Accelerating catalysis simulations using surrogate machine learning models

Vishart, Andreas Lynge

DTU Energy
Department of Energy Conversion and Storage

Accelerating catalysis simulations using surrogate machine learning models

PhD Thesis

Andreas Lynge Vishart

Accelerating catalysis simulations using surrogate machine learning models

PhD Thesis
Submitted January 31, 2023

Author
Andreas Lynge Vishart
alyvi@dtu.dk

Supervisor
Professor Thomas Bligaard
Section for Atomic Scale Materials Modelling
Department of Energy Conversion and Storage
Technical University of Denmark
tbli@dtu.dk

Co-supervisor
Associate Professor Karen Chan
Catalysis Theory Center
Department of Physics
Technical University of Denmark
kchan@fysik.dtu.dk

Cover photo: Vibeke Hempler, 2012
Published by: DTU, Department of Energy Conversion and Storage, Anker Engelunds Vej , Building 301, 2800 Kgs. Lyngby Denmark
www.energy.dtu.dk/english

Abstract

Climate change is evident, and it calls for an immediate global transition to a green and sustainable energy structure. However, an effective transition requires the discovery of new materials for solar cells, batteries, catalysts, etc. Artificial intelligence, or machine learning, has proven that it can accelerate the search for new materials significantly. A Gaussian process can be a self-taught machine learning method by applying an active learning approach since the Gaussian process can predict energies and corresponding uncertainty estimations. Thereby, a substantial amount of time is saved on the manual setup of databases and screenings for new materials.

In this thesis, the robustness of a Gaussian process and how common mistakes are avoided when training the Gaussian process are discussed. A correction to the covariance matrix is derived, which ensures that exception errors are avoided when the Gaussian process is optimized. Furthermore, boundary conditions for the hyperparameters are defined, which makes variable transformations of the hyperparameters possible. The variable transformations make the important regions of the hyperparameter space larger and more probable without restricting the hyperparameters. By applying the variable transformation, a new method is developed that globally optimizes the hyperparameters. The new method locates the global maximum for the hyperparameters in all the test systems with different training set sizes, which is not the case for any other investigated optimizers. Another important advantage of the new method is that the time of the optimization is reduced compared to the other investigated global optimizers. Therefore, a new method has been implemented which makes the Gaussian process robust and reliable.

Different objective functions are tested to investigate if they improve the Gaussian process. The most used objective function, log-likelihood, is confirmed to be the best objective function in terms of the prediction of energies and uncertainties for the chosen test systems. The evaluation was possible due to a newly defined uncertainty measure. The uncertainty predictions from the Gaussian process are improved by modifying the solution obtained from log-likelihood without changing the energy predictions or increasing the computational cost.

The uncertainty predictions are also improved by deriving a new process called a Student's t process. The new process has the same energy predictions as the Gaussian process, but it has one hyperparameter less, which is removed with a Bayesian approach. The fully Bayesian solution to the predictions of the energies and uncertainties is approximated by applying the Kullback-Leibler divergence. This is a substantial improvement to the uncertainty predictions. The approximated solution does not require retraining of the Gaussian process to predict a new point, which is normally required for a fully Bayesian solution.

A developed structure optimization method for finding the most stable adsorption structure for any surface is presented. The optimization method finds the most stable adsorption structures for all tested systems. Furthermore, the quantum calculations are significantly reduced by a factor of 40. This reduction is expected to be even larger for more complex surfaces. The new robust Student's t process is implemented into a new version of the machine learning accelerated Nudged Elastic Band method, which is essential for finding activation energies. A reduction factor of 200 compared to the required quantum mechanical calculations for the Nudge Elastic Band method is obtained. Therefore, it is expected that the developed and robust methods can be powerful tools in automated material discovery.

Resumé

Klimaforandringerne er tydelige, og der er brug for en omgående omstilling til en grøn og bæredygtig energistruktur på verdensplan. Dog er det et krav for en effektiv omstilling, at der findes nye materialer inden for solceller, batterier, katalysatorer med mere. Kunstig intelligens, eller maskinlæring, har vist, at de kan accelerere søgningen efter nye materialer væsentligt. En Gaussisk proces er i stand til at være en selvlærende maskinlæringsmetode ved hjælp af en aktiv læringstilgang, da den kan forudsige energier og tilsvarende usikkerhedsestimater. Derved kan der spares betydelig manuel tid på oprettelsen af databaser og søgning efter nye materialer.

I denne afhandling diskuteres robustheden af en Gaussisk proces, og hvordan man undgår de hyppige fejl ved træningen af den, hvilket er essentielt for dens benyttelse. Der er blevet udledt en korrektion til kovariansmatricen, som gør, at der ikke opstår fejl, når den Gaussiske proces bliver optimeret. Derudover er der blevet defineret grænsebetingelser for hyperparameterne i den Gaussiske proces, som gør det muligt at lave variabeltransformationer af hyperparameterne. Variabeltransformationerne gør de vigtige dele af hyperparameterrummet større og mere sandsynlige uden at begrænse hyperparameterne. Ved brug af variabeltransformationerne er der blevet udviklet en ny metode, som optimerer hyperparameterne globalt. Den nye metode finder det globale maksimum for hyperparameterne i alle undersøgte test-systemer med forskellige træningssæt-størrelser, hvilket ikke er opnået med de andre undersøgte optimeringsmetoder. En anden vigtig fordel ved den nye metode er, at optimeringstiden er reduceret i forhold til andre globale optimeringsmetoder. Altså er der implementeret en ny metode, som gør den Gaussiske proces robust og pålidelig.

Flere forskellige objektive funktioner er blevet testet for at undersøge, om de forbedrer den Gaussiske proces. Den mest brugte objektive funktion, log-likelihood, er blevet bekræftet som værende den bedste objektive funktion til forudsigelser af energier og deres usikkerhedsestimater af de valgte testsystemer. Evalueringen var mulig på grund af et nyt defineret usikkerhedsmål. Usikkerhedsforudsigelserne fra den Gaussiske proces er også blevet forbedret ved at modificere løsningen fra log-likelihood, uden at det ændrer på energiforudsigelserne eller forøger beregningsomkostningerne.

En forbedring til usikkerhedsforudsigelserne er også opnået ved at udlede en helt ny proces, kaldet en Students t proces. Den nye proces har samme energiforudsigelser, som den Gaussiske proces, men den har en hyperparameter mindre, som er blevet fjernet Bayesiansk. Den fulde Bayesianske løsning til forudsigelse af energierne og usikkerhederne er blevet estimeret med brug af Kullback–Leibler divergens. Dette giver en markant forbedring til usikkerhedsforudsigelserne. Denne estimerede løsning kræver ikke en gentræning af den Gaussiske proces for forudsigelser af helt nye punkter, hvilket almindeligvis er tilfældet for fulde Bayesianske løsninger.

En nyudviklet strukturoptimeringsmetode for at finde de mest stabile adsorptionsstrukturer for en vilkårlig overflade er blevet præsenteret. Optimeringsmetoden finder de mest stabile adsorptionsstrukturer for alle testede systemer. Derudover er de kvanatemekaniske beregninger blevet betydeligt reduceret med op til en faktor 40. Denne reduktion forudsiges til at være endnu større for mere komplicerede overflader. Den nye robuste Students t proces er blevet integreret i en ny version af den maskinlæring-accelererede ”Nudged Elastic Band”-metode, som er essentiel for at finde aktiveringsenergier. En reduktionsfaktor på op til 200 i forhold til de påkrævede antal kvantemekaniske beregninger for ”Nudged Elastic Band”-metoden er opnået. Derfor kan det forventes at disse udviklede og robuste metoder vil være stærke værktøjer i automatiserede materialesøgninger.

Preface

This thesis is submitted in candidacy for a Doctor of Philosophy (PhD) degree from the Technical University of Denmark (DTU). The work has been carried out between February 2020 and January 2023 at the Section for Atomic Scale Materials Modelling (ASM) at the Department of Energy Conversion and Storage. The studies have been supervised by Thomas Bligaard.

Kongens Lyngby, January 31, 2023

Andreas L Vishart

Andreas Lynge Vishart

Acknowledgements

First and foremost, I would like to say a sincere thank you to my supervisor, Thomas Bligaard, for a fantastic and exciting project with countless hours of exciting profound discussions. Thank you for always prioritizing helping me and keeping a positive mind.

I also owe a big gratitude to Kirsten Winther for an incredible external stay at SLAC and for many great ideas to the project. Furthermore, I will also like to thank the members of the SUNCAT group for a warm welcome and for sharing their very interesting research.

Thanks to Pablo Lustemberg and M. Veronica Ganduglia Pirovano at CSIC for an insightful collaboration. Furthermore, I would like to thank the members of the I-LINK project for a great experience in Madrid with an introduction to some new knowledge.

A special thanks to the participants of the weekly machine learning meeting at CAMD for sharing their very relevant and insightful research. Especially thank you to Karsten Wedel Jacobsen for teaching and explaining his comprehensive knowledge about Bayesian statistics and Gaussian processes. A special thanks to Casper Larsen for discussing the coding of the Gaussian process.

I also want to thank all former and current members of the section for Atomic Scale Materials Modelling and the section for Autonomous Materials Discovery. Thanks for all their many great ideas and discussions. A special thanks to Shuang Han for detailed discussions of MLNEB and to Bjarke Arnskjær Hastrup for discussions about machine learning in general.

I also owe a thank you to all former and current members of CatTheory for a great insight into catalysis. A special thanks to Georg Kastlunger for his encouragement and curiosity. Thanks to Benjamin Thomas Hinrichsen for a fruitful collaboration that hopefully will continue.

Thanks to Jose Antonio Garrido Torres for sharing his code with MLNEB and being a part of the introduction to my project.

Especially, I owe enormous gratitude to Pernille Pedersen for her indispensable support, encouragement, help, and proofreading.

I am also very grateful to my friends and family for their support, interest, and understanding during my busy three years. Last but not least, I owe Kristine Rask Andreasen a heartfelt appreciation for always supporting and encouraging me. I could not have finished this thesis without your support.

