in the context of computational chemistry. However, since forces are readily available after a ground state DFT calculation, we train the model also with the gradients of the target value, i.e. with the forces on each atom. The inclusion of gradients is crucial in local structure optimization based on surrogate models [8] and has also been shown to generally improve model predictions [25].

The construction of the PES based on a GP usually involves the introduction of a distance measure or similarity of two different atomic configurations. If two configurations are close, it is assumed that energies and forces will be close as well. To this end it may be advantageous to describe the atomic configurations using a structural fingerprint (alias descriptor), which in the simplest case that we shall consider here, is simply a mapping from the atomic Cartesian coordinates to a (typically high-dimensional) vector. The similarity of two atomic configurations can then be estimated based on the difference between the two fingerprint vectors.

The introduction of a fingerprint vector may have several advantages. For example, a fingerprint can be constructed to reflect the translational, rotational, and permutational invariances of the atomic configuration, *i.e.* if two configurations differ by only a permutation of identical atoms, the fingerprint will be unchanged. This has the consequence that the predicted PES will exhibit the same symmetries. Furthermore, a good descriptor is able to catch the relevant information of the configuration for the underlying problem. A simple example of an atomic fingerprint is the Coulomb matrix [26], which represents the atomic configuration using inverse distances, but more elaborate fingerprints have been developed during the latest years, such as SOAP [27], ACSF [28], many-body tensor representation [29], and FCHL [30]. Most of the common fingerprints are ready-to-use in DScribe package [31], although currently the gradients of the descriptions are not available, that are highly relevant in optimization problems.

In this work, we use Bayesian optimization [32] in order to find global minimum structures for various systems. The work follows the pioneering approach for efficient global optimization of atomic structures by Bisbo and Hammer [13, 33] with the essential difference that we

train our GP regression model with both energies and forces. We also note that the implementation of the approach involves many choices and parameters, where we might differ from Bisbo and Hammer. For example, we use a similar global fingerprint, but introduce an additional smooth cutoff function to obtain a smoother representation of the gradients. In general, the gradients are seen to improve the efficiency of the global optimization, and we illustrate this through applications to a 15-atom Cu cluster, bulk SiO_2, Ti_4O_8 cluster, bulk TiO_2 and bulk silicon.

This article is organized as follows. In section II, we describe the surrogate model that we use to predict energies and forces of atomic structures during a global search. In section III, we illustrate the predictive power of the model by generating learning curves for a Cu_{15} cluster and bulk SiO_2. In section IV, we describe the global optimization approach, and then in section V we demonstrate its performance on a Cu_{15} cluster, bulk SiO_2, $[Ta_2O_5]_x$ clusters, a ZrN-O surface, a Ti_4O_8 cluster, bulk TiO_2, and bulk silicon. In section VI, we discuss computational performance issues before we finally conclude.

II. SURROGATE MODEL

A. Gaussian process with gradients

To model the potential energy surface of an atomic structure, we use a Gaussian process that learns energies and forces (*i.e.* negative gradients) from existing data. A Gaussian process uses Bayesian inference and is based on the assumption that the prior distribution for the data is given by a multi-dimensional normal distribution. The result is that the predicted energy and forces, $\mu(\mathbf{x}) = (E(\mathbf{x}), -\mathbf{F}(\mathbf{x}))$, at a given atomic configuration $\mathbf{x}$ with fingerprint $\rho(\mathbf{x})$ can be described [23, 34] as

$$\mu(\mathbf{x}) = \mu_p(\mathbf{x}) + K(\rho(\mathbf{x}), P)C(P,P)^{-1}(y - \mu_p(X)), \quad (1)$$

where X is a list with the atomic configurations of the training data and $P = \rho(X)$ a list with the corresponding fingerprints. y is a vector that contains energies and negative forces of the training data, μ_p denotes the prior energy and negative forces ($\mu_p(\mathbf{x})$ for the given configuration and $\mu_p(X)$ for the training data points), K denotes the covariance matrix, and C is the regularized K-matrix of covariances between training data points. The forces are inserted as their negatives because the mathematical expression of a Gaussian process works with the gradients and not with the forces. The resulting vector $\mu(\mathbf{x})$ contains the predicted total energy and the negative predicted forces for each atom (in Cartesian coordinates). For the sake of readability, we will denote $\rho = \rho(\mathbf{x})$ for the fingerprint for the rest of this section.

The covariance matrix K is built as follows. A covariance matrix $\tilde{K}$ between two atomic configurations is written as [34]

$$\tilde{K}(\rho_1, \rho_2) = \begin{pmatrix} k(\rho_1,\rho_2) & (\nabla_2 k(\rho_1,\rho_2))^T \\ \nabla_1 k(\rho_1,\rho_2) & \nabla_1 (\nabla_2 k(\rho_1,\rho_2))^T \end{pmatrix} \quad (2)$$

with kernel (or covariance) function $k(\rho_1, \rho_2)$. Here, ∇_i operates on the Cartesian coordinates of the atomic configuration $\mathbf{x}_i$, represented by the fingerprint ρ_i. The components in the matrix on the right-hand side of Eq. 2 can be interpreted as the covariances between energies (k), covariances between energies and forces ($\nabla_i k$) and covariances between different force components ($\nabla_i \nabla_j k$).

Using the formulation of $\tilde{K}$, we store the covariances between a single fingerprint ρ and all the training point fingerprints in P in the matrix $K(\rho, P)$ as

$$K_j(\rho, P) = \tilde{K}(\rho, P_j) \quad (3)$$

where j runs over the number of training points. Finally, the matrix C contains the covariance between all the training points and a diagonal regularization that describes the estimated noise, or uncertainty, of the training data. Its elements are thus given by

$$C_{ij}(P,P) = \tilde{K}(P_i, P_j) + \mathrm{diag}\left(\sigma_{n,E}^2, \sigma_{n,f}^2\right)\delta_{ij} \quad (4)$$

where $\mathrm{diag}(\sigma_{n,E}^2, \sigma_{n,f}^2)$ represents the diagonal matrix with $\sigma_{n,E}^2$ in the first entry and $\sigma_{n,f}^2$ in the remaining ones. In this way, we have introduced separate regularizations $\sigma_{n,E}^2$ and $\sigma_{n,f}^2$ for energy and force covariances, respectively. Throughout this work, the regularization is set so that the ratios between the regularization parameters and the kernel prefactor σ (defined below in equation 9) are $\sigma_{n,E}/\sigma = 0.0005$ for energies and $\sigma_{n,f}/\sigma = 0.001$ for forces.

The power of using Bayesian inference in searching the global minimum comes from the estimated uncertainties of the predictions that are easily attainable. For the Gaussian process, the estimated standard deviation is given by [23]

$$\Sigma(\mathbf{x}) = \left[\tilde{K}(\rho,\rho) - K(\rho,P)C(P,P)^{-1}K(P,\rho)\right]^{1/2} \quad (5)$$

The uncertainties are used for the global optimization through the acquisition function as described below in section IV.

B. The model

We describe the atomic structure by a fingerprint which has two terms: a radial distribution and an angular distribution. Using such global distributions ensure the rotational, translational, and permutational symmetries for a system. The radial distribution is motivated by the radial distribution function described by Valle and Oganov [35]. However, to remove discontinuity at the

cutoff distance, we use a smooth weighting factor. The radial part of the fingerprint for element pair AB is calculated as

$$\rho^R_{AB}(r;\mathbf{x}) = \sum_{\substack{i\in A\\ j\in B}} \frac{1}{r_{ij}^2} f_c(r_{ij};R^R_c)\, e^{-|r-r_{ij}|^2/2\delta_R^2} \quad (6)$$

where r_{ij} is distance between atoms i and j in the set of coordinates $\mathbf{x}$, and δ_R is a smearing factor with a fixed value $\delta_R = 0.4$ Å. A and B indicate different elements in the system, and the i-sum goes only over atoms that are of element A and the j-sum goes only over atoms that are of element B. r denotes the discrete variable with 200 values, ranging from 0 to the cutoff distance R^R_c. The smooth function f_c has the form

$$f_c(r;R_c) = \begin{cases} 1-(1+\gamma)\left(\frac{r}{R_c}\right)^{\gamma} + \gamma\left(\frac{r}{R_c}\right)^{1+\gamma} & \text{if } r \le R_c \\ 0 & \text{if } r > R_c \end{cases} \quad (7)$$

with a cutoff distance R_c and $\gamma = 2$. This form for $f_c(r)$ has zero value and zero derivative at $r = R_c$. Due to the factor $1/r^2$, the form of equation 6 has the property of giving more weight on small distances below R^R_c. The angular part of the fingerprint is given by

$$\rho^\alpha_{ABC}(\theta;\mathbf{x}) = \sum_{\substack{i\in A\\ j\in B\\ k\in C}} f_c(r_{ij};R^\alpha_c) f_c(r_{jk};R^\alpha_c)\, e^{-|\theta-\theta_{ijk}|^2/2\delta_\alpha^2} \quad (8)$$

where θ is a discrete variable with 100 values that range from 0 to π, θ_{ijk} is the angle between atoms i, j and k, and δ_α is a smearing factor with a fixed value $\delta_\alpha = 0.4$ rad. The chosen values for δ_R and δ_α were determined by trying a few different values. The ones chosen were observed to work well for all the systems studied in this work. In the smooth function f_c we use the value of $\gamma = 0.5$ that again ensures a smooth behavior of the fingerprint at cutoff $R_c = R^\alpha_c$. The different γ value to that of the radial part comes from our observation that the predicted potential energy surfaces were not smooth enough at the angular cutoff radius when using the value of $\gamma = 2$ in the angular part. In our work, the cutoff radius R^R_c has values between 4.0 and 8.0 Å, and R^α_c has values between 3.6 and 4.0 Å, comparable to the radii studied in, e.g., [22]. The fingerprint is very similar to the one used by Bisbo and Hammer [33] but with the additional cut-off function for the radial part.

The total fingerprint for an atomic configuration $\mathbf{x}$ is obtained by concatenating all vectors $\rho^R_{AB}(r;\mathbf{x})$ and $\rho^\alpha_{ABC}(\theta;\mathbf{x})$ with elements A, B and C of the system, resulting in a single vector that we denote as ρ. To clarify, for a single-element system such as a Cu cluster, the radial part is just $\rho^R = \rho^R_{\mathrm{Cu,Cu}}(r;\mathbf{x})$, and for a two-element system, like SiO_2, the radial fingerprint consists of vectors $\rho^R_{\mathrm{Si,Si}}(r;\mathbf{x})$, $\rho^R_{\mathrm{Si,O}}(r;\mathbf{x})$, $\rho^R_{\mathrm{O,Si}}(r;\mathbf{x})$, and $\rho^R_{\mathrm{O,O}}(r;\mathbf{x})$. A similar procedure is used for the angular parts of the fingerprint.

We calculate the covariance between data points using a squared-exponential kernel

$$k(\rho_1,\rho_2) = \sigma^2 \exp\left(\frac{-D(\rho_1,\rho_2)^2}{2l^2}\right) \quad (9)$$

with the distance function $D(\rho_1,\rho_2)$ and two descriptive hyperparameters, the prefactor σ and length scale l. (Note, that we use the term prefactor for σ and not σ^2.) It is good to note that, although the dimensionality of the fingerprint can be thousands, the kernel function only includes distances between the fingerprint vectors. Therefore, the efficiency of a Gaussian process does not suffer from the high dimensionality of the fingerprint.

The distance function we take as simply the Euclidean distance between the fingerprint vectors as

$$D(\rho_1,\rho_2) = \left[\sum_i (\rho_{1i}-\rho_{2i})^2\right]^{1/2}. \quad (10)$$

Since the gradients of the kernel function in Eq. 9 are required by a Gaussian process that is trained on forces (in accordance to Eq. 2), the full formulas to calculate the gradients with this specific distance function in fingerprint space are given in the Supplemental information [36]. The forces can be predicted also for the model that is trained on energies only, as we also show in the Supplemental information [36].

We note that Bisbo and Hammer [33] use a kernel function, which is a sum of two squared-exponential kernels with two different length scales. We tried this, but did not see any systematic improvement by adding an extra length scale.

We determine the hyperparameters σ and l, by maximizing the logarithmic marginal likelihood, which is written as [23]

$$\begin{aligned} \log\mathcal{P} = &-\frac{1}{2}\log(\det C(P,P)) \\ &-\frac{1}{2}(y-\mu_p(X))^T C(P,P)^{-1}(y-\mu_p(X)) \\ &-\frac{N(3N_{\mathrm{atoms}}+1)}{2}\log 2\pi \end{aligned} \quad (11)$$

where N is the number of training points and N_{atoms} is the number of atoms in a single training data point. The prefactor, σ, can be determined analytically for fixed values of $\sigma_{n,E}/\sigma$ and $\sigma_{n,f}/\sigma$, so the numerical optimization problem is only one-dimensional.

The Gaussian process allows for the specification of a prior function, $E_p(\mathbf{x})$, for the energy landscape. In equation (1), the prior energy landscape is inserted as $\mu_p(\mathbf{x}) = (E_p(\mathbf{x}), \nabla E_p(\mathbf{x}))$. Here, we apply the prior function suggested by Bisbo and Hammer [13], which is a repulsive potential of the form

$$E_p(\mathbf{x}) = E_c + E_r(\mathbf{x}) = E_c + \sum_{ij}\left(0.7\,\frac{R_i+R_j}{r_{ij}(\mathbf{x})}\right)^{12}, \quad (12)$$

where E_c is a constant, R_i and R_j are the covalent radii of atoms with indices i and j, and $r_{ij}(\mathbf{x})$ is the distance between the atoms in the set of atomic coordinates $\mathbf{x}$. The prior energy function expresses the expectation that the energy rises steeply if two atoms come very close. As we shall see later, this helps avoiding very high energy structures in the training data.

The constant value E_c is determined by maximizing the marginal likelihood, and is given by the analytic formula

$$E_c = \frac{\mathbf{U}^T C(P,P)^{-1}(y - E_r(X))}{\mathbf{U}^T C(P,P)^{-1}\mathbf{U}} \tag{13}$$

where $E_r(X)$ is a vector that consists of repulsive priors of the training data, that is obtained using the second term in equation (12), and $\mathbf{U}$ is a vector of length $N(3N_{\text{atoms}} + 1)$ with elements

$$\mathbf{U}_i = \begin{cases} 1, & \text{if} \quad i \bmod (3N_{\text{atoms}} + 1) = 0 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where indexing of i starts from 0. Therefore, $\mathbf{U}_i = 1$ if y_i is an energy value, and $\mathbf{U}_i = 0$ if y_i is a force.

III. MODEL VALIDATION

A. Learning curves and cross validation

To validate the model, we examine two different systems: A Cu_{15} cluster using an effective-medium theory (EMT) potential as implemented in the ASE package [37–39] and bulk SiO_2 using DFT with the PBE functional [40], implemented in GPAW [41]. The unit cell for SiO_2 consists of 12 atoms and has the lowest energy in the cristobalite structure, comprising tetrahedral SiO_4 units. For both training and validation sets, random structures are generated and relaxed loosely so that the maximum force criterion is 10 eV/Å. (The random structures are generated in the same way as in the global optimization runs, which is discussed in detail below, in section IV.) Different sizes of training sets are then used to generate learning curves for models with a variety of length scales in the squared-exponential kernel. We limit the training set size to 100 because with larger sizes there is a risk of memory problems when gradients are trained. The validation set size is kept as 100.

The reason for showing the learning curves for different length scales is that the length scale has, as we shall see, a dominant effect on the predictive power of the model, and we also observe that the optimal length scales do not necessarily follow the traditional expectations for a Gaussian process. The constant term in the prior function (Equation (12)) is set to the mean energy of the training set and the kernel prefactor is kept constant since it does not affect the mean of the predictions, as deduced from equation (1). In addition to fixed length scales, the learning curves are computed for models where the length scale, the prior constant and the kernel prefactor are obtained by maximizing the marginal likelihood separately for each training set size.

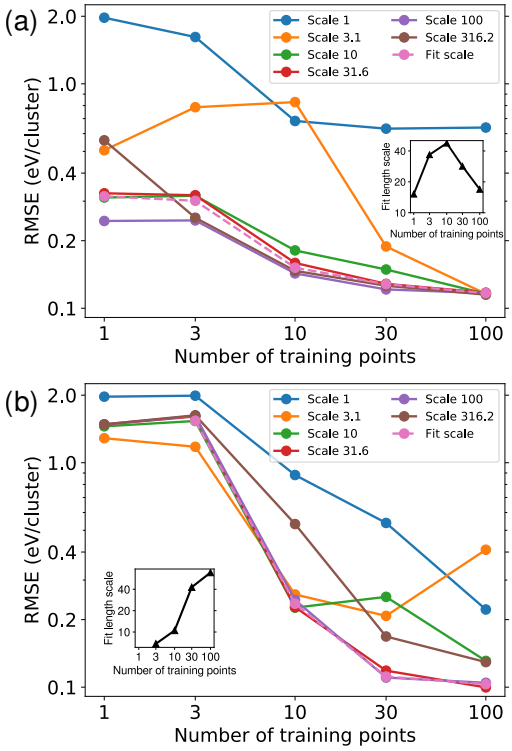


FIG. 1. Cu_{15} learning curves obtained by (a) training on both energies and forces and (b) training on energies alone. The inset graphs show the values of the fit length scales for each training set size.

The learning curves for Cu_{15} are shown in Fig. 1 both with and without training the gradients. For the gradient-trained curves, we observe that the root-mean-square error (RMSE) saturates to approximately 0.12 eV/cluster with length scales greater than 30 when the training set size is increased up to 100 data points. The curves with scale 3.1 and 10 do not seem to saturate to the same degree but they end up at the same RMSE at 100 training points as the other models. The standard deviation of the test set energies is 1.5 eV/cluster, meaning that our model can decrease the RMSE to about 7% of what random sampling of energies would produce. Amazingly, a RMSE of 0.25 eV/cluster is achieved by having only one data point in the training set. This might be due to the ability of the fingerprint to catch the relatively simple radial dependence of the EMT potential, together with the smooth squared-exponential kernel.

Despite the good start with few training points, the curves develop quite slowly with increasing number of training data. The observed saturation is in contrast with the power law behavior that the learning curves should optimally follow with linear learning curves on the log-log scale [42, 43]. One reason for the saturation could be the difficulty to resolve differences between some configurations as compared to others [44]. We have observed that small perturbations of the atomic structure that lead to similar variations in the energy may differ in several orders of magnitude in the variation of the fingerprint. This may indicate that the potential energy surface is highly anisotropic in fingerprint space. We note that the high dimensionality of the fingerprint also makes correcting for the anisotropy with different scales in each direction impractical.

Another explanation for the shape of the learning curves could be that the distribution functions used in the fingerprint have a finite range. Deringer and Csányi have shown in the case of amorphous carbon [45] that a finite cutoff may lead to substantial residual forces for a per-atom fingerprint. We have a global fingerprint, but still the finite range may limit the quality of the model prediction.

In practice, it seems that the gradient-trained model learns everything it can after training of the order of only 30 data points. On the other hand, an accuracy of 0.12 eV/cluster is clearly sufficient to be of relevance for the determination of the basins with low energy configurations.

The effect of training the gradients is apparent in Fig. 1: the prediction error is clearly lower up to training set sizes of the order of 10. At 30 training data points and above, the energy-trained model seems to do as well as the gradient-trained one. With the largest training set sizes, the energy-trained model becomes even slightly better than the one including the gradients. The difference is small, though, with the RMSE difference being only 0.012 eV/cluster at training set size of 100 between the models with the best length scales. However, given the saturation of the gradient-trained curves, it is natural to expect that the energy-trained model reaches the performance of the gradient-model at some point in general, and for this particular system that happens after around 30 training points.

The models with scales 10 and above saturate to similar performance when the training set size of 100 is reached. All of these scales are comparable to or longer than the distances between the data points in the data set with 100 points. For this data set, the distances vary between 0.1 and 2.3 in fingerprint space. The model with scale 3.1 is seen to perform less well for small data sets, but is not saturated when 100 data points is reached. The limited distances in fingerprint space has to do with the character of the fingerprint. If the atoms in a given configuration are displaced, the Cartesian distance corresponding to the displacement can grow indefinitely. In fingerprint space the distances are calculated based on differences in distribution functions, and these will saturate at some point.

The length scales obtained by maximizing the log-likelihood are always above 15 when gradients are included in the training, and above 6.8 while training on the energies alone (see Fig. 1). These scales are also surprisingly long considering the distances in the training set. There is a clear increasing trend of the optimal length scale in the energy-trained models when the training set size is increasing, but no trend is observed within the gradient-trained models. Nevertheless, maximizing the log likelihood gives roughly the best model as evaluated with cross validation.