List of publications

Paper I

Best Conventional Gaussian Process
Andreas Lynge Vishart and Thomas Bligaard
To be submitted

Paper II

Machine-learning enabled optimization of atomic structures using atoms with fractional existence
Casper Larsen, Sami Kaappa, Andreas Lynge Vishart, Thomas Bligaard, and Karsten Wedel Jacobsen
Submitted to *Physical Review Letters*

Acronyms

AIE	All-Image-Evaluation method. 48, 50
ASE	Atomic Simulation Environment. 36, 37, 41, 42, 45, 46, 65, 66
AuAl	A gold atom on an aluminium(100) surface. 17, 47, 48, 65
BC	Boundary Conditions. 18, 25, 32
BFGS	Broyden–Fletcher–Goldfarb–Shanno. 20
CG	Conjugate gradient. 20
CI-NEB	Climbing Image Nudged Elastic Band method. 7, 46
CONi	Carbon monoxide on a nickel(111) surface. 17, 66
CPUs	Central Processing Units. 36, 37
Cu13	A cluster of thirteen copper atoms. 17, 66
Cu5	A cluster of five copper atoms. 17, 66
CV	Cross-Validation. 22
DFT	Density Functional Theory. 5, 40–45, 49, 51, 66
EGBC	Educated Guessed Boundary Conditions. 18, 20, 21, 32
EMT	Effective Medium Theory. 45, 47, 49, 51, 65, 66
FBMGP	Fully Bayesian Mimicking Gaussian Process. 14, 17, 31, 53
GGA	Generalized Gradient Approximation functional. 6
GMES	Global Minimum Energy Structure. 35, 36, 39–44, 54
GP	Gaussian Process. 3, 4, 7, 8, 10, 12–14, 17–19, 22–24, 28–33, 38, 39, 47, 53, 54, 65
GPE	Geisser's Predictive mean square Error. 22, 29
GPP	Geisser's surrogate Predictive Probability. 22, 29
H2Cufcc	Hydrogen atoms at fcc sites on a copper(111) surface. 47

H2Cuhcp	Hydrogen atoms at hcp sites on a copper(111) surface. 47
Heptamer	Platinum heptamer island on a platinum(111) surface. 47
IDPP	Image Dependent Pair Potential. 45
KL	Kullback–Leibler divergence. 13, 14, 53, 65
KS	Kohn-Sham. 5, 6
L-BFGS-B	Limited-memory BFGS with boundaries. 20, 26, 67
LDA	Local Density Approximation functional. 6
LL	Log-Likelihood. 11–13, 17, 18, 20–29, 31–33, 53
LOOCV	Leave-One-Out Cross-Validation. 22, 28, 29, 69
LP	Log-Posterior. 11, 14, 15, 20, 22, 23, 31, 33, 38, 47
MAP	Maximum A Posteriori estimation. 11, 31, 53
MB	The Müller-Brown potential energy surface. 17, 47, 49, 51, 65
MD	Molecular Dynamics simulation. 2, 45, 66
MEP	Minimum Energy Path. 2–4, 6, 45, 48–51, 53, 54
mGGA	Meta Generalized Gradient Approximation functional. 6
ML	Machine Learning. 3, 4, 22, 35–38, 40, 42, 44–51, 53, 54
MLE	Maximum Likelihood Estimation. 11, 13, 17, 20, 24, 31–33, 53
MLGO	Machine Learning Accelerated Global Adsorption Optimization method. 36, 38–44, 53, 54
MLNEB	Machine Learning Accelerated Nudged Elastic Band method. 44–51, 53, 54
N2Cufcc	Nitrogen atoms at fcc sites on a copper(111) surface. 47
N2Cuhcp	Nitrogen atoms at hcp sites on a copper(111) surface. 47
NEB	Nudged Elastic Band method. 2–4, 6, 7, 45, 46, 48–51, 53, 54
NLPP	Negative Log Predictive Probability. 19, 30, 31
NVT	Canonical ensemble. 66

O2Pt	Two oxygen atoms adsorbed on a platinum(100) surface. 17
OIE	One-Image-Evaluation method. 45, 48, 50
Oxad	Oxadiazoline formation from ethene and Nitrous oxide. 47
PES	Potential Energy Surface. 2, 3, 35, 37, 46, 48–51, 53, 54
QM	Quantum Mechanical. 2, 3, 5
RMSE	Root-Mean-Square Error. 19
SDGs	Sustainable Development Goals. 1
SE	Schrödinger Equation. 2, 5
SEC	Squared Exponential Covariance. 8, 9, 11, 18, 24, 38
SLURM	Simple Linux Utility for Resource Management. 71
SP	Saddle Point. 2, 7, 49, 51
TerPt	A platinum atom on a platinum terrace surface. 47
TNC	Truncated Newton. 20, 26, 67
TP	Student’s T Process. 12, 13, 17, 29–31, 33, 45, 47, 53, 54
TS	Transition State. 2, 3
UD	Uncertainty Deviation. 19, 53
WaterPt	Four water molecules above a platinum(111) surface. 17
XC	Exchange-correlation. 6, 38, 42, 47

Contents

1 Introduction

1.1 Energy demand

The global energy demand is higher than ever and is still increasing despite a growing political awareness of the need to decrease carbon dioxide (CO_2) emissions by transitioning to renewable energy sources[1]. This increase is due to a growing global population that requires more energy. Unfortunately, a large part of the global energy consumption (83% in 2021) is still from fossil fuels[1]. There is no doubt that climate change is due to the high levels of CO_2 and other greenhouse gas emissions[2]. The correlation between the increase in the average global temperature[3, 4] and the emission of CO_2 due to energy consumption is evident (see Fig. 1.1). The rapid increase in the average temperature has devastating consequences for the climate, which is increasingly affecting the biodiversity, wildlife, and humans in all areas of the world.

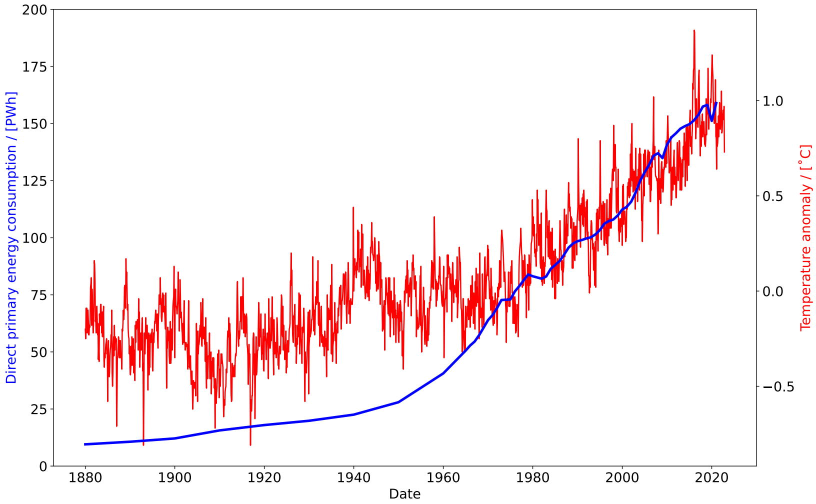


Figure 1.1: The global energy consumption (blue curve) and the average temperature anomaly compared to the mean temperature of the years 1951–1980 (red curve) as a function of time in years. The source data is from the references [1, 3, 4].

Political initiatives to raise awareness of climate change and the need to transition to a CO_2-neutral society have been taken in recent years. The 17 Sustainable Development Goals (SDGs) established by the United Nations is an example of an initiative for acting on climate changes[5]. Actions toward a society with net zero CO_2 emissions will primarily affect the SDGs: ”7. Affordable and clean energy”, ”8. Decent work and economic growth”, ”12. Responsible consumption and production”, and ”13. Climate action”. In the long run, climate actions will secondarily affect most of the SDGs, including the ”14. life below water” and ”15. life on land”.

Despite political initiatives, it is clear that there is an urgent need to accelerate the transition to renewable energy. Energy storage is an essential prerequisite for a society based entirely on renewable energy, which is a fluctuating energy source. To harvest, store, and convert renewable energy, new and improved materials are central[6, 7]. Solar cells are an essential part of renewable energy harvesting with excessive energy potential[8, 9]. However, new materials are required in solar cells to get higher efficiency, reduce the cost, find non-toxic materials, etc. [8, 9, 10, 11]. The energy must be stored when the renewable energy sources fluctuate and for mobility. The energy can e.g. be stored in batteries or as fuels. Batteries are an indispensable part of our society and a growing part of the transportation sector. Therefore, the discovery of new sustainable materials for batteries with higher energy density, faster charging, higher safety, lower cost, etc. is crucial[12]. The conversion from electricity from renewable energy sources to fuels (Power-to-X) is essential for a sustainable society with net zero CO_2 emissions[13]. The fuels can be hydrogen, methane, methanol, etc. and they can be stored for a long time and have high energy density[14]. However, the current catalysts are expensive and inadequate[13, 15, 16]. Thus, the discovery of improved catalyst materials is a key prospective of the green transition. However, the experimental search for new materials can take 10-15 years[17, 18]. In recent years, significant improvements in computational resources have allowed computational chemistry to emerge as a significantly faster way to screen for new materials.

1.2 Machine learning in quantum mechanics

A reaction mechanism consists of multiple elementary reactions. An elementary reaction has an initial state and a final state. The transition path on the Potential Energy Surface (PES) from the initial state to the final state has a Saddle Point (SP). The energy difference between the energy of the initial state and the SP is the activation energy. The reaction rate is dependent on the activation energies. Thus, all the energies of the initial and final states with the SPs must be calculated for all the elementary reactions in the studied reaction mechanism. The energies are calculated with Quantum Mechanical (QM) methods. The QM methods calculate the energy and the electronic structure of the atomistic system from the Schrödinger Equation (SE)[19]. The atomistic structures of the initial and final states are obtained by structure optimization of the energy. The SP is acquired from either a Transition State (TS) search or the Minimum Energy Path (MEP)[20, 21, 22]. The MEP is the transition path from the initial state to the final state with the lowest energy and therefore the most probable transition. The Nudged Elastic Band method (NEB) is the standard method for finding the MEP[23, 22, 24] (see Section 2.2). Thus, a single reaction mechanism requires many computationally expensive QM calculations. Furthermore, thousands or more reaction mechanisms are possible[25]. In material discovery, different surfaces are also studied which makes the required number of QM evaluations even vaster.

Another approach for studying a reaction is Molecular Dynamics simulation (MD)[26]. The MD is initialized from an initial state and specifications of its physical environment. The dynamics of the atoms are then calculated, giving an accurate description of the equilibrium structures under the given physical environment. However, in a MD, only a single trajectory is treated, and the exact structure of the lowest SP is rarely observed. Therefore, despite the huge amount of computationally expensive QM calculations, there is no guarantee that the resulting reaction path is the MEP. Metadynamics can be applied to enforce the MD into unsampled chemical space and therefore more likely to sample SPs and products of reactions[27, 28, 29, 30, 31].