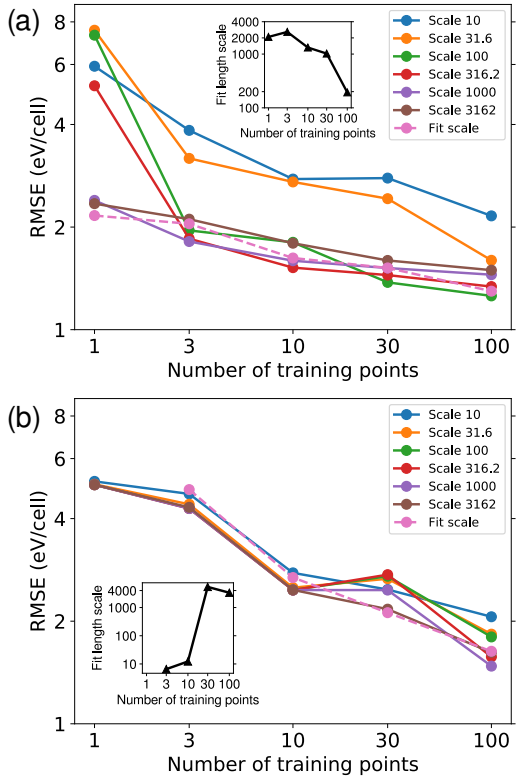


FIG. 2. SiO_2 learning curves (a) training energies and forces and (b) training only energies. The inset graphs show the values of the length scales obtained by maximizing the log-likelihood for each training set size.

For bulk SiO_2 (figure 2), the picture is different in that the power law decay of the learning curves is roughly maintained with most models with different length scales up to 100 training points. But, the prediction power is not too impressive with the best RMSE of 1.27 eV/cell at 100 training points although it is lower than the standard deviation of the validation set, which is 3.6 eV/cell.

The energy-trained model reaches towards the gradient-trained one again, but at 100 training points the gradient-trained model predicts still slightly better than the model based on energies alone. No clear saturation is observed within this data range with neither of the approaches.

Compared to the EMT cluster, the learning ability with a single training point is reduced. This is expected when moving to a more complex potential energy surface: the quantum effects of a more realistic potential are difficult to track with a relatively simple fingerprint in contrast to the simple radial dependency of EMT.

From the SiO_2 learning curves we can also see that the length scales above 100 are clearly favored over the smaller ones. Again, this is remarkable for a Gaussian process with a single squared-exponential kernel as covariance function, since the distances between the test data points vary between 1.6 and 21.3, that is one or two orders of magnitude smaller than the most optimal length scales. According to Fig. 2, the most optimal length scale is as much as 1000 while training only on energies, even further from the deviation of the data point distances compared to the gradient-trained model.

The fitted length scales, obtained by maximizing the log-likelihood and shown in the insets in Fig. 2, have a clear decreasing trend for the models trained using gradients. For the models trained on only energies the variation is dominated by a large jump between 10 and 30 training points. This behaviour indicates a high sensitivity of the model to the addition of more data, since we are dealing with a small numbers of training data points. Another reason might be that there are multiple local maxima in the marginal likelihood, and different maximization runs end up at different local maxima.

Finally, it should be noted that for both Cu_{15} and SiO_2, fitting the length scale by maximizing the marginal likelihood gives roughly the best model in the cross validation. This is a desired behavior since maximizing the marginal likelihood is an easy and computationally relatively cheap procedure to carry out, and it can therefore be done repeatedly during a global search where the training set is updated often.

B. Local relaxations in the surrogate model

Since our global optimization method relies on local relaxations rather than single-point calculations in the surrogate PES, it is relevant to examine how well the model performs local relaxations. We create a training data set of 40 data points of the Cu_{15} cluster, relaxed with EMT so that the maximum force residual is less than 1.0 eV/Å. After this, the model is trained on the data both with and without gradients. Then, 80 random structures are created independently and relaxed locally in the surrogate model. We note that the minimizations on the surrogate surface are always performed using the predicted forces. This can be done even if the model is trained on energies only, as we note in section II B.

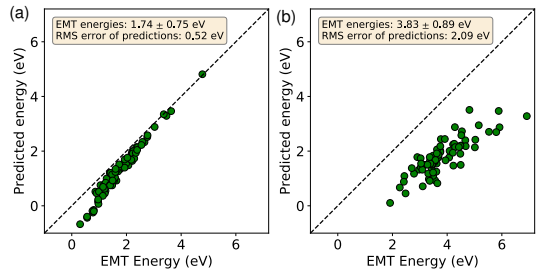


FIG. 3. Predicted versus true energies of the final structures of relaxations with surrogate models. 80 local relaxations of Cu_{15} are run with surrogate models with (a) training on the gradients and (b) training only on energies. The training set of 40 data points is the same for both models. The total energies of the training data points have distribution with mean of 6.0 eV and standard deviation of 1.14 eV. The reference point 0.0 eV is set to the global minimum energy. Statistics of the energies of the final structures, as well as the statistical prediction errors, are shown in the text boxes. The dashed line shows the ideal 1-to-1 mapping of the predictions.

A comparison between the EMT energies and the energies of the models is shown in Fig. 3. The EMT total energies of the obtained structures range from 0.38 to 4.8 eV with the gradient-trained model, and from 2.0 to 6.9 eV with training only on energies. Actually, the lowest energy structure (of 0.38 eV) corresponds to a structure that is very close to the true global minimum structure. The prediction errors range from -1.0 to 0.0 eV/cluster with gradients and -3.6 to -1.2 eV/cluster without gradients. The data demonstrates that the gradient model is able to reach both lower energies and higher accuracy than the model including only energies, although we work at a training set size of 40 where the learning curves show similar performance to each other. The model trained on gradients exhibits some systematic errors in particular for small energies, which is most probably due to these structures being relatively far from the training data, making the prediction of their energies more difficult. However, the ordering of the energies seems to be well reproduced. The errors are much larger for the model trained on energies alone, but again the ordering of the states is reproduced fairly well.

IV. GLOBAL OPTIMIZATION ALGORITHM

The algorithm for the global optimization is relatively simple: in each iterative step, multiple local relaxations are carried out on the surrogate surface, and the energy and forces for the most promising structure are evaluated using the true potential (EMT or DFT). To select the most promising configuration of all the relaxed structures, we make use of the estimated uncertainties by cal-

culating the acquisition function

$$f(\mathbf{x}) = \mu(\mathbf{x}) - \kappa\Sigma(\mathbf{x}) \tag{15}$$

for each structure, where we set the parameter $\kappa = 2$. This form of the acquisition function for minimization problems is called lower confidence bound in the literature, and the choice of $\kappa = 2$ provides a good balance between exploitation (low energy) and exploration (large uncertainty) [13, 33, 46]. The structure with the lowest acquisition function is selected for evaluation with the true potential. This structure is then added to the training set for the Gaussian process and another set of surrogate relaxations are performed with the updated model, as illustrated in Fig. 4. The effect of using the fingerprint is visualized in Fig. 4 as well: after just a single training point, the predicted PES exhibits several important features like the existence of the two local minima. This would not be the case if using Cartesian coordinates as the descriptor due to the missing permutational, rotational and translational symmetries. The gradients also play an important role for quick learning of the features of the PES. Moreover, adding the second training point in one of the basins makes the prediction in the second basin more accurate.

The initial training set consists of randomly generated structures with energies and forces evaluated. In this work, the optimization routine is always started with 2 initial training points. The starting structures for surrogate relaxations comprise three different types: 1) Already visited structures with the lowest energies, 2) already visited structures with random displacement (also called rattling), and 3) randomly generated structures. The total number of surrogate relaxations per step in this work varies between 20 and 40, depending on the system but kept constant during a run. About 25% of the relaxations start from structures of type 1, 25% of type 2 and 50% of type 3. Before accepting the structure given by a surrogate relaxation, it is checked that none of the bond lengths in the system are less than 0.7 times the covalent distance of the atoms. If no valid structures are acquired in an iterative step, a random structure is generated, evaluated and added to the training set, to achieve a model that will fail with smaller probability in the next iteration.

For clusters, the random structures are generated as follows. First, one of the atoms is placed at the origin. After this, the position of the next atom is always given by $\mathbf{r}_1 = \mathbf{r}_{\text{rand}} + (r, \theta, \phi)$ in spherical coordinates where $\mathbf{r}_{\text{rand}}$ is the position of one of the previously set atoms, randomly selected, and (r, θ, ϕ) are randomly generated spherical coordinates. Here the sample distribution of r is selected manually and system-specifically, but it was observed that selecting the upper bound of r slightly smaller than the standard covalent distance of the two atoms works best in general. After generating a new position, we make sure that adding an atom in the acquired coordinates does not violate our restriction of the short bond lengths with each atom added to the cluster

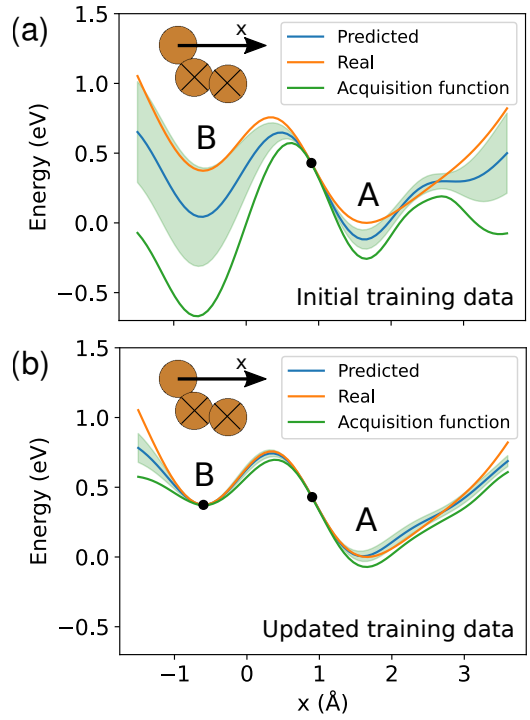


FIG. 4. One-dimensional demonstration of surrogate surface. In the figures, the true potential surface, evaluated with EMT, is shown for a system where one Cu atom is moved on top of two other Cu atoms. The two basins of the potential energy surface are labelled as A and B, where the bottom of A has lower energy than B. In (a), a data point is evaluated at $x = 0.96$, and a Gaussian process is trained with that single training point. The resulting predicted surface, uncertainty (green area) and acquisition function are shown. The acquisition function is minimized in basin B, at $x = -0.74$, and in (b), we show the predictions after evaluating and adding this point to the training set. Now, the acquisition function is minimized in A, the global minimum basin of the true potential energy surface.

already, as described above. For TaO clusters, we set another restriction to add some chemical intuition: when O is being added to the system, we enforce that $\mathbf{r}_{\text{rand}}$ is the coordinate of a Ta atom and not another O atom. This way we avoid introducing chains and clusters of oxygen in the randomly generated structures.

For bulk systems, the atoms are simply put at random coordinates inside the unit cell, and then relaxed in a repulsive potential of the form of

$$V_{\text{rep}}(\mathbf{x}) = \sum_{ij} \left(0.4\, \frac{R_i + R_j}{r_{ij}(\mathbf{x})}\right)^{12} \tag{16}$$

with similar notation to that of the prior in equation 12. For surfaces, the atoms are put inside a manually-defined box inside the true unit cell, where the atoms are placed in a similar fashion to that of the bulk systems and then relaxed in the repulsive potential, given by eq. (16).

The hyperparameters E_c, σ, ℓ are updated during the global optimization assuming fixed values of the noise parameters relative to the prefactor as explained above. The prior constant E_c and the prefactor σ are obtained analytically and are updated at every step of the global optimization. The update of the length scale has to be done numerically and is performed every five steps with an initial value of 20 times the distance in fingerprint space between the two initial structures. It is our experience that the length scale obtained from maximizing the log-likelihood may be too short leading to a too rapidly varying surrogate PES with more local minima than the true PES. This affects the search so that it becomes too local. This is unfortunate, especially in the beginning of the search, where large parts of the configuration space has to be explored. We therefore introduce a lower bound on the value of the length scale during update. The lower bound is set to the mean value of all distances between the training data points in fingerprint space. Using this value ensures that a large number of training data are used in each energy/force prediction and too local searches are avoided. We also note that the investigations of the learning curves above indicate that a length scale considerably longer than the one obtained from maximizing the log-likelihood still results in a reasonable model. In some cases the prediction error is in fact reduced by increasing the length scale.

The algorithm is implemented so that the user can choose whether to train using the gradients or not. In both cases, the energy and force predictions can be obtained analytically from the surrogate potential energy surface. The approach where the gradients are not included in the training has a reduced memory usage and the time to train the model is also significantly reduced. However, as we have seen in the investigations of the learning curves the models without training on gradients are less accurate. In the GOFEE method [13], training is only performed on the energies, but another training point, adjacent to the one selected by the acquisition function, is always evaluated with the true potential and added to the training set. The second training point is obtained by moving the atoms a small distance along the direction of the forces. Adding the neighboring data point allows the GP model to have the information about the amplitude of the gradient in the PES in one direction, presumably leading to more accurate predictions.

In this paper, we refer to our approach with training on both energies an forces as BEACON (from Bayesian Exploration of Atomic Configurations for OptimizatioN) and the approach where the training is only on energies as L-BEACON, where L stands for "light". Although the forces are not trained in L-BEACON, we can still predict the forces in the system (as noted in section II B) to be used in the relaxations. We also show results of L-BEACON-exact where a neighboring data point is evaluated and added to the training set similarly to GOFEE, and L-BEACON-FD, where, for each DFT-evaluated data point, we add a neighboring data point where the energy is obtained by a finite difference estimation based on the DFT forces. The step length in the finite difference method is discussed in section V. We will see that for the systems we investigate here, adding extra displaced training points does not lead to significant improvement even if gradients are not trained. Furthermore, we find that training only the energies of single points (L-BEACON) makes a surrogate potential energy surface that has similar or almost similar performance in global optimization, compared to L-BEACON-FD and L-BEACON-exact.

As usual in global optimization, Bayesian optimization gives no information whether the true global minimum is achieved, unless the full search space of interest is explored. In BEACON, the search is continued until a given number of DFT calculations is performed, or if time or memory resources are run out. Therefore, the best indication of whether the true global minimum is found is that several separate BEACON runs end up with the same lowest-energy structure.

We also note that the algorithm of BEACON does not include any geometry relaxations on the true PES, but all the relaxations are done on the surrogate PES. In this work, DFT relaxations are only performed if explicitly stated.

V. RESULTS AND DISCUSSION

A. Cu_{15}

In Fig. 5, we show the success curves of different types of global optimization runs for the Cu_{15} cluster in the EMT potential. We perform 40 separate runs with BEACON, L-BEACON, L-BEACON-FD and L-BEACON-exact with different lengths of displacements, where neighboring training points are included, and 16 separate runs with GOFEE [13]. In the figure, the cumulative curves increase by a step each time a single run finds the global minimum with an energy threshold of 0.01 eV/cluster. The threshold value means that we declare a run successful once it hits a true energy that is at maximum 0.01 eV higher than the lowest energy that was found during the runs. The reference energy here corresponds to the geometry of a centered icosahedron of 13 atoms and two adsorbed atoms in neighboring hollow sites of the icosahedron surface, as shown in the inset of Fig. 5. The second lowest-lying local minimum was found at 0.15 eV/cluster above the global minimum, possessing a centered, gyroelongated hexagonal bipyramid. This observation illustrates the ability of the global optimization approach to distinguish between local minima that are close in energy.

Let us first discuss the three most optimal success curves: BEACON, L-BEACON, and L-BEACON-FD with step length dx=0.001 Å. The ability of the gradient-trained model to simulate the potential energy surface of EMT is manifested again as 50 % of the BEACON runs found the true global minimum after only 7 EMT evaluations. The convergence of 7 evaluations would be appealing even for local optimization with the 45 degrees of freedom in the system, although we remind ourselves that convergence here is defined via energy whereas in the context of local relaxation convergence is determined through stricter requirements on the forces in the system. For L-BEACON-FD, the respective number of 50 % success is 16 evaluations and for L-BEACON where no force information is used, 50 % success is acquired after 20 evaluations. Our runs with GOFEE [13] show that 46 evaluations are required for 50 % success. It is worth noting that despite the fast success, the program does not know that it has reached the optimal configuration but keeps searching even after the (known) global minimum is found.

Let us compare our results with random searches, that are shown to be surprisingly efficient when certain chemical intuition is considered [7]. We run 480 relaxations with the true calculator, EMT, starting from similarly generated random structures as those for the global optimization. The result is that 60, or 12.5 %, out of all relaxations end up in the global minimum energy structure. If we perform one such EMT relaxation per step, this would statistically result in 61 % success at step 7, since $\sum_{i=1}^{7}(1-1/8)^{i-1} \times 1/8 = 0.61$. The important difference is that a single relaxation takes 20-200 EMT calculations, whereas all of our relaxations are performed within the surrogate model and not with the true potential. From this perspective, we conclude again that the model and the global optimization approach of BEACON are together very efficient in the search for the global minimum. Running EMT relaxations is in fact computationally faster than running the relaxations in the surrogate surface within the global search algorithm, as we will discuss further in section VI, but when using the algorithm with DFT this situation is of course completely different.

BEACON is seen to be the fastest method up to 80 % success rate. Most of the L-BEACON-FD (with step length dx=0.001 Å) runs find the minimum with between 10 and 20 EMT calculations. L-BEACON lags only a little behind, and finally all the three approaches end up with a somewhat similar performance, although the full success of BEACON curve takes slightly more steps with one run finding the correct local minimum after 48 EMT evaluations.

Let us now take a closer look to the different approaches of L-BEACON where the neighboring data points are included into the model, i.e. L-BEACON-FD and L-BEACON-exact. First of all, the success curves with the EMT evaluated neighboring points, that is the L-BEACON-exact curves, are always behind the curve for L-BEACON where no neighboring points are included

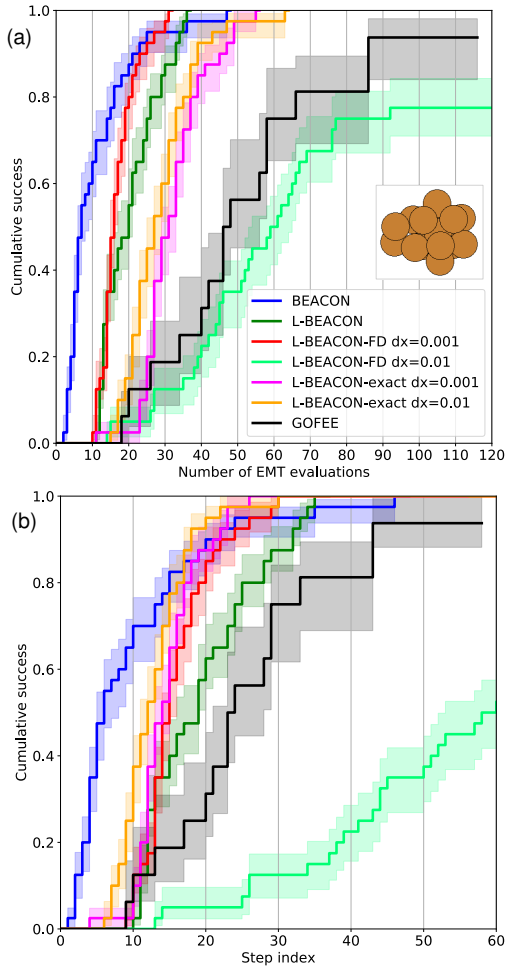


FIG. 5. Cu_{15} success curves with success threshold of 0.01 eV/cluster. The colored areas denote the standard deviation of bootstrap simulations with 1000 samples of each curve. (a) Success curves as function of number of EMT calculations. (b) Success curves as function of step index. Here, step means an iteration of the train-search-select-evaluate cycle. FD refers to adding a neighboring point per each EMT-evaluated data point, where the energy of the second point is estimated with finite-difference method with step length dx (Å), based on the energy and forces of the EMT point. The notation "exact" refers to evaluating the energy of the second point using EMT.

in the model. With arbitrary increase of step length in L-BEACON-exact, we expect the success curves to reach the performance of L-BEACON at best, since in that case all the data points are more or less individual and the neighboring points do not represent simulating the forces anymore. Also, the step length of dx=0.001 Å leads to

slightly slower performance than that of dx=0.01 Å, indicating that the step length of the size 0.001 Å is too small to have the desired effect on the model; due to the noise term in the model, the two neighboring points cannot be distinguished properly when they are too close to each other. However, comparing the red and pink curves in Fig. 5b, the smaller step length seems to work similarly between L-BEACON-exact and L-BEACON-FD despite the approximation. This indicates that the linear approximation is accurate enough to produce a reasonable surrogate PES. On the other hand, the step length 0.01 Å is clearly too large for a stable model in L-BEACON-FD although it results in better performance with L-BEACON-exact. The value of dx=0.001 Å shows the fastest global optimization of all the double-point approaches in numbers of EMT calculations, but the performance is not much better than with L-BEACON. Also, the problem of selecting a suitable step length might be difficult for other systems and more complex potentials. Thus, we do not find inclusion of neighboring points in general beneficial. We will return to this topic later in the case of bulk SiO_2.