An essential approach to automate the screening for new catalysts and materials are by applying workflows and high-throughput screening[32, 33, 16, 34, 35, 36, 37, 38]. The chemical space is too vast to consider manually. Therefore, a range of automated subsequent computational tasks is a must. Furthermore, the results from workflows are reliable and reproducible. High-throughput screening starts by considering a huge number of materials with a computationally inexpensive calculation method and then decreasing the number of considered systems while increasing the accuracy of the calculation method for the price of a higher computational cost. However, the workflows and high-throughput screening still rely on the standard methods for structure optimizations and TS searches.

The standard methods are extremely computationally expensive and not feasible for most purposes. Therefore, acceleration in the standard methods are essential given the urgent need for new energy harvesting and storage materials. Machine Learning (ML) has recently shown to be an essential tool for accelerating structure optimizations, TS searches, and the QM calculations themselves [39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 31].

ML models can be categorized as supervised, unsupervised, or reinforcement learning[49, 17]. In supervised learning, data of pairs is given to the ML model. The pairs consist of a feature, also called a descriptor or a fingerprint, that uniquely describes the data, e.g. a vector that describes the atomic configuration of a system, and a target, also called a label, that is the observable of interest, e.g. the potential energy. The correlation between the features and the targets is then learned by the ML model so that it can predict a target from a new given feature. Unsupervised learning uses only the features, often to cluster the data and find patterns. In reinforcement learning, an agent is used to interact with an environment through some actions that give rewards.

Supervised learning is often used for learning the PES to avoid QM calculations[39, 40, 41, 42, 43, 44, 46, 47, 48, 31]. The prediction time of the PES with the ML models is negligible compared to the QM calculations. Thus, the structure optimizations and TS searches can be significantly accelerated. However, a ML model requires a database of atomistic systems with energies from QM calculations similar to the atomistic systems studied. The training time of the ML model must be taken into account since the computational complexity of the used ML model scales with the number of training points. Generating a database can be time-consuming in terms of the user's time and the computational time due to the many QM calculations carried out. In active learning, the ML model is enabled to decide the data that is included in the database[50, 51]. Active learning optimizes an acquisition function to find the next data point that is evaluated by a QM calculation. The acquisition function can be, e.g. an uncertainty prediction, an energy prediction, or a combination of the two. Hence, the database consists of the structures suggested by the ML model that gives the most information. Therefore, the database becomes as small as possible, and no data generation and assumptions are required. The Gaussian Process (GP) is an example of a ML model well suited for active learning since it predicts an observable and a corresponding uncertainty[52, 53]. Furthermore, the GP also performs well with only a small number of training points.

Local structure optimizations have been accelerated with ML and GPes[44]. Global structure optimizations of catalyst and cluster compositions have also been accelerated significantly with GPes [41, 47, 48]. The NEB has also been significantly accelerated with GPes [42, 43, 54]. Reliable and robust ML models are the foundation of using ML for accelerating the standard methods that find stable structures and MEPs. Often, the GP is optimized by a local optimization of its hyperparameters, which is unreliable.

1.3 Outline of thesis

In this thesis, the hyperparameter optimization of a GP is studied to achieve a robust ML model that can be used for accelerating catalysis simulations. A robust ML model enables an active learning scheme that is used for constructing a database with no or few previous data points. Therefore, global adsorption searches are significantly accelerated with ML without predefining databases. Furthermore, the NEB is also accelerated with ML for locating the MEP and obtaining the activation energy.

The fundamental theory used throughout the thesis is described in Chapter 2. Furthermore, new methods and equations are introduced and derived. The new methods and equations include two new forms of the objective function optimized in a GP and a new ML model.

In Chapter 3, the problematics of optimizing the hyperparameters are discussed, and approaches to avoid them are introduced. Furthermore, new methods are implemented and explained to robustly optimize the hyperparameters of the GP. A better Bayesian approach of the GP is introduced as a new ML model. At last, an approximation for a fully Bayesian approach is presented and discussed.

A global adsorption search method with ML is introduced in Chapter 4. The method uses a simple fingerprint and a robust GP to search all adsorption positions of simple adsorbates at different surfaces.

At last, the NEB is significantly accelerated by applying a new robust ML model with better uncertainty predictions in Chapter 5.

2 Theory

2.1 Electronic structure theory

QM calculations are performed by solving the time-independent SE[19]. The time-independent SE is expressed as:

$$\hat{H}\Psi\left(\{\vec{r}\},\{\vec{R}\}\right) = E\Psi\left(\{\vec{r}\},\{\vec{R}\}\right) \tag{2.1}$$

where $\hat{H}$ is the Hamiltonian operator, E, the corresponding eigenvalue, is the total energy, Ψ is the total wavefunction of the electrons with their coordinates $\{\vec{r}\}$ and the nuclei with their coordinates $\{\vec{R}\}$. The Hamiltonian operator for an atomistic system is expressed as:

$$\begin{aligned}\hat{H} =& \frac{-\hbar^2}{2}\sum_{i}^{N_N}\frac{1}{M_i}\hat{\nabla}_i^2 + \frac{q_e^2}{8\pi\epsilon_0}\sum_{i=1}^{N_N}\sum_{j\neq i}\frac{Z_iZ_j}{|R_i - R_j|}\\ &+ \frac{-\hbar^2}{2m_e}\sum_{i=1}^{N_e}\hat{\nabla}_i^2 + \frac{q_e^2}{8\pi\epsilon_0}\sum_{i=1}^{N_e}\sum_{j\neq i}\frac{1}{|r_i - r_j|} - \frac{q_e^2}{4\pi\epsilon_0}\sum_{i=1}^{N_N}\sum_{j=1}^{N_e}\frac{Z_i}{|\vec{R}_i - \vec{r}_j|}\end{aligned} \tag{2.2}$$

where N_N is the number of nuclei in the system, M_i is the mass of nucleus i, $\hat{\nabla}_i^2$ is the second derivative wrt. to the Cartesian coordinates of nucleus or electron i, q_e is the charge of one electron, Z_i is the number of protons in nucleus i, ϵ_0 is the vacuum permittivity, and m_e is the mass of an electron.

The Born-Oppenheimer approximation[55] assumes that the wavefunction can be separated into nuclear and electronic parts since the nuclei can be assumed to be stationary relative to the electrons due to the mass and speed differences. Hence, the Hamiltonian operator can also be separated into nuclear and electronic parts. The nuclear Hamiltonian operator consists of the two first terms from Eq. 2.2, and the electronic Hamiltonian operator consists of the rest of the terms.

The problem is that the electronic wavefunction can not be solved analytically when two or more electrons are present in the system due to the electron-electron repulsion (the fourth term in Eq. 2.2).

2.1.1 Density Functional Theory

To solve the SE for larger systems, further approximations are necessary. Density Functional Theory (DFT) is the most extensively used approach. DFT is based on the fact that the electrons can be described exactly and uniquely as an electron probability density, ρ_e, instead of an electronic wavefunction[56, 57]. Hence, the electrons depend only on 3 Cartesian coordinates instead of $3N_e$ Cartesian coordinates. The electronic energy can also be expressed as a functional of the electron density as:

$$E[\rho_e] = T_e[\rho_e] + V_{ee}[\rho_e] + \int \rho_e(\vec{r})v_{Ne}(\vec{r};\{\vec{R}\})\mathrm{d}\vec{r} \tag{2.3}$$

where T_e is the kinetic energy of the electrons, V_{ee} is the electron-electron repulsion energy, and $v_{Ne}(\vec{r};\{\vec{R}\}) = \frac{-q_e^2}{4\pi\epsilon_0}\sum_{i=1}^{N_N}\frac{Z_i}{|\vec{R}_i-\vec{r}|}$ is the nuclei-electrons attractive potential. The energy dependence of the electronic density function gives the name of Density Functional Theory.

However, the electron density can not be obtained from the SE due to the electronic kinetic energy and the electron-electron repulsion energy terms. Therefore, the Kohn-Sham (KS) orbitals, ψ_i, are introduced for each electron to calculate the kinetic energy term and to separate the electronic wavefunction into one-electron wavefunctions. The electron density can be expressed from the one-electron orbitals as:

$$\rho_e(\vec{r};\{\vec{R}\}) = \sum_{i=1}^{N_e} |\psi_i(\vec{r};\{\vec{R}\})|^2 \tag{2.4}$$

The KS orbitals are non-interacting orbitals solved from the KS equation[58]. The KS equation is expressed as:

$$\left(\frac{-\hbar^2}{2m_e}\hat{\nabla}_i^2 + \frac{q_e^2}{4\pi\epsilon_0}\int_{-\infty}^{\infty}\frac{\rho_e(\vec{\tilde{r}})}{|\vec{r}-\vec{\tilde{r}}|}\mathrm{d}\vec{\tilde{r}} + v_{Ne}(\vec{r};\{\vec{R}\}) + v_{XC}(\vec{r})\right)\psi_i\left(\vec{r};\{\vec{R}\}\right) = \varepsilon_i\psi_i\left(\vec{r};\{\vec{R}\}\right) \tag{2.5}$$

where ε_i is the eigenvalue of the ith KS orbital and $v_{XC} = \frac{\delta E_{XC}[\rho_e]}{\delta\rho_e(\vec{r})}$ is the Exchange-correlation (XC) potential or the functional derivative of the XC energy, E_{XC}, wrt. the electron density. Hence, the energy can be expressed as:

$$E[\rho_e] = T_s[\rho_e] + E_J[\rho_e] + \int \rho_e(\vec{r})v_{Ne}(\vec{r};\{\vec{R}\})\mathrm{d}\vec{r} + E_{XC}[\rho_e] \tag{2.6}$$

where $E_J = \frac{q_e^2}{8\pi\epsilon_0}\int_{-\infty}^{\infty}\frac{\rho_e(\vec{r})\rho_e(\vec{\tilde{r}})}{|\vec{r}-\vec{\tilde{r}}|}\mathrm{d}\vec{\tilde{r}}\mathrm{d}\vec{r}$ is the classical electron-electron repulsion energy and T_s is the kinetic energy of the non-interacting electrons. The XC energy, $E_{XC} = (T_e - T_s) + (V_{ee} - E_J)$, is the correction to the kinetic energy and non-classical electron-electron repulsion energy for interacting electrons. The energy with the same XC potential follows the variational principle, and it is therefore minimized self-consistently since the classical electron-electron repulsion, and the XC terms depend on the electron density in Eq. 2.5. Eq. 2.6 is, in principle, the exact energy. However, the correct XC functional is unknown.

2.1.2 Exchange-Correlation Functionals

Due to the unknown form of the exact XC functional, various approximate forms exist. Increasingly complex assumptions are made, with the expectation of higher accuracy, but also an increasing computational cost. The XC functionals are categorized with increasing complexity as Local Density Approximation functional (LDA), Generalized Gradient Approximation functional (GGA), Meta Generalized Gradient Approximation functional (mGGA), and hybrid functionals. The LDA assumes a uniform electronic density[59, 60]. The GGA uses the first-order derivative of the electronic density. Examples of GGAs are BLYP[61, 62] and PBE[63]. The mGGA uses higher-order derivatives of the electronic density. Hybrid functionals[64] use fractions of the exchange from Hartree-Fock theory[65, 66, 67, 68], where the KS orbitals are used.

2.2 Nudged Elastic Band method

The NEB is the standard method for obtaining the activation energies and MEPs for catalysis simulations[22, 23]. A transition path from the initial state to the final state is constructed by a series of undergoing structures with coordinates $\{\vec{R}_i\}$. Those structures are called images or replicas. The images are connected with spring interactions. Therefore, the path is presented as an elastic band. To obtain the MEP, the images are moved accordingly to their forces. The forces of an image i, $\vec{F}_i$, is the sum of the spring forces along the tangent of the path, $\vec{F}_{i\|}$, and the true force from the energy, $\hat{\nabla}_i E_i$, perpendicular to the tangent of the path, $\vec{F}_{i\perp}$, expressed as:

$$\vec{F}_i = \vec{F}_{i\|} + \vec{F}_{i\perp} \tag{2.7}$$

The forces perpendicular to the tangent of the path, $\vec{\tau}_i$, are expressed as:

$$\vec{F}_{i\perp} = \hat{\nabla}_i E_i - \hat{\nabla}_i E_i \cdot \frac{\vec{\tau}_i}{|\vec{\tau}_i|} \tag{2.8}$$

The spring forces of image i along the path with the improved tangent method[22] is:

$$\vec{F}_{i\|} = k_s \left(|\vec{R}_{i+1} - \vec{R}_i| - |\vec{R}_i - \vec{R}_{i-1}| \right) \frac{\vec{\tau}_i}{|\vec{\tau}_i|} \tag{2.9}$$

where k_s is the spring constant. The improved tangent of the path at image i is expressed as:

$$\vec{\tau}_i = \begin{cases} \vec{R}_{i+1} - \vec{R}_i & \text{if } E_{i+1} > E_i > E_{i-1} \\ \vec{R}_i - \vec{R}_{i-1} & \text{if } E_{i+1} < E_i < E_{i-1} \\ \left(\vec{R}_{i+1} - \vec{R}_i\right) |E_{i+1} - E_i| + \left(\vec{R}_i - \vec{R}_{i-1}\right) |E_i - E_{i-1}| & \text{if } E_{i+1} > E_{i-1} > E_i \\ \left(\vec{R}_{i+1} - \vec{R}_i\right) |E_i - E_{i-1}| + \left(\vec{R}_i - \vec{R}_{i-1}\right) |E_{i+1} - E_i| & \text{if } E_i > E_{i+1} > E_{i-1} \\ \left(\vec{R}_{i+1} - \vec{R}_i\right) |E_i - E_{i-1}| + \left(\vec{R}_i - \vec{R}_{i-1}\right) |E_{i+1} - E_i| & \text{if } E_{i+1} < E_{i-1} < E_i \\ \left(\vec{R}_{i+1} - \vec{R}_i\right) |E_{i+1} - E_i| + \left(\vec{R}_i - \vec{R}_{i-1}\right) |E_i - E_{i-1}| & \text{if } E_i < E_{i+1} < E_{i-1} \end{cases} \tag{2.10}$$

The Climbing Image Nudged Elastic Band method (CI-NEB)[24] is an extension of the NEB which can improve the accuracy of the activation energy. The CI-NEB releases the image j with the largest energy from the spring interactions. The image j is then moved towards the SP with the forces:

$$\vec{F}_j = -\hat{\nabla}_j E_j + 2\hat{\nabla}_j E_j \cdot \frac{\vec{\tau}_j}{|\vec{\tau}_j|} \cdot \frac{\vec{\tau}_j}{|\vec{\tau}_j|} \tag{2.11}$$

2.3 Gaussian Process Regression

A GP is a multivariate Gaussian distribution, $\mathcal{N}$, for a collection of random variables[49, 53, 69]. The collection of random variables, $\vec{f}(\mathbf{x})$, can be sampled from the multivariate normal distribution as:

$$\vec{f}(\mathbf{X}) \sim \mathcal{N}\left(\vec{m}(\mathbf{X}), \mathbf{\Sigma}\right) = \frac{\exp\left(\frac{-1}{2}(\vec{f}(\mathbf{X}) - \vec{m}(\mathbf{X}))^{\mathrm{T}} \mathbf{\Sigma}^{-1} (\vec{f}(\mathbf{X}) - \vec{m}(\mathbf{X}))\right)}{\sqrt{(2\pi)^{N_v} |\mathbf{\Sigma}|}} \tag{2.12}$$

where $\vec{m}$ is the prior mean functions, $\mathbf{\Sigma}$ is the covariance matrix, $|\mathbf{\Sigma}|$ is the determinant of the covariance matrix, and N_v is the number of variables.

The collected random variables can be split into the training targets, $\vec{f}$, with the training features, $\mathbf{X}$, and the test targets, $\vec{f}_*$, with the test features, $\mathbf{X}_*$. The collected training targets are a column vector with N elements, and the training features have the dimensions $N \times D$ with D as the number of descriptor elements or coordinates. The collected test targets are also expressed as a column vector with the size of M, and the test features have the dimensions of $M \times D$. Equivalent to Eq. 2.12, the collected training and test targets can be sampled from the joint posterior distribution, $p(\vec{f}, \vec{f}_* \mid \mathbf{X}, \mathbf{X}_*, \vec{\theta})$, as:

$$\begin{bmatrix} \vec{f}(\mathbf{X}) \\ \vec{f}_*(\mathbf{X}_*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \vec{\mu}(\mathbf{X}) \\ \vec{\mu}_*(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) & \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}) & \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix} \right) = p(\vec{f}, \vec{f}_* \mid \mathbf{X}, \mathbf{X}_*, \vec{\theta}) \tag{2.13}$$

where $\mathbf{K}$ are subset matrices of $\mathbf{\Sigma}$, $\vec{\mu}$ is the prior mean of the training targets, $\vec{\mu}_*$ is the prior mean of the test targets, and $\vec{\theta}$ is a set of hyperparameters that the covariance matrices depend on. The covariance matrix elements are covariance function values (see Section 2.3.1). The covariance functions are usually functions of the distances between the features to describe the correlations of the targets.

In real data, the training and test targets frequently include some noise, $y(\vec{x}) = f(\vec{x}) + \varepsilon_n$. The noise can be assumed to be independently Gaussian distributed, $\varepsilon_n \sim \mathcal{N}(0, \sigma_n^2)$, with a noise variance of σ_n^2. Thereby, the noisy targets can be expressed from the marginalized joint posterior distribution over the underlying targets:

$$\begin{bmatrix} \vec{y}(\mathbf{X}) \\ \vec{y}_*(\mathbf{X}_*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \vec{\mu}(\mathbf{X}) \\ \vec{\mu}_*(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I} & \mathbf{K}(\mathbf{X},\mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*,\mathbf{X}) & \mathbf{K}(\mathbf{X}_*,\mathbf{X}_*) + \sigma_n^2\mathbf{I} \end{bmatrix} \right) = p(\vec{y}, \vec{y}_* | \mathbf{X}, \mathbf{X}_*, \vec{\theta}) \tag{2.14}$$

Now, the new covariance matrices include the noise variance at the diagonal elements.

A single noisy test target can be sampled from the conditional distribution of the GP as:

$$y_* \mid \vec{x}_*, \vec{y}, \mathbf{X} \sim \mathcal{N}\left(\bar{y}_*(\vec{x}_*), \sigma_*^2(\vec{x}_*)\right) = p(y_* \mid \vec{x}_*, \vec{y}, \mathbf{X}, \vec{\theta}) \tag{2.15}$$

where $\bar{y}_*$ is the predictive mean and σ_*^2 is the predictive variance. The predictive mean and variance are:

$$\mathrm{E}[y_*] = \bar{y}_*(\vec{x}_*) = \mu_*(\vec{x}_*) + \mathbf{K}(\vec{x}_*, \mathbf{X})\mathbf{C}^{-1}\left(\vec{y}(\mathbf{X}) - \vec{\mu}(\mathbf{X})\right) \tag{2.16}$$

$$\mathrm{var}[y_*] = \sigma_*^2(\vec{x}_*) = k(\vec{x}_*, \vec{x}_*) + \sigma_n^2 - \mathbf{K}(\vec{x}_*, \mathbf{X})\mathbf{C}^{-1}\mathbf{K}(\vec{x}_*, \mathbf{X})^\top \tag{2.17}$$

where the covariance matrix with noise is:

$$\mathbf{C} = \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I} \tag{2.18}$$

The predictive mean is the best prediction of the test target given the training data. The prediction uncertainty is the square root of the predictive variance.

The predictive mean can be treated as a linear combination of basis functions:

$$\bar{y}_*(\vec{x}_*) = \mu_*(\vec{x}_*) + \mathbf{K}(\vec{x}_*, \mathbf{X})\vec{c} = \mu_*(\vec{x}_*) + \sum_{i=1}^{N} k(\vec{x}_*, \vec{x}_i)c_i \tag{2.19}$$

where N is the number of training data and $\vec{c}$ corresponds to the coefficients. The optimal expression of the coefficients is:

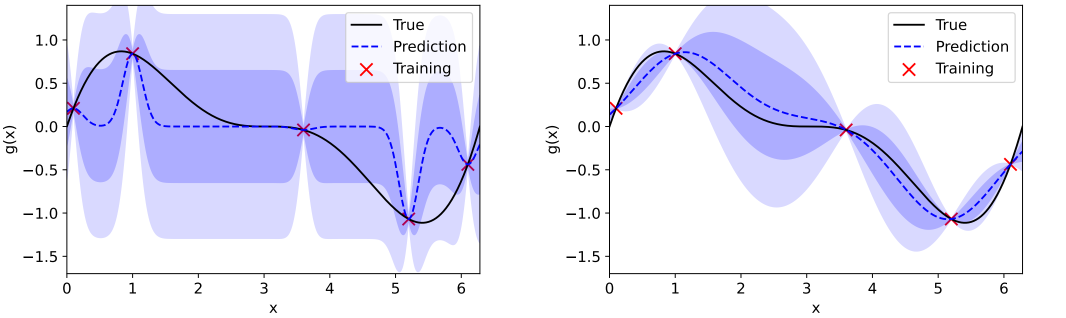
$$\vec{c} = \mathbf{C}^{-1}\left(\vec{y}(\mathbf{X}) - \vec{\mu}(\mathbf{X})\right) \tag{2.20}$$

2.3.1 Kernels

A frequently used covariance function is the Squared Exponential Covariance (SEC) function. The SEC function is the exponential of the squared Euclidean distance between two points scaled with a length-scale hyperparameter, l, defined as:

$$k_{SEC}(\vec{x}_p, \vec{x}_q) = \alpha^2 \exp\left(\frac{-(\vec{x}_p - \vec{x}_q)^{\mathrm{T}}(\vec{x}_p - \vec{x}_q)}{2l^2} \right) \tag{2.21}$$

where α is the prefactor hyperparameter. The prefactor hyperparameter controls the magnitude of the covariance matrix. The SEC function can also be treated as a Gaussian function with $\vec{x}_q$ as the mean. The length-scale hyperparameter determines the broadness of the Gaussian functions. The broadness affects the flexibility of the prediction (see Fig. 2.1).