Comparing our curves with GOFEE, we see that the number of EMT calculations is two-fold or more for GOFEE. As mentioned above, the essential point of interest in the methods is the way in which the energy of the neighboring data point is evaluated. In this respect, L-BEACON-exact is similar to GOFEE. It is observed that L-BEACON-exact is faster in finding the true global minimum, which we attribute to the slight differences in the details of the fingerprint, the kernel function and fitting the hyperparameters along the way. Comparing BEACON and GOFEE, there is about a factor of 6 difference between the number of required energy evaluations.

Finally, let us connect the result with BEACON to the examination of local relaxations above (section III B). There, the training set size was 40, and even with 80 relaxations the global minimum was not found. In the global search we find the minimum after only 7 training points. This indicates that updating the model along the run is an important feature of the global optimization algorithm.

B. SiO_2

Figure 6 shows the success curves for bulk SiO_2 in the low-cristobalite phase with DFT/PBE. In this case, training the gradients is clearly favourable in the search for the global minimum structure. BEACON finds the correct structure after less than 34 DFT evaluations for all runs. L-BEACON and L-BEACON-FD are slower than BEACON and eventually fail in 2/20 runs to find the true global minimum using 80 DFT calls.

We saw that for the Cu_{15} cluster the search identified the global minimum based on very few EMT energy and force evaluations compared to what could be expected based on the learning curves. This feature is even more pronounced for the SiO_2 system. The learning curves indicate a rather poor accuracy with errors of more than 1 eV/cell (for the 12 atom system) in the range all the way up to 100 training points, but still the global minimum is found within 0.05 eV/cell using only of the order 25 DFT calculations. The explanation for this behavior must have to do with the fact that, in the global search, states around local minima of the PES are of preferential interest and included in the training set. The model is thus exclusively trained to predict a special part of the PES. In the cross validation studies above, the model is trained on a wider range of points and also evaluated broadly in configuration space.

This idea is to some extent illustrated with the simple one-dimensional potential energy surface of the Cu_3 cluster discussed above in Fig. 4. The evaluation of the model at the minimum point of the well B considerably improves the prediction at the other minimum A, because some of the local bonding characteristics are the same.

Interestingly, the energy-trained models, L-BEACON and L-BEACON-FD, have rather similar performance in the global optimization although the training set of L-BEACON-FD includes twice the number of data points. The difference to the Cu_{15} is the more complex true potential energy surface, introducing more error in the finite-difference method. For the case of SiO_2, we assert that a step length of 0.001 Å is too small for the energies of the neighboring points to be distinguishable by the Gaussian process, resulting in failure of simulating the slopes of the PES. Increasing the step length increases the risk of running into problems with an unstable Gaussian process, as observed with Cu_{15}. We thus conclude, somewhat at variance with Bisbo and Hammer [33], that adding neighboring data points, as done in L-BEACON-FD, L-BEACON-exact, or GOFEE, is not beneficial in general for the Bayesian approach to global optimization.

Let us now investigate how the success curves depend on the threshold value for the energy. In Fig. 6 an energy threshold of 0.05 eV/cell is used for SiO_2, and we show the success curves with energy thresholds of 0.2 eV/cell and 0.01 eV/cell in the Supplemental material [36]. We see that BEACON is more efficient than the L-BEACON methods, which are quite similar to each other, and therefore the overall analysis is not too sensitive to the choice of the threshold value. However, the choice of an appropriate threshold may depend on the system being investigated. For example, an energy threshold of 0.2 eV/cell for the Cu_{15} cluster has the consequence that finding the second lowest local minimum is also counted as a successful run. For SiO_2, we did not determine the second most stable structure explicitly, but no other stable structure was found within the highest threshold of 0.2 eV/cell. On the other hand, small thresholds like that of 0.01 eV/cell might fall below the accuracy of the DFT implementation, parameters in use (such as k-point density), convergence criteria of the self-consistent cycle etc., making the judgment whether the global minimum was found or not. The threshold of 0.05 eV/cell seems to

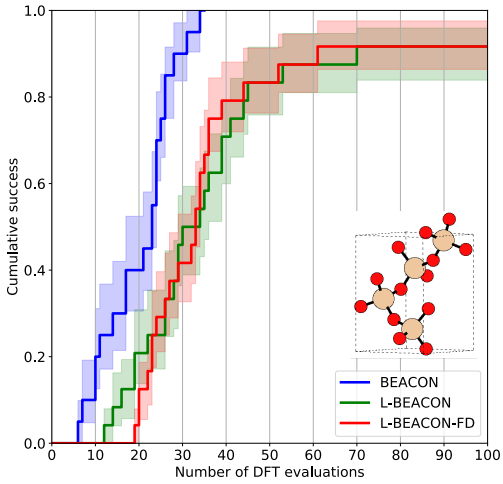


FIG. 6. SiO_2 success curves with success threshold of 0.05 eV/cell. The unit cell includes 12 atoms. The global minimum structure is shown in the inset, where 5 extra O atoms are added to the unit cell to illustrate the tetrahedral coordination of Si. For L-BEACON-FD, the step length is 0.001 Å in the finite difference method.

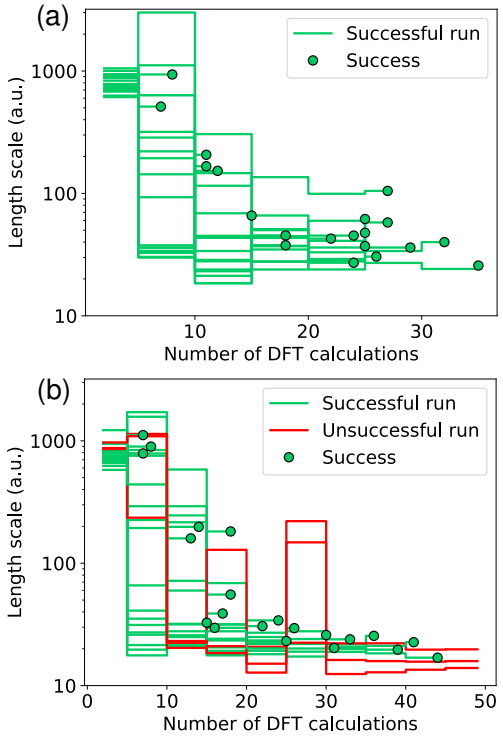


FIG. 7. Evolution of the updated length scales of the Gaussian process kernel during the runs of SiO_2 in (a) BEACON and (b) BEACON without lower bound for the updated length scales.

be a good compromise for global optimization of systems with 10-20 atoms using DFT calculators like GPAW with default settings. In this work, we use the threshold value of 0.05 eV/cell for all systems studied with DFT.

The length scale is updated every 5 steps by maximizing the marginal log-likelihood as discussed above in the section IV. Furthermore, as also discussed above, the length scale is bounded from below by the mean value of all distances between the training data points in fingerprint space. In Fig. 7, we show the evolution of the fitted length scale in the global optimization runs of SiO_2 when constraining the fitted length scales from below and when not. For both cases, we observe that after the first fitting at 5 DFT calculations there is a huge variance in the optimal length scales over the different runs: the values range from 30 to 3000. As the searches proceed, this range gets narrower, and after 20 DFT calculations the updated scales vary only between 25 and 100 when constraints are turned on. The corresponding figure for L-BEACON runs is shown in Supplemental material [36]. For L-BEACON, most of the values lie below 100, but the updating also converges to higher values along the search. This does not seem to be a problem though, as global minima are also found with the large scales.

For BEACON, the length scales where the correct global minimum structure is found are gradually decreasing as the search proceeds. Furthermore, it can be seen that if the length scale is below 100, the global minimum is never found in fewer than 15 steps. Apparently, the length scales obtained by maximizing the marginal log-likelihood are not necessarily optimal in the early part of the global search, where exploration is particularly important. The large variation in the length scales in the beginning of the search is hardly surprising in light of the small number of training points. In this perspective, it seems reasonable to limit the acceptable length scales from below to prevent strong overfitting in the beginning of the search. It appears from the data in Fig. 7 that the lower bound could be set even higher in the beginning of the runs.

Without the lower bound on the length scale, three of the runs fail to find the global minimum in 50 DFT calculations as shown in Fig. 7b. It is clear that the failing runs after 20 steps have very short length scales and even though the length scale is increased after 25 steps they fall back to a less exploratory mode, where the global minimum is not identified.

C. $[Ta_2O_5]_x$

Ta_2O_5 is an optically interesting material, whose crystal structure is still under debate [47–49]. Furthermore, clusters of the material might be of interest in photocatalysis [50, 51].

To test our approach, we investigate small clusters with the composition $[Ta_2O_5]_x$ with $x = 1, 2, 3$ corresponding to the stoichiometry of the bulk material. The structures that we find to minimize the potential energy are shown in Fig. 8. All the globally optimal structures for clusters are such that O is sticking out of the TaO core, making it impossible to use these units as building blocks for a bulk system while preserving the correct stoichiometry Ta_2O_5. Nevertheless, it is interesting that in all the structures every Ta atom has a similar bonding environment to four O atoms, one of which is pointing outwards from the cluster. Interpreting this as a double bond makes the oxidation number of each Ta atom to be +5.

In any case, the global optimization of the clusters provides a good demonstration of our method. Of course there is no rigorous proof that the obtained structures are in fact the global minimum energy structures, but some indications of this are obtained by just repeating the searches and noting the variety of structures visited during the search. The shown structures were found in 4/4 runs for Ta_2O_5, 5/6 runs for $[Ta_2O_5]_2$, and 4/8 runs for $[Ta_2O_5]_3$. The second lowest local minimum for Ta_2O_5 was observed at 0.16 eV/cluster higher than the lowest one, and this was visited in all runs. This again indicates that the model and the acquisition function are able to identify small energy differences between different structures.

The average number of required single-point DFT calculations before hitting the global minimum structure (with threshold of 0.05 eV/cluster) was 42 for Ta_2O_5, 40 for $[Ta_2O_5]_2$, and 35 for $[Ta_2O_5]_3$, calculated among the successful runs. The small number of steps required is a highly desired result, because it means that one can limit the length of the BEACON runs. This does not only have the advantage that the total number of DFT calculations is small, but long runs of BEACON with many steps lead to surrogate models with many data points, which require more memory and computational time. It is interesting that the average number of DFT calculations is smallest for the largest cluster where the number of degrees of freedom is the largest. This could be explained by the fact that whenever we train the model with a larger cluster, more training data is available since the number of trained gradients is larger. However, we note that in general larger systems can be expected to exhibit considerably more local minima to be explored making the global search more difficult.

We now take a closer look at some of the BEACON runs in order to get a better understanding of how the algorithm behaves. We first investigate why some of the runs fail for the larger clusters by checking how the global minimum structure is predicted with surrogate models of the unsuccessful runs. That is, we use the surrogate models at step number 40 of the global optimization runs (42 training points), and perform a local relaxation on this surrogate potential energy surface starting from the global minimum structure. In every case, the relaxation does not change the structure significantly, and the acquisition function of the relaxed structure is low enough so that it would have been selected for DFT evaluation in the original BEACON run. We thus conclude that for these runs, the search within the surrogate space is insufficient whereas the accuracy of the model is good enough to find the global minimum.

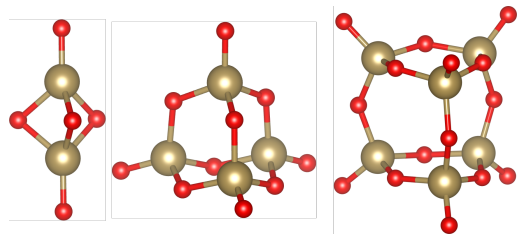


FIG. 8. Global minimum structures of Ta_2O_5 (D_{3h} symmetry), Ta_4O_{10} (T_d) and Ta_6O_{15} (D_{3h}) clusters as found by BEACON.

In Fig. 9, we show how a single run of Ta_6O_{15} proceeds along the search. The search starts with high-energy structures where the prediction and the true energy do not match at all, but as the high uncertainties show, the model knows it might be wrong about the predictions. As the search continues, different structures are suggested and evaluated with a good agreement between the predictions and the true energies, taking the estimated uncertainties into account. The global minimum structure is found at step 42 in this particular run, and the exploration continues after that, as indicated by the higher-energy structures that are visited between step 42 and step 68. As noted in the description of the algorithm, we remove structures, which have been obtained by local relaxations in the surrogate model, if a bond distance is smaller than 0.7 times the sum of the covalent radii of the atoms in the bond. What happens at step 68 in this BEACON run is that the model begins to develop local minima with O-O bond lengths, which are just above this threshold. So the new predicted structures contain unphysically short O-O bonds or even clustering oxygens linked to the rest of the cluster. The model predicts their energies to be very low, and they are therefore always selected by the acquisition function (eq. 15). However, after step 80 we see that the model has learned that the unphysical structures have high energies, and the search continues in a more reasonable way both exploring new areas and exploiting the known structures, the global minimum included.

This issue illustrates how additional conditions on the search might be helpful to avoid unphysical structures,

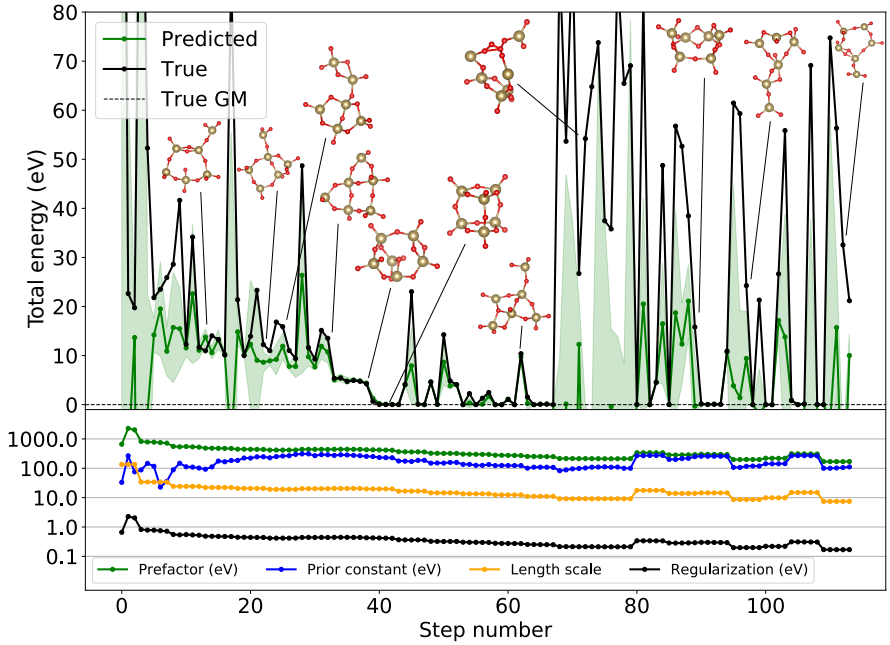


FIG. 9. Energy, prediction, and hyperparameter evolution in a single global optimization run for a Ta_6O_{15} cluster. Also, selected structures are shown that were evaluated with DFT along the run. The global minimum structure is visited at step number 42, and at various steps after that. The green area shows the estimated uncertainty of the prediction. The prior constant is shown with respect to the global minimum energy.

but also that the surrogate model might react to these in unexpected ways.

Let us briefly look at how the hyperparameters evolve as shown in Fig. 9. To first recapitulate, we have four hyperparameters: the length scale in the kernel, the kernel prefactor, the prior constant, and the noise parameter. For simplicity, we keep the ratio between the noise and the prefactor fixed, because the maximization of the log-likelihood can then be done analytically with respect to both the prefactor and the prior constant. Only the length scale has to be obtained numerically and this update is done every 5 steps. We see in Fig. 9 that the prior constant has some variation in the beginning but the changes become smoother as more training data is acquired. The dramatic changes in the model at step 68 also lead to a recalibration of the prior constant. It should be noted that the prior constant is not only a weighted mean of the observed energies, because the gradients also play a role in the determination as seen from Equation (13).

The length scale is significantly reduced during the run, and we take this as an indication that the model is initially focusing on getting the large-scale features of the PES correct with subsequent refinements. This is an appropriate behavior for a global search strategy that we already examined in the case of SiO_2.

The kernel prefactor also gradually decreases during the global optimization with the most significant changes every 5 steps when the length scale is updated. The prefactor is unimportant for the prediction of energies and forces as can be seen from Equation (1), but it plays a major role for the estimated uncertainties. The decaying value might therefore indicate an increasing confidence of the model. The reduction of the prefactor can also be seen as coupled to the change in the length scale. The uncertainty at a particular point in fingerprint space is roughly given by the kernel function to neighboring data points. This estimate involves both the prefactor of the kernel function and the distances to the neighboring data points measured in units of the length scale. A reduction of the length scale thus leads to a less "stiff" model with larger variances and this is to some extent compensated by the reduced prefactor. The regularization follows exactly the variation of the kernel prefactor because the quotient of these quantities is kept constant throughout the run.

In the Supplemental material [36], we show another, similar run to that in Fig. 9. Although the global minimum energy structure is not found, the overall behaviour of the predictions and hyperparameters is similar with a good balance between exploration and exploitation. Even the stage of unphysical structures with short O-O

bonds is the same after 60 steps, after which the search becomes stable again at around step 80. Although the prior constant goes below the global minimum energy in the beginning, we note that it does not have a negative effect on the model or the search: in the beginning the length scale is relatively long, and therefore all the points in the (reasonable) coordinate space are considered close to each other compared to the length scale, and consequently the model is not very sensitive to the absolute value of the prior.

An important feature observed in both runs is that the estimated uncertainties are reasonable so that when the error of the prediction is large, the model knows that it might be wrong. This is an essential property for using the lower confidence bound as the acquisition function, Equation 15, to control the balance between exploration and exploitation in the global search.

D. ZrN-O surface

As the last demonstration, we briefly illustrate the applicability of the approach to surface structures. Recently, a high catalytic activity of ZrN for oxygen reduction was observed [52]. To see whether anything interesting occurs on the ZrN surface, exposed to oxygen, we investigate the surface structure of ZrN with adsorbed oxygen using the global optimization method. We use a surface slab of 4 layers with the 2 bottom layers fixed during the optimization. The unit cell is orthogonal containing 16 zirconium atoms, 16 nitrogen atoms and 1 oxygen atom corresponding to a coverage of one oxygen atom per four zirconium atoms in the surface layer. The electronic exchange-correlation effects are modelled using RPBE [53]. With this setup, the global optimization algorithm finds that a structure where the oxygen atom and also one of the nitrogen atoms occupy the hollow surface sites minimizes the potential energy of the system as shown in Fig. 10. Consequently, there is a nitrogen vacancy in the first layer, below the oxygen atom. The Zr lattice stays close to the cubic (111) surface form, although the 3 Zr atoms that are neighbors to oxygen tend to move away from the oxygen atom and lie closer to the nitrogen on the surface, as compared to the bulk structure.

To verify the surprising finding that one of the N atoms in the unit cell prefers a surface site, we relax the obtained global minimum structure with a maximum residual force of 0.05 eV/Å. In addition, we relax three other structures that are built manually (see Fig. 10). S1: This is a structure where only oxygen is on the surface and the ZrN lattice has its bulk form. S2: In this structure a single nitrogen atom is on the surface and oxygen is moved to the vacancy left behind by the nitrogen. S3: A structure where a single nitrogen is on the surface and oxygen is in the second N-layer. Comparing with these structures, the structure identified by the global search has the lowest energy with energy differences of 0.25 eV/cell,

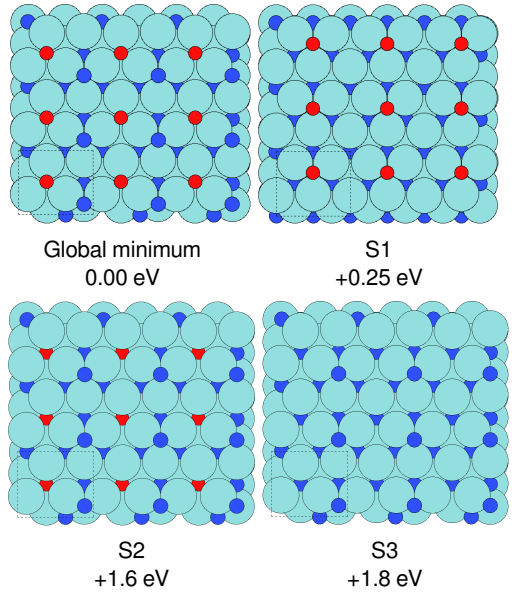


FIG. 10. Investigated structures for ZrN-O surface and their respective total energies. The structures are shown from above the surface. See text for details about the structures. Light blue: Zr, sky blue: N, red: O.