(a) Short length-scale hyperparameter

(b) Optimized length-scale hyperparameter

Figure 2.1: Two Gaussian process predictions of a simple function, $g(x) = 0.25 \sin(x)(x-3)^2$, from five training points. The linear combination of Gaussian functions is clearly seen in figure (a). The dark blue areas show the uncertainty predictions and the light blue areas show two times the uncertainty predictions from the Gaussian processes.

Multiple length-scale hyperparameters can also be used in the SEC function. Then, different length-scale hyperparameters are used for each dimension:

$$k_{MLSEC}(\vec{x}_p, \vec{x}_q) = \alpha^2 \exp\left(\sum_{d=1}^{D} \frac{-(x_{p,d} - x_{q,d})^2}{2l_d^2}\right) \tag{2.22}$$

Different length-scale hyperparameters are important if the feature elements vary with different magnitudes. They can also be applied as an automatic relevance determination.

2.3.2 Derivatives of the targets

The derivatives of the targets can also be implemented to improve the predictions[42]. Hence, the derivatives (or negative forces for energies) can also be predicted together with their uncertainties. Then, the training targets are extended with the derivatives of each target wrt. each feature coordinate as:

$$\vec{y}_{ext}(\mathbf{X}) = [\vec{y}^{\mathrm{T}}, \frac{\partial \vec{y}^{\mathrm{T}}}{\partial x_1}, \cdots, \frac{\partial \vec{y}^{\mathrm{T}}}{\partial x_D}]^{\mathrm{T}} \tag{2.23}$$

The covariance matrix must also be extended, $\mathbf{K}_{ext}$, if the derivatives of the training targets are applied. The covariance matrix is extended with its first and second-order derivatives wrt. each features element:

$$\mathbf{K}_{ext}(\mathbf{X}, \mathbf{X}') = \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}') & \frac{\partial \mathbf{K}(\mathbf{X},\mathbf{X}')}{\partial x_1'} & \cdots & \frac{\partial \mathbf{K}(\mathbf{X},\mathbf{X}')}{\partial x_D'} \\ \frac{\partial \mathbf{K}(\mathbf{X},\mathbf{X}')}{\partial x_1} & \frac{\partial^2 \mathbf{K}(\mathbf{X},\mathbf{X}')}{\partial x_1 \partial x_1'} & \cdots & \frac{\partial^2 \mathbf{K}(\mathbf{X},\mathbf{X}')}{\partial x_1 \partial x_D'} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{K}(\mathbf{X},\mathbf{X}')}{\partial x_D} & \frac{\partial^2 \mathbf{K}(\mathbf{X},\mathbf{X}')}{\partial x_D \partial x_1'} & \cdots & \frac{\partial^2 \mathbf{K}(\mathbf{X},\mathbf{X}')}{\partial x_D \partial x_D'} \end{bmatrix} \tag{2.24}$$

The first and second-order derivatives of the SEC function wrt. the features are:

$$\frac{\partial k_{SEC}(\vec{x}_p, \vec{x}_q)}{\partial x_{q,d}} = k_{SEC}(\vec{x}_p, \vec{x}_q) \frac{(x_{p,d} - x_{q,d})}{l^2} \tag{2.25}$$

$$\frac{\partial^2 k_{SEC}(\vec{x}_p, \vec{x}_q)}{\partial x_{p,d_1} \partial x_{q,d_2}} = k_{SEC}(\vec{x}_p, \vec{x}_q) \left(\frac{\delta_{d_1 d_2}}{l^2} - \frac{(x_{p,d_1} - x_{q,d_1})(x_{p,d_2} - x_{q,d_2})}{l^4}\right) \tag{2.26}$$

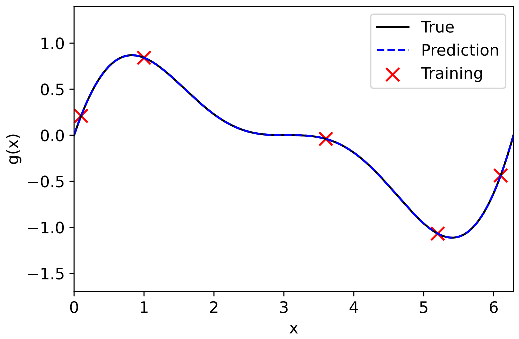


Figure 2.2: A Gaussian process prediction of a simple function, $g(x) = 0.25\sin(x)(x-3)^2$, from five training points with derivatives. The light blue areas show small uncertainty predictions from the Gaussian process. The prediction mean is a perfect fit with low uncertainty to the simple function when derivatives are included.

Exploiting the derivatives of the targets significantly improves the prediction mean since it gets the correct derivatives (see Fig. 2.2). Besides the information from the derivatives, the prediction mean also has more flexibility from an additional term as:

$$\bar{y}_*(\vec{x}_*) = \mu_*(\vec{x}_*) + \sum_{i=1}^{N} k(\vec{x}_*, \vec{x}_i)c_i + \sum_{i=1}^{N}\sum_{d=1}^{D} \frac{\partial k(\vec{x}_*, \vec{x}_i)}{\partial x_{i,d}} c_{d\cdot N+i} \tag{2.27}$$

Unfortunately, the computational complexity increases from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^3(1+D)^3)$ when the derivatives are used.

2.3.3 Hyperparameters

The performance of the GP strictly depends on its hyperparameters. The prior mean hyperparameter can be dependent on the features, but often the zero prior, $\mu = 0$, or the mean of the training targets are used, $\mu = \frac{1}{N}\sum_{i=1}^{N} y_i(\vec{x}_i)$. The prediction mean becomes the prior mean when the test feature is far from the training features.

The prefactor hyperparameter is a general part of a covariance function. The relation between the prefactor and noise hyperparameter affects the regularization. Hence, a new free hyperparameter is introduced as the relative-noise hyperparameter expressed as:

$$\sigma_r \equiv \frac{\sigma_n}{\alpha} \tag{2.28}$$

The relative-noise hyperparameter, σ_r, replaces the noise hyperparameter to decouple them. Therefore, the covariance matrices can be factorized as $\mathbf{K} = \alpha^2\mathbf{K}_0$ and $\mathbf{C} = \alpha^2\mathbf{C}_0$. As a consequence, it can be observed that the prediction mean (Eq. 2.16) is independent of the prefactor hyperparameter. Furthermore, the prediction variance is proportional to the squared prefactor hyperparameter, $\sigma_*^2 = \alpha^2\sigma_{*0}^2$. The σ_{*0}^2 is expressed as:

$$\sigma_{*0}^2 = k_0(\vec{x}_*, \vec{x}_*) + \sigma_r^2 - \mathbf{K}_0(\vec{x}_*, \mathbf{X})\mathbf{C}_0^{-1}\mathbf{K}_0(\vec{x}_*, \mathbf{X})^\top \tag{2.29}$$

Thereby, the prefactor hyperparameter only influences the magnitude of the uncertainty predictions.

The relative-noise hyperparameter has multiple purposes:

1. It makes the covariance matrix of the training features invertible.

2. It works as a regularization.
3. It identifies the noise-to-signal of the targets.

Further hyperparameters originate from the chosen covariance function. The SEC function has one or multiple length-scale hyperparameters. As previously mentioned, the length-scale hyperparameter recognizes the flexibility of the targets.

The posterior distribution of the hyperparameters, $\vec{\theta}$, given the training features and targets are expressed from Bayes' theorem as:

$$p(\vec{\theta} \mid \vec{y}, \mathbf{X}) = \frac{p(\vec{y} \mid \vec{\theta}, \mathbf{X}) p(\vec{\theta})}{p(\vec{y} \mid \mathbf{X})} \tag{2.30}$$

where the marginal likelihood, $p(\vec{y} \mid \mathbf{X})$, is a normalization constant. Different objective functions are used to optimize the hyperparameters, but the most common is the Log-Likelihood (LL). Under the approximation that uniform prior distributions can be used as the prior of the hyperparameters, $p(\vec{\theta}) = 1$, the unnormalized posterior distribution of the hyperparameters becomes the likelihood of the targets given the hyperparameters, $p(\vec{y}|\vec{\theta}, \mathbf{X})$. The likelihood has the same expression as in Eq. 2.12, but only with the training targets. The LL is expressed as:

$$LL \equiv \frac{-1}{2\alpha^2} (\vec{y} - \vec{\mu})^\top \mathbf{C}_0^{-1} (\vec{y} - \vec{\mu}) - \frac{1}{2} \ln\left(|\mathbf{C}_0|\right) - \frac{N}{2} \ln\left(\alpha^2\right) - \frac{N}{2} \ln\left(2\pi\right) \tag{2.31}$$

Often, the Cholesky factorization[70] matrix, $\mathbf{L}\mathbf{L}^\mathrm{T} = \mathbf{C}_0$, is used. The Cholesky factorization is faster than the inversion of the covariance matrix, which is the rate-determining step with its computational complexity of $\mathcal{O}(N^3)$. The coefficients are obtained using back and forward substitution of the Cholesky factorization matrix and the training target. The term with the determinant of the covariance matrix in Eq. 2.31 can also be expressed with the Cholesky factorization, $\ln\left(|\mathbf{C}_0|\right) = 2\sum_{i=1}^{N} \ln\left(L_{ii}\right)$.