1.6 eV/cell and 1.8 eV/cell to S1, S2, and S3, respectively. The method thus finds a local minimum structure that is lower in energy than the most intuitive configurations.

The progress of one of the successful global optimization runs is shown in Fig. 11. It illustrates that the method is visiting a diverse set of structures. For the first 11 steps, the program produces rather unrealistic structures with energies above 10 eV/cell higher than the global minimum. After step 11, the low-energy structures are exploited more thoroughly, and the global minimum structure is found at step number 22. The program continues with a balance between exploring structures at fairly high energies and identifying competing low energy structures, for example at steps 36 and 61. It is also notable that some non-trivial Zr surface structures are explored, such as those at steps 1 and 29. In general, we note that a fair share of the structures investigated are pretty high in energy. This seems to be necessary to achieve a proper exploration and training of the model.

E. Other systems

One of the main points of this paper is to show how including gradients in the Gaussian process affects the performance of Bayesian global optimization in comparison with GOFEE [13] where only energies are used for training. Our results with both learning curves and suc-

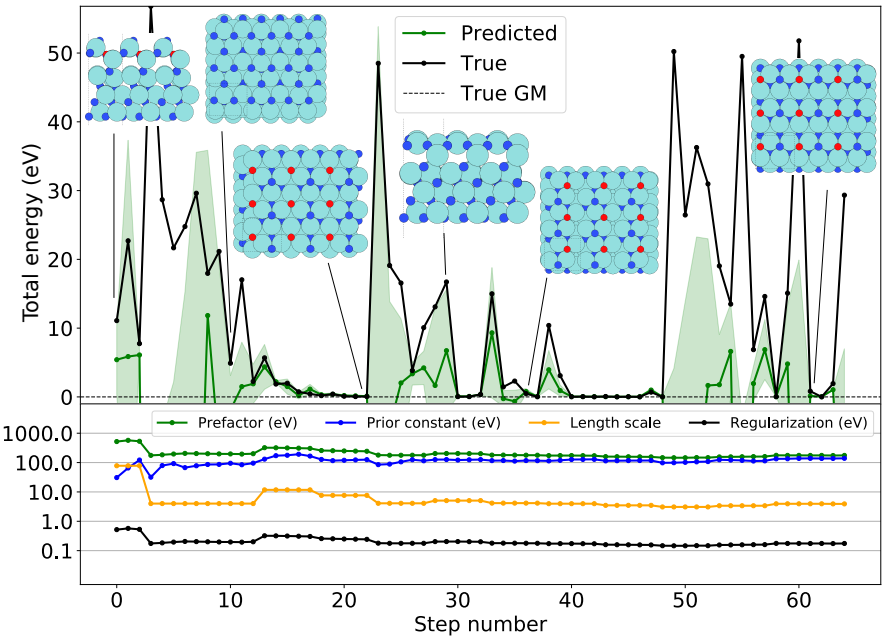


FIG. 11. A single global optimization run of ZrN-O surface. The global minimum structure was found at step 22; the structure is shown in the figure.

cess curves indicate that adding the gradient information into the model improves the performance of the search. In Fig. 12, we show success curves for three more systems, to investigate the effect of training on the gradients as well as the energies. We run several optimizations for a Ti_4O_8 cluster, bulk TiO_2, and bulk silicon. The bulk TiO_2 system consists of 12 atoms and the unit cell is fixed as appropriate for the rutile phase. The bulk silicon system consists of 16 atoms and the unit cell is fixed corresponding to the diamond lattice. For the Ti_4O_8 cluster and bulk silicon the improvement is notable. In contrast, and a bit surprisingly, training on the gradients does not have any effect on the overall performance for bulk TiO_2, so the potential gain of including gradients does depend on the system under study. However, at this point, we cannot tell how much different properties like size, symmetry, number of elements, shape of the true potential etc. matter for the acceleration obtained by using gradients.

VI. COMPUTATIONAL TIME AND MEMORY LIMITATIONS

The primary goal for this work is to reduce the number of expensive DFT calculations necessary to find the global minimum of a PES. However, it should be noted that in some cases the processor time needed to run the relaxations on the surrogate surface may become comparable to the time spent on DFT, especially if the DFT implementation is parallelized efficiently while the Gaussian process is not. In our current implementation each surrogate relaxation is always run on a separate, single processor, but no further parallelization is performed. In the beginning, most of the time of the surrogate model is spent on calculating the fingerprints and their gradients at every step of the local relaxations. Later in a global optimization run, as the training set size becomes larger, calculating the Hessians of the kernel function in fingerprint space is the computational bottleneck. (See the computational times of training and predicting in Supplemental material [36].) In our examples, training the model with gradients typically takes more than one order of magnitude more time compared to using the energies alone. Time consumed in predicting is roughly the same with few training points, but as the number increases, ever larger Hessian matrices need to be calculated with the gradients, leading to an increase in computer time. Predicting without training the gradients is faster since the kernel Hessians are not calculated, and the linear algebra is applied to much smaller matrices. The numerical updating of the length scale involves training the model at each optimization step, and with a large number of atoms and training set sizes, this becomes too heavy in practice with the gradient-trained models unless the training is parallelized. Eventually, if the DFT cal-

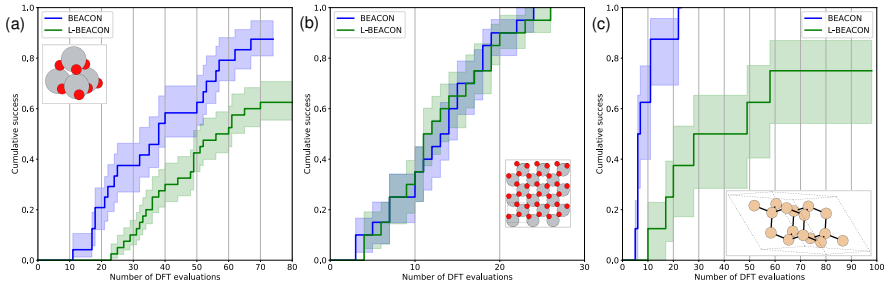


FIG. 12. (a) Ti_4O_8 cluster, (b) bulk TiO_2 and (c) bulk Si/diamond success curves with success threshold 0.05 eV/cell. For $Ti4O_8$ cluster, tight-binding DFT was used as true potential [54, 55], whereas DFT was used for the bulk systems with the same setups as for the SiO_2. The unit cells of TiO_2 and Si include 12 and 16 atoms, respectively.

culation is fast enough, the energy-trained L-BEACON might become faster in total processor time than BEACON even if more DFT calculations are required to train a sufficient model. However, as we observe from the success curves, BEACON is more robust in finding the global minimum and does not get stuck as often as L-BEACON.

The memory usage is limited by inversion of the C-matrix, as required by equation 1, that scales as $\mathcal{O}(n^2)$ memory-wise [8, 16] where n is the order of the square matrix, i.e. in this context $n = N(1+3N_{\mathrm{atoms}})$. If memory or speed becomes an issue with large systems, one is forced to use L-BEACON (or switch to L-BEACON on the fly during the optimization), where we have merely $n =$ number of training points.

VII. CONCLUSIONS

We think that the use of Bayesian strategies and more specifically Gaussian processes for global atomic structure determination is only at its beginning. As demonstrated by Bisbo and Hammer [13, 33], a surrogate model based on a global atomic fingerprint in combination with a Bayesian optimization can outperform earlier global optimization methods by orders of magnitude in reduced computer power. In the present paper we show that including gradient information, *i.e.* the atomic forces, in the training of the model in most cases leads to a further reduction in the number of DFT calculations necessary to identify the global minimum energy structure. The approach was successfully applied to clusters, surfaces, and bulk systems.

However, many aspects of the approach are still unexplored. The surrogate models are not particularly accurate as shown by the learning curves, but still they work very well in the optimization process. So far only a single global fingerprint was investigated, and it is not known if other fingerprints like SOAP [27], MBTR [29], or FCHL [30] would perform even better. The way of suggesting new candidate structures could potentially also be improved, for example in combination with a genetic algorithm, and other acquisition functions may be relevant. Finally, the approach could be combined with other artificial intelligence techniques providing additional guiding of the search.

The code for BEACON is available at `https://gitlab.com/gpatom/ase-gpatom`. It is integrated with the Atomic Simulation Environment (ASE) [38, 39], so that any energy and force calculator supported by ASE can be used together with BEACON. In the present implementation the unit cell is kept fixed during the search. However, it should be possible to update the unit cell based on the currently applied fingerprint, and we expect to implement that in the near future.

ACKNOWLEDGMENTS

We acknowledge support from the VILLUM Center for Science of Sustainable Fuels and Chemicals, which is funded by the VILLUM Fonden research grant (9455).

[1] J. Zhang and V. A. Glezakou, Global optimization of chemical cluster structures: Methods, applications, and challenges, International Journal of Quantum Chemistry **121**, 044114 (2020).
[2] D. J. Wales and J. P. K. Doye, Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms, The Journal of Physical Chemistry A **101**, 5111 (1997), https://doi.org/10.1021/jp970984n.
[3] L. B. Vilhelmsen and B. Hammer, A genetic algorithm for first principles global structure optimization of supported nano structures, The Journal of Chemical Physics **141**, 044711 (2014), https://doi.org/10.1063/1.4886337.
[4] S. V. Lepeshkin, V. S. Baturin, Y. A. Uspenskii,

and A. R. Oganov, Method for simultaneous prediction of atomic structure and stability of nanoclusters in a wide area of compositions, The Journal of Physical Chemistry Letters **10**, 102 (2019), https://doi.org/10.1021/acs.jpclett.8b03510.

[5] M. Jäger, R. Schäfer, and R. L. Johnston, Giga: a versatile genetic algorithm for free and supported clusters and nanoparticles in the presence of ligands, Nanoscale **11**, 9042 (2019).

[6] Z. Chen, W. Jia, X. Jiang, S.-S. Li, and L.-W. Wang, Sgo: A fast engine for ab initio atomic structure global optimization by differential evolution, Computer Physics Communications **219**, 35 (2017).

[7] C. J. Pickard and R. J. Needs, Ab initio random structure searching, Journal of Physics: Condensed Matter **23**, 053201 (2011).

[8] E. Garijo del Río, J. J. Mortensen, and K. W. Jacobsen, Local bayesian optimizer for atomic structures, Phys. Rev. B **100**, 104103 (2019).

[9] S. Carr, R. Garnett, and C. Lo, Basc: Applying bayesian optimization to the search for global minima on potential energy surfaces, in *Proceedings of The 33rd International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 48, edited by M. F. Balcan and K. Q. Weinberger (PMLR, New York, New York, USA, 2016) pp. 898–907.

[10] T. Yamashita, N. Sato, H. Kino, T. Miyake, K. Tsuda, and T. Oguchi, Crystal structure prediction accelerated by bayesian optimization, Phys. Rev. Materials **2**, 013803 (2018).

[11] H. L. Mortensen, S. A. Meldgaard, M. K. Bisbo, M.-P. V. Christiansen, and B. Hammer, Atomistic structure learning algorithm with surrogate energy model relaxation, Phys. Rev. B **102**, 075427 (2020).

[12] M. O. Jäger, E. V. Morooka, F. F. Canova, L. Himanen, and A. S. Foster, Machine learning hydrogen adsorption on nanoclusters through structural descriptors, npj Computational Materials **4**, 1 (2018).

[13] M. K. Bisbo and B. Hammer, Efficient global structure optimization with a machine-learned surrogate model, Physical Review Letters **124**, 086102 (2020).

[14] B. C. Rinderspacher, Heuristic global optimization in chemical compound space, The Journal of Physical Chemistry A **124**, 9044 (2020), pMID: 33079549, https://doi.org/10.1021/acs.jpca.0c05941.

[15] L. Fang, E. Makkonen, M. Todorović, P. Rinke, and X. Chen, Efficient amino acid conformer search with bayesian optimization, Journal of Chemical Theory and Computation **17**, 1955 (2021), pMID: 33577313, https://doi.org/10.1021/acs.jctc.0c00648.

[16] O.-P. Koistinen, F. B. Dagbjartsdóttir, V. Ásgeirsson, A. Vehtari, and H. Jónsson, Nudged elastic band calculations accelerated with gaussian process regression, The Journal of Chemical Physics **147**, 152720 (2017), https://doi.org/10.1063/1.4986787.

[17] J. A. Garrido Torres, E. Garijo del Río, V. Streibel, M. H. Hansen, T. S. Choksi, J. J. Mortensen, A. Urban, M. Bajdich, F. Abild-Pedersen, K. W. Jacobsen, and T. Bligaard, An artificial intelligence approach for navigating potential energy surfaces (2021), in preparation.

[18] A. Denzel and J. Kästner, Gaussian process regression for transition state search, Journal of Chemical Theory and Computation **14**, 5777 (2018).

[19] M. Todorović, M. U. Gutmann, J. Corander, and P. Rinke, Bayesian inference of atomistic structure in functional materials, Npj computational materials **5**, 1 (2019).

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

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

[22] J. Vandermause, S. B. Torrisi, S. Batzner, Y. Xie, L. Sun, A. M. Kolpak, and B. Kozinsky, On-the-fly active learning of interpretable bayesian force fields for atomistic rare events, npj Computational Materials **6**, 20 (2020).

[23] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, 2006).

[24] C. M. Bishop, *Pattern recognition and machine learning* (New York : Springer, [2006] ©2006, 2006) textbook for graduates.;Includes bibliographical references (pages 711-728) and index.

[25] A. S. Christensen and O. A. von Lilienfeld, On the role of gradients for machine learning of molecular energies and forces, Machine Learning: Science and Technology **1**, 045018 (2020).

[26] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, Fast and accurate modeling of molecular atomization energies with machine learning, Phys. Rev. Lett. **108**, 058301 (2012).

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

[28] J. Behler, Atom-centered symmetry functions for constructing high-dimensional neural network potentials, The Journal of Chemical Physics **134**, 074106 (2011), https://doi.org/10.1063/1.3553717.

[29] H. Huo and M. Rupp, Unified representation of molecules and crystals for machine learning (2018), arXiv:1704.06439 [physics.chem-ph].

[30] A. S. Christensen, L. A. Bratholm, F. A. Faber, and O. Anatole von Lilienfeld, Fchl revisited: Faster and more accurate quantum machine learning, The Journal of Chemical Physics **152**, 044107 (2020), https://doi.org/10.1063/1.5126701.

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

[32] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, Taking the human out of the loop: A review of bayesian optimization, Proceedings of the IEEE **104**, 148 (2016).

[33] M. K. Bisbo and B. Hammer, Global optimization of atomistic structure enhanced by machine learning, arXiv.org , arXiv:2012.15222 (2020), 2012.15222.

[34] J. Wu, M. Poloczek, A. G. Wilson, and P. Frazier, Bayesian optimization with gradients, in *Advances in Neural Information Processing Systems* (2017) pp. 5267–5278.

[35] M. Valle and A. R. Oganov, Crystal fingerprint space – a novel paradigm for studying crystal-structure sets, Acta Crystallographica Section A **66**, 507 (2010), https://doi.org/10.1107/S0108767310026395.

[36] See Supplemental Material at [URL will be inserted by publisher] for the formulas for kernel gradients, supplementary results, and the details of the computational performance of BEACON.

[37] K. W. Jacobsen, J. K. Norskov, and M. J. Puska, Interatomic interactions in the effective-medium theory, Phys. Rev. B **35**, 7423 (1987).

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

[39] Atomic Simulation Environment (ASE), `https://wiki.fysik.dtu.dk/ase/` (2020).

[40] J. P. Perdew, K. Burke, and M. Ernzerhof, Generalized gradient approximation made simple, Phys. Rev. Lett. **77**, 3865 (1996).

[41] J. Enkovaara, C. Rostgaard, J. J. Mortensen, J. Chen, M. Dułak, L. Ferrighi, J. Gavnholt, C. Glinsvad, V. Haikola, H. A. Hansen, H. H. Kristoffersen, M. Kuisma, A. H. Larsen, L. Lehtovaara, M. Ljungberg, O. Lopez-Acevedo, P. G. Moses, J. Ojanen, T. Olsen, V. Petzold, N. A. Romero, J. Stausholm-Møller, M. Strange, G. A. Tritsaris, M. Vanin, M. Walter, B. Hammer, H. Häkkinen, G. K. H. Madsen, R. M. Nieminen, J. K. Nørskov, M. Puska, T. T. Rantala, J. Schiøtz, K. S. Thygesen, and K. W. Jacobsen, Electronic structure calculations with GPAW: a real-space implementation of the projector augmented-wave method, Journal of Physics: Condensed Matter **22**, 253202 (2010).

[42] B. Huang and O. A. von Lilienfeld, Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity, The Journal of Chemical Physics **145**, 161102 (2016).

[43] B. Huang and O. A. von Lilienfeld, Ab initio machine learning in chemical compound space, arXiv 2012.07502 (2020), arXiv:2012.07502 [physics.chem-ph].

[44] B. Parsaeifard, D. S. De, A. S. Christensen, F. A. Faber, E. Kocer, S. De, J. Behler, A. von Lilienfeld, and S. Goedecker, An assessment of the structural resolution of various fingerprints commonly used in machine learning, Machine Learning: Science and Technology **2**, 015018 (2020).

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

[46] M. S. Jørgensen, U. F. Larsen, K. W. Jacobsen, and B. Hammer, Exploration versus Exploitation in Global Atomistic Structure Optimization, The Journal of Physical Chemistry A **122**, 1504 (2018).

[47] S. Pérez-Walton, C. Valencia-Balvín, A. C. M. Padilha, G. M. Dalpian, and J. M. Osorio-Guillén, A search for the ground state structure and the phase stability of tantalum pentoxide, Journal of Physics: Condensed Matter **28**, 035801 (2015).

[48] Y. Yang and Y. Kawazoe, Prediction of new ground-state crystal structure of Ta_2o_5, Phys. Rev. Materials **2**, 034602 (2018).

[49] J.-H. Yuan, K.-H. Xue, Q. Chen, L. R. C. Fonseca, and X.-S. Miao, Ab initio simulation of ta2o5: A high symmetry ground state phase with application to interface calculation, Annalen der Physik **531**, 1800524 (2019), https://doi.org/10.1002/andp.201800524.

[50] S. Srivastava, J. P. Thomas, N. Heinig, M. Abd-Ellah, M. A. Rahman, and K. T. Leung, Efficient photoelectrochemical water splitting on ultrasmall defect-rich TaO xnanoclusters enhanced by size-selected Pt nanocluster promoters, Nanoscale **9**, 14395 (2017).

[51] Y. Chen, J. L. G. Fierro, T. Tanaka, and I. E. Wachs, Supported Tantalum Oxide Catalysts: Synthesis, Physical Characterization, and Methanol Oxidation Chemical Probe Reaction, The Journal of Physical Chemistry B **107**, 5243 (2003).

[52] Y. Yuan, J. Wang, S. Adimi, H. Shen, T. Thomas, R. Ma, J. P. Attfield, and M. Yang, Zirconium nitride catalysts surpass platinum for oxygen reduction, Nature Materials **19**, 282 (2020).

[53] B. Hammer, L. B. Hansen, and J. K. Nørskov, Improved adsorption energetics within density-functional theory using revised perdew-burke-ernzerhof functionals, Phys. Rev. B **59**, 7413 (1999).

[54] B. Hourahine, B. Aradi, V. Blum, F. Bonafé, A. Buccheri, C. Camacho, C. Cevallos, M. Y. Deshaye, T. Dumitrică, A. Dominguez, S. Ehlert, M. Elstner, T. van der Heide, J. Hermann, S. Irle, J. J. Kranz, C. Köhler, T. Kowalczyk, T. Kubař, I. S. Lee, V. Lutsker, R. J. Maurer, S. K. Min, I. Mitchell, C. Negre, T. A. Niehaus, A. M. N. Niklasson, A. J. Page, A. Pecchia, G. Penazzi, M. P. Persson, J. Řezáč, C. G. Sánchez, M. Sternberg, M. Stöhr, F. Stuckenberg, A. Tkatchenko, V. W.-z. Yu, and T. Frauenheim, Dftb+, a software package for efficient approximate density functional theory based atomistic simulations, The Journal of Chemical Physics **152**, 124101 (2020), https://doi.org/10.1063/1.5143190.

[55] R. Luschtinetz, J. Frenzel, T. Milek, and G. Seifert, Adsorption of phosphonic acid at the tio2 anatase (101) and rutile (110) surfaces, The Journal of Physical Chemistry C **113**, 5730 (2009), https://doi.org/10.1021/jp8110343.