Usually, the LL is maximized, corresponding to the point estimate or Maximum Likelihood Estimation (MLE). The analytical expression of prefactor hyperparameter from maximization of the LL is[69, 71]:

$$\alpha^2_{\mathrm{MLE}} = \frac{1}{N} (\vec{y} - \vec{\mu})^\top \mathbf{C}_0^{-1} (\vec{y} - \vec{\mu}) \tag{2.32}$$

The LL expression with the maximized prefactor hyperparameter, LL_{MLE}, is:

$$\begin{aligned} LL_{MLE} =& \frac{-N}{2}\left(1 + \ln\left(2\pi\right)\right) - \frac{1}{2}\ln\left(|\mathbf{C}_0|\right) - \frac{N}{2}\ln\left(\frac{1}{N}(\vec{y} - \vec{\mu})^\top \mathbf{C}_0^{-1}(\vec{y} - \vec{\mu})\right) \\ =& \frac{-N}{2}\left(1 + \ln\left(2\pi\right)\right) - \frac{1}{2}\sum_{i=1}^{N}\ln\left([\mathbf{\Lambda}]_{ii} + \sigma_r^2\right) - \frac{N}{2}\ln\left(\frac{1}{N}\sum_{i=1}^{N}\frac{[\mathbf{U}^\top(\vec{y} - \vec{\mu})]_i^2}{[\mathbf{\Lambda}]_{ii} + \sigma_r^2}\right) \end{aligned} \tag{2.33}$$

where the eigendecomposition, $\mathbf{K}_0 = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$. A single eigendecomposition is enough to search after all values of the relative-noise hyperparameter for a given length-scale hyperparameter value.

However, the prior distributions are crucial in Bayes' theorem. The Log-Posterior (LP) of the hyperparameter is easy to calculate when the LL is obtained. The logarithm of the prior distribution is just added to LL as:

$$LP = LL + \sum_{\theta_i} \ln\left(p(\theta_i)\right) \tag{2.34}$$

Thus, the Maximum A Posteriori estimation (MAP) can be obtained in the same way as the MLE.

2.4 T Process Regression

The prefactor hyperparameter can be marginalized by a Bayesian approach instead of maximizing the LL.

The likelihood of the training targets from a GP can be rewritten as an inverse-gamma distribution for the prefactor hyperparameter[49]:

$$p(\vec{y} \mid \mathbf{X}, \alpha^2, \sigma_r, l) = (\alpha^2)^{-N/2} \frac{\exp\left(\frac{-1}{2}\frac{1}{\alpha^2}(\vec{y}-\vec{\mu})^{\mathrm{T}}\boldsymbol{C}_0^{-1}(\vec{y}-\vec{\mu})\right)}{\sqrt{(2\pi)^N|\boldsymbol{C}_0|}} \tag{2.35}$$

The prior distribution of the prefactor is also chosen to be an inverse-gamma distribution:

$$p(\alpha^2) = \mathrm{Ga}^{-1}(\alpha^2|a,b) = \frac{b^a}{\Gamma(a)}(\alpha^2)^{-a-1}\exp\left(-\frac{b}{\alpha^2}\right) \tag{2.36}$$

The hyperprior parameters a and b are chosen to be $a = b = 1.0 \cdot 10^{-20}$ in this work. Then, the prior distribution is weakly informative in the logarithmic space, but it avoids prefactor values below machine precision.

The likelihood of the training targets without the prefactor hyperparameter is now obtainable as a multivariate Student's t distribution:

$$\begin{aligned}
p(\vec{y} \mid \mathbf{X}, \sigma_r, l) = & \int_0^\infty p(\vec{y} \mid \mathbf{X}, \alpha^2, \sigma_r, l)p(\alpha^2)\,\mathrm{d}\alpha^2 \\
= & \frac{\Gamma(\frac{2a+N}{2})}{\Gamma(a)}\frac{1}{\sqrt{(2\pi b)^N|\boldsymbol{C}_0|}}\left(1+\frac{1}{2b}(\vec{y}-\vec{\mu})^{\mathrm{T}}\boldsymbol{C}_0^{-1}(\vec{y}-\vec{\mu})\right)^{-a-\frac{N}{2}} \\
= & t_{\nu=2a}\left(\vec{y} \mid \vec{\mu}, \frac{b}{a}\mathbf{C}_0\right)
\end{aligned} \tag{2.37}$$

Then, the posterior predictive distribution of a test target without the prefactor can be derived as follows:

$$\begin{aligned}
p(y_* \mid \vec{x}_*, \vec{y}, \mathbf{X}, \sigma_r, l) = & \int_0^\infty p(y_* \mid \vec{x}_*, \vec{y}, \mathbf{X}, \alpha^2, \sigma_r, l)\frac{p(\vec{y} \mid \mathbf{X}, \alpha^2, \sigma_r, l)p(\alpha^2)}{p(\vec{y} \mid \mathbf{X}, \sigma_r, l)}\,\mathrm{d}\alpha^2 \\
= & \frac{\Gamma(\frac{2a_N+1}{2})}{\Gamma(a_N)\sqrt{b_N 2\pi\sigma_{*0}^2}}\left(1+\frac{1}{2b_N}\frac{1}{\sigma_{*0}^2}|y_*-\bar{y}_*|^2\right)^{-\frac{2a_N+1}{2}} \\
= & t_{\nu_*=2a+N}\left(y_* \mid \bar{y}_*, \frac{b_N}{a_N}\sigma_{*0}^2\right)
\end{aligned} \tag{2.38}$$

where a_N and b_N is:

$$a_N \equiv \frac{2a+N}{2} \tag{2.39}$$

$$b_N \equiv b + \frac{1}{2}(\vec{y}-\vec{\mu})^{\mathrm{T}}\boldsymbol{C}_0^{-1}(\vec{y}-\vec{\mu}) \tag{2.40}$$

Thus, $p(y_*|\vec{x}_*, \vec{y}, \mathbf{X}, \sigma_r, l)$ is a Student's T Process (TP). The TP has the same prediction mean as the GP and a prediction variance that is scaled compared to the prediction variance of the GP as:

$$\mathrm{E}[y_*] = \bar{y}_* = \mu_*(\vec{x}_*) + \mathbf{K}_0(\vec{x}_*, \mathbf{X})\mathbf{C}_0^{-1}\left(\vec{y}(\mathbf{X}) - \vec{\mu}(\mathbf{X})\right) \tag{2.41}$$

$$\mathrm{var}[y_*] = \frac{2b + (\vec{y}-\vec{\mu})^\top\mathbf{C}_0^{-1}(\vec{y}-\vec{\mu})}{2a+N-2}\sigma_{0*}^2 \tag{2.42}$$

The scaling of the variance is closely related to the analytical solution of the prefactor hyperparameter in Eq. 2.32.

The rest of the hyperparameters (relative-noise and length-scale) can be found by using the posterior distribution of the hyperparameters:

$$p(\sigma_r, l|\vec{y}, \mathbf{X}) \propto p(\vec{y}|\sigma_r, l, \mathbf{X})p(\sigma_r, l) \tag{2.43}$$

The prior of the hyperparameters can be chosen to be uniform distributions like for the GP. The corresponding LL for the TP is:

$$\begin{aligned} LL =& \ln\left(p(\vec{y}|\mathbf{X}, \sigma_r, l)\right) \\ \propto& -\frac{1}{2}\ln\left(|\mathbf{C}_0|\right) - \frac{2a+N}{2}\ln\left(1 + \frac{1}{2b}(\vec{y}-\vec{\mu})^{\mathrm{T}}\mathbf{C}_0^{-1}(\vec{y}-\vec{\mu})\right) \end{aligned} \tag{2.44}$$

2.5 Fully Bayesian Mimicking Gaussian Process

The posterior predictive distribution of a target marginalized over the hyperparameters can be obtained with numerical integration[72, 73]. However, the numerical process must then be repeated for every new prediction. The use of information theory in the form of the Kullback–Leibler divergence (KL) [74] can be generalized to find the hyperparameters that give the solution closest to the fully Bayesian solution.

Initially, a grid in the hyperparameter space is constructed. A hyperparameter set is denoted as $\vec{\theta} = (l, \sigma_r, \alpha)$. Furthermore, a grid coordinate of those hyperparameters is defined as $[\mathbf{\Theta}]_{ijr} = ([l]_i, [\alpha]_j, [\sigma_r]_r)$. The posterior predictive distribution of the test target given only the training targets can be approximated as:

$$\begin{aligned} p(y_* \mid \vec{y}) =& \frac{1}{p(\vec{y})}\int_{-\infty}^{\infty} p(y_* \mid \vec{y}, \vec{\theta})p(\vec{y} \mid \vec{\theta})p(\vec{\theta})\mathrm{d}\vec{\theta} \\ \approx& \frac{1}{N_c}\sum_{i=1}^{G_l}\sum_{j=1}^{G_\alpha}\sum_{r=1}^{G_\sigma} p(y_* \mid \vec{y}, [\mathbf{\Theta}]_{ijr})\tilde{c}([\mathbf{\Theta}]_{ijr}, \vec{y}) \\ =& \frac{1}{N_c}\sum_{i,j,r}\tilde{c}([\mathbf{\Theta}]_{ijr}, \vec{y})\mathcal{N}(y_*|\bar{y}_*([\mathbf{\Theta}]_{ijr}), \sigma_*^2([\mathbf{\Theta}]_{ijr})) \end{aligned} \tag{2.45}$$

where the trapezoidal rule is applied, the given features are removed from the notation for clarity, G is the last index in the grid of one of the hyperparameters, $\tilde{c}([\mathbf{\Theta}]_{ijr}, \vec{y})$ is the defined adapted trapezoidal coefficient, and N_c is the approximated marginal likelihood. The notations are defined as:

$$\tilde{c}([\mathbf{\Theta}]_{ijr}, \vec{y}) \equiv p(\vec{y} \mid [\mathbf{\Theta}]_{ijr})p([\mathbf{\Theta}]_{ijr})c([l]_i)c([\alpha]_j])c([\sigma_r]_r) \tag{2.46}$$

$$c(\theta_i) \equiv \begin{cases} \frac{\theta_{(i+1)}-\theta_{(i-1)}}{2} & \text{if } 1 < i < G \\ \frac{\theta_2-\theta_1}{2} & \text{if } i = 1 \\ \frac{\theta_G-\theta_{(G-1)}}{2} & \text{if } i = G \end{cases} \tag{2.47}$$

$$N_c \equiv \sum_{i,j,r}\tilde{c}([\mathbf{\Theta}]_{ijr}, \vec{y}) \approx p(\vec{y}|\mathbf{X}) \tag{2.48}$$

where $c(\theta_i)$ is the defined trapezoidal coefficient.

It is assumed a single GP, $p(y_* \mid \vec{y}, \vec{\theta}_0)$, exists with a set of hyperparameters, $\vec{\theta}_0$, that can be a good approximation to the fully Bayesian solution, $p(y_* \mid \vec{y})$. The MLE is often assumed to be a good approximation to the fully Bayesian solution though it is only the case when a large amount of training data is used. Here, the best approximation to the fully Bayesian solution is calculated by KL, D_{FB}, as:

$$D_{FB} \propto \frac{\sum_{i,j,r} \tilde{c}([\mathbf{\Theta}]_{ijr},\vec{y}) \left(\sigma_*^2([\mathbf{\Theta}]_{ijr}) + \left(\bar{y}_*([\mathbf{\Theta}]_{ijr}) - \bar{y}_*(\vec{\theta_0})\right)^2\right)}{2N_c\sigma_*^2(\vec{\theta_0})} + \frac{1}{2}\ln\left(2\pi\sigma_*^2(\vec{\theta_0})\right) \tag{2.49}$$

The complete derivation of Eq. 2.49 is seen in Section A.1. The weighted averages of the prediction mean and uncertainty with the LP values at the grid points can be used to store fewer variables:

$$D_{FB} = \frac{\overline{\sigma_*^2} + \overline{\bar{y}_*^2} + \bar{y}_*^2(\vec{\theta_0}) - 2\bar{y}_*(\vec{\theta_0})\overline{\bar{y}_*}}{2\alpha_0^2\sigma_{*0}^2(\vec{\theta_0})} + \frac{1}{2}\ln\left(2\pi\sigma_{*0}^2(\vec{\theta_0})\right) + \frac{1}{2}\ln\left(\alpha_0^2\right) \tag{2.50}$$

where the weighted averages and the analytical solution for the prefactor hyperparameter, α_0^2, are expressed as:

$$\overline{\sigma_*^2} = \frac{1}{N_c}\sum_{i,j,r} \tilde{c}([\mathbf{\Theta}]_{ijr},\vec{y})\sigma_*^2([\mathbf{\Theta}]_{ijr}) \tag{2.51}$$

$$\overline{\bar{y}_*^2} = \frac{1}{N_c}\sum_{i,j,r} \tilde{c}([\mathbf{\Theta}]_{ijr},\vec{y})\bar{y}_*^2([\mathbf{\Theta}]_{ijr}) \tag{2.52}$$

$$\overline{\bar{y}_*} = \frac{1}{N_c}\sum_{i,j,r} \tilde{c}([\mathbf{\Theta}]_{ijr},\vec{y})\bar{y}_*([\mathbf{\Theta}]_{ijr}) \tag{2.53}$$

$$\alpha_0^2 = \frac{\overline{\sigma_*^2} + \overline{\bar{y}_*^2} + \bar{y}_*^2(\vec{\theta_0}) - 2\bar{y}_*(\vec{\theta_0})\overline{\bar{y}_*}}{\sigma_{*0}^2(\vec{\theta_0})} \tag{2.54}$$

The analytical solution for the prefactor hyperparameter can be inserted into Eq. 2.50 to give:

$$D_{FB} = \frac{1}{2} + \frac{1}{2}\ln\left(2\pi\right)) + \frac{1}{2}\ln\left(\sigma_{*0}^2(\vec{\theta_0})\right) + \frac{1}{2}\ln\left(\alpha_0^2\right) \tag{2.55}$$

Thus, the hyperparameter set of the GP that mimics the fully Bayesian solution best is found from a grid by minimizing Eq. 2.55. However, it requires a validation target to compare its predictions. Fortunately, the predictions of the validation target are just compared, and the feature, not the target, is only required.

Multiple validation targets give a better estimate of the Fully Bayesian Mimicking Gaussian Process (FBMGP). The D_{FB} for N_t validation point is simply a sum of D_{FB} for each validation feature:

$$D_{FB} = \frac{1}{2}\left(N_t + N_t\ln\left(2\pi\right) + \sum_{t=1}^{N_t}\ln\left(\sigma_{*0t}^2(\vec{\theta_0})\right) + N_t\ln\left(\alpha_0^2\right)\right) \tag{2.56}$$

where the analytic solution to the prefactor hyperparameter is now expressed as:

$$\alpha_0^2 = \frac{1}{N_t}\sum_{t=1}^{N_t}\frac{\overline{\sigma_{*t}^2} + \overline{\bar{y}_{*t}^2} + \bar{y}_{*t}^2(\vec{\theta_0}) - 2\bar{y}_{*t}(\vec{\theta_0})\overline{\bar{y}_{*t}}}{\sigma_{*0t}^2(\vec{\theta_0})} \tag{2.57}$$

The validation features can be sampled between the training features.

The trick to avoiding numeric under- or overflow in the weighted averages is to subtract the LP with the greatest observed value, LP_{max}, at the time. E.g. can Eq. 2.53 be rewritten as:

$$\overline{\bar{y}_*} = = \frac{\sum_{i,j,r} \exp\left(\ln\left(\tilde{c}([\boldsymbol{\Theta}]_{ijr}, \vec{y})\right) - LP_{max}\right)\bar{y}_*([\boldsymbol{\Theta}]_{ijr})}{\sum_{i,j,r} \exp\left(\ln\left(\tilde{c}([\boldsymbol{\Theta}]_{ijr}, \vec{y})\right) - LP_{max}\right)} \tag{2.58}$$

3 Optimization of hyperparameters

3.1 Introduction

The hyperparameters of the GP (described in Section 2.3.3) are essential for the performance of the GP. Usually, the hyperparameters are optimized by a local optimization of the LL to get the MLE in the literature. However, the local optimization of LL is problematic and not robust. Therefore, a large part of the work in this thesis has been invested in identifying common problems and developing an improved approach to optimize the hyperparamteres. In this chapter, this approach is discussed.

First, the problems of maximizing the LL and how to avoid them are addressed. Local and global optimizers with optimized parameters are tested on the LL surface. Many different optimizers exist, each coming with advantages and disadvantages. Therefore, the existing optimizers have been systematically investigated to determine which performs best in searching for the optimal hyperparameters. Furthermore, different objective functions are also considered. The performances of the objective functions are evaluated on a set of different atomistic test systems.

A new and improved optimization method specially designed for optimizing the hyperparameters of a GP is developed and discussed. Modifications are also made to improve some of the common objective functions. A reduction in hyperparameter space by a Bayesian approach is discussed in the form of the new TP (see Section 2.4). At last, the FBMGP (see Section 2.5) is compared to the fully Bayesian solution and the MLE solutions.

3.2 Methods

The test systems used for the investigation of the hyperparameter optimizations are:

1. An analytical one-dimensional test function
2. The Müller-Brown potential energy surface[75] (MB)
3. A gold atom on an aluminium(100) surface (AuAl)
4. Carbon monoxide on a nickel(100) surface (CONi)
5. A cluster of five copper atoms (Cu5)
6. A cluster of thirteen copper atoms (Cu13)
7. Two oxygen atoms adsorbed on a platinum(100) surface (O2Pt)
8. Four water molecules above a platinum(111) surface (WaterPt).

The test systems are described in detail in Section A.2.1.

The one-dimensional test system is an analytical function with the expression:

$$g(x) = 3\sin\left(\frac{x^2}{20^2}\right) - 9\sin\left(\frac{0.6x}{20}\right) + 17 \tag{3.1}$$

A database containing 800 data points is constructed from the one-dimensional test function using x values ranging from -40 to 1000. Random noise from a normal distribution with a standard deviation of $1.0{\cdot}10^{-4}$ eV is added to the test function. The one-dimensional test function is introduced as an illustrative function that can be difficult for the GP to learn due to its change in frequency. Therefore, it illustrates the challenges of maximizing the LL. Furthermore, it is also applied to determine the parameters of the optimization methods and illustrate the effects of the applied methods.

The test systems are used with different training set sizes: 3, 6, 12, 25, 50, 100, and 200. 8 different random seeds are also used for each training set size. Thus, success curves with deviations can be constructed. The test set sizes consist of 400 data points that are not included in the used training set.

The mean of the training energies is used as a prior mean for the GP. Furthermore, the SEC with and without derivatives of the energies are used to evaluate the optimization of the hyperparameters. The Cartesian coordinates of the moving atoms are used as features for the atomistic test systems.

Boundary Conditions (BC) are necessary for defining the search space of the hyperparameters. In this work, a set of educated guesses of the hyperparameters is achieved from experiences. The best Educated Guessed Boundary Conditions (EGBC) for the hyperparameters are summarised in Table. 3.1. The EGBC restricts the search space as much as possible. The BC of the length-scale hyperparameter depends on the median of the nearest neighbor distance and the median distance between the training features. A length-scale hyperparameter shorter than the nearest neighbor distance will give a process that overfits the training data. Contrary, a length-scale hyperparameter larger than the largest distance gives a process that underfits the training points. The prefactor hyperparameter controls the magnitude of the uncertainty prediction. Therefore, the prediction uncertainty can not generally be much larger or smaller than the deviation in the training target compared to the prior mean. It is assumed that all the targets are not pure noise and a reasonable interpolation is possible within the EGBC. Therefore, the upper limit of the relative-noise hyperparameter is the largest possible eigenvalue of the factorized covariance matrix when derivatives of the targets are not used. A smaller relative-noise hyperparameter than the machine precision does not change the process. The limits of the length-scale and prefactor hyperparameters in the EGBC can be scaled with a factor s.

Hyperparameter	Min. bound	Max. bound
Length-scale (l)	$\frac{\text{median}(\vec{\text{NN}})}{5s}$	$4s \cdot \text{median}(\mathbf{D})$
Prefactor (α)	$\frac{1}{10s}\sqrt{\frac{1}{N}\|\vec{y}-\vec{\mu}\|^2}$	$10s\sqrt{\frac{1}{N}\|\vec{y}-\vec{\mu}\|^2}$
Relative-noise (σ_r)	$10\sqrt{2\varepsilon_M}$	N

Table 3.1: Table of the boundary conditions obtained by the educated guesses of the hyperparameters when using the squared exponential kernel family. s is the scaling factor chosen, $\vec{\text{NN}}$ is the nearest neighbor distance for each training data in the feature space, $\mathbf{D}$ is the distance matrix in the feature space, and ϵ_M is the machine precision.

100 initial hyperparameter sets are sampled from a uniform distribution given by the defined EGBC for each random seed.

3.2.1 Evaluation measures

The greatest LL values observed for each test system with the specific number of training points and random seed are defined as the global maxima, LL_G.

The success rate, $\mathcal{S}$, is calculated compared to the global maximum:

$$\mathcal{S} = \frac{\sum_{i=1}^{N_s} H(|LL_i - LL_G|)}{N_s} \tag{3.2}$$

where N_s is the number of initial hyperparameter sets and $H(|LL_i - LL_G|)$ is a step function. The step function is defined as:

$$H(|LL_i - LL_G|) = \begin{cases} 1 & \text{if } |LL_i - LL_G| \leq 10^{-3} + 10^{-3}|LL_G| \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

The prediction mean error is measured as the Root-Mean-Square Error (RMSE) given as:

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(y_{*i} - \bar{y}_{*i})^2} \tag{3.4}$$

where M is the number of test points.

In this study, a measure for the prediction uncertainty error is derived. The measure is called the Uncertainty Deviation (UD). The prediction mean error scaled with the prediction uncertainty, z_i, is defined as:

$$z_i = \frac{y_{*i} - \overline{y}_{*i}}{\sigma_{*i}} \tag{3.5}$$

z_i will be called the scaled prediction error. The variance of the scaled prediction errors, σ_z^2, can be expressed as:

$$\begin{aligned} \sigma_z^2 =& \frac{1}{M}\sum_{i=1}^{M}(z_i - \overline{z})^2 = \overline{z^2} - \overline{z}^2 \\ =& \left(\frac{1}{M}\sum_{i=1}^{M}\frac{(y_{*i} - \overline{y}_{*i})^2}{\sigma_{*i}^2}\right) - \left(\frac{1}{M}\sum_{i=1}^{M}\frac{y_{*i} - \overline{y}_{*i}}{\sigma_{*i}}\right)^2 \end{aligned} \tag{3.6}$$

The best variance of the scaled prediction errors must be 1.0. This is caused by the posterior predictive distribution being a standardized Gaussian distribution when the prediction means are scaled with their prediction uncertainties and a Gaussian distribution is assumed from the prediction of a GP. The error in the variance of the scaled prediction error should be symmetric around $\sigma_z^2 = 1.0$. Thereby, the UD is expressed as:

$$UD = \ln\left(\sigma_z^2\right)^2 \tag{3.7}$$

The geometric mean is used for summarising the prediction mean and uncertainty error.

The Negative Log Predictive Probability (NLPP) can also be used as a measure for the prediction error[76, 72]. The NLPP is expressed as:

$$\begin{aligned} NLPP =& -\sum_{i=1}^{M}\ln\left(p(y_{*i} \mid \bar{y}_{*i}, \sigma_{*i}^2)\right) \\ =& \sum_{i=1}^{M}\left(\frac{(y_{*i} - \bar{y}_{*i})^2}{2\sigma_{*i}^2} + \ln(\sigma_{*i}) + \frac{1}{2}\ln(2\pi)\right) \end{aligned} \tag{3.8}$$

3.2.2 Local optimizers

The Python package *SciPy*[77] includes both local and global optimizers. The investigated local optimizers from *SciPy* are:

1. Nelder-Mead [78, 79]
2. Powell [80]
3. Conjugate gradient (CG) [81]
4. Broyden–Fletcher–Goldfarb–Shanno (BFGS) [82, 83, 84, 85]
5. Limited-memory BFGS with boundaries (L-BFGS-B) [86]
6. Truncated Newton (TNC) [87]

The Nelder-Mead and Powell optimizers are non-gradient-based local optimizers, and the rest of the local optimizers use gradients. The local optimizers also have parameters to be tuned, which are considered in Section 3.3. The general parameters are the maximum number of iterations used and the tolerance criterion. The tested maximum numbers of iterations are 500 and 5000. The tolerance criteria tested are 10^{-3}, 10^{-8}, and 10^{-12}. Especially, the TNC method has many parameters. All the tested parameters can be seen in detail in Section A.3. The local optimizers are tested on the one-dimensional test function and by maximizing the LL.

3.2.3 Global optimizers

The best optimizer for finding the MLE is investigated. The investigated optimizers are:

1. Local optimization
2. Local optimization with prior distributions
3. Local optimization with an educated guess
4. Grid search
5. Iterative line search
6. Basin-hopping
7. Random sampling with local optimizations
8. Simulated annealing
9. Simulated annealing with analytical prefactor hyperparameter
10. Factorized line search

The best local optimizer is compared to the global optimizers. Furthermore, a local optimizer that initially maximizes the LP and then maximizes the LL from the result is also tested. Another local optimizer is tested, which maximizes the LL of the initial hyperparameter set and an educated guess of the hyperparameter from the geometric mean of the EGBC.

Some of the global optimizers use a variable transformation of the hyperparameters. The variable transformation is introduced to enlarge the region of interest and simultaneously permit all values of the hyperparameters. The variable transformation is an inverse-scaled logit transformation. The newly transformed hyperparameter, t_θ, is defined in the open interval of $(0.0, 1.0)$. Therefore, the transformed hyperparameter can be sampled from a uniform distribution from 0.0 to 1.0. The inverse variable transformation is expressed as:

$$\ln(\theta) = \mu_\theta + s_\theta \ln\left(\frac{t_\theta}{1 - t_\theta}\right) \tag{3.9}$$

where μ_θ is the mean of the logistic distributions and s_θ is the scale parameter of the logistic distributions. The mean value of the logistic distribution is expressed as:

$$\mu_\theta = \frac{1}{2}\left(\ln(b_{\theta,\min}) + \ln(b_{\theta,\max})\right) \tag{3.10}$$

where $b_{\theta,\min}$ and $b_{\theta,\max}$ are the minimum and maximum EGBC for hyperparameter θ, respectively. The scaling of the logistic distribution is set to 0.14 times the difference of the logarithm of the EGBC of the hyperparameter:

$$s_\theta = 0.14\left(\ln(b_{\theta,\max}) - \ln(b_{\theta,\min})\right) \tag{3.11}$$

The EGBC corresponds to the 95% percentile of the logistic distribution when the 0.14 value is used.

The grid search is the brute force method for finding the hyperparameters. The grid is constructed in the transformed hyperparameter space. The LL is evaluated in all grid points. The grid search is a robust method if the grid is strictly dense. However, the number of evaluations needed for the grid search scales with n^{D_θ}, where n is the number of points in each dimension of the hyperparameters and D_θ is the number of hyperparameters. Often, the grid search method becomes too expensive to use due to the curse of dimensionality. Furthermore, a local optimization is performed for the hyperparameter set that gives the largest LL in the grid.

The iterative line search method is similar to the grid search method. A one-dimensional grid is constructed in each dimension of the transformed hyperparameters. The LL is evaluated in all the points in the one-dimensional grid for one of the hyperparamters. Meanwhile, the rest of the hyperparameters are fixed. Then, the grid point that gives the largest value of LL value is selected, and the procedure is continued for the rest of the transformed hyperparameters. One loop through all the hyperparameters will cost $n \cdot D_\theta$ iterations. Multiple loops are performed. In the end, a local optimization of the best candidate is executed.

Scipy's basin-hopping implementation is used with 15 jumps[77, 88].

19 sets of transformed hyperparameters are sampled from the uniform distributions in the random sampling method. All the sampled hyperparameter sets and one given hyperparameter set are locally optimized.

The simulated annealing method from *Scipy* (called `dual_annealing`) is used[89, 90, 91, 92]. The transformed hyperparameters are searched within a required box in the simulated annealing method. The simulated annealing method does not have any convergence criteria, which makes it an expensive method. The analytical solution of the prefactor hyperparameter (see Eq. 2.33) are also used to search a reduced hyperparameter space.

A new developed global search method specially designed for maximizing the LL is introduced as the factorized line search method. For simplicity, the factorized line search method is denoted as the factorization method. The factorization method uses the analytical solution of the prefactor hyperparameter and performs the eigendecomposition of the covariance matrix with noise as in Eq. 2.33. A grid of 50 points in the transformed relative-noise hyperparameter is constructed. Furthermore, a grid of 80 points is also constructed in the transformed length-scale hyperparameter space. Then, the LL is evaluated in all grid points of the length-scale hyperparameter. All relative-noise grid points can be evaluated from a single eigendecomposition for each length-scale hyperparameter. Hence, an inversion of the covariance matrix is only needed when a new length-scale hyperparameter is used. The largest value of LL in the relative-noise space is located, and a golden-section search[93] is performed in the surrounding interval. All maxima are located with the finite difference method in the grid of the length-scale hyperparameter. A golden-section search is performed in all intervals that surround a maximum. The LP can be used instead of the LL in the same way.

The parameters of the global optimization methods are tested to ensure that the best parameters are used. The investigated parameters are listed in Section A.4.

3.2.4 Objective function

Other objective functions can also be used[76, 94, 95]. Usually, the hyperparameters of ML models are optimized by minimizing the prediction error, which is calculated using Cross-validation. A special case of CV is the Leave-One-Out Cross-Validation (LOOCV) method, in which a ML model is trained on all except one training point, and the prediction error is calculated for the excluded point. This process is repeated for all training targets to give an unbiased error. A GP has the advantage that the LOOCV can be analytically calculated without retraining the GP N times[76]. The analytical expression of the LOOCV, LOO, is:

$$\begin{aligned} LOO \equiv & \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_{-i})^2 \\ = & \frac{1}{N}\sum_{i=1}^{N}\left(\frac{[\mathbf{C}_0^{-1}(\vec{y}(\mathbf{X}) - \vec{\mu}(\mathbf{X}))]_i}{[\mathbf{C}_0^{-1}]_{ii}}\right)^2 \end{aligned} \tag{3.12}$$

where $\bar{y}_{-i}$ is the prediction of the excluded training target that the GP is not trained on.

The prediction uncertainty of the excluded point, $\sigma^2_{y_{-i}}$, can also be calculated analytically for the GP as:

$$\sigma^2_{y_{-i}} = \frac{\alpha^2}{[\mathbf{C}_0^{-1}]_{ii}} \tag{3.13}$$

Another objective function can be derived that minimizes the prediction mean error and the magnitude of the uncertainty prediction. The objective function is called Geisser's Predictive mean square Error (GPE) and is expressed as:

$$\begin{aligned} GPE \equiv & \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_{-i})^2 + \frac{1}{N}\sum_{i=1}^{N}\sigma^2_{y_{-i}} \\ = & LOO + \frac{\alpha^2}{N}\sum_{i=1}^{N}\frac{1}{[\mathbf{C}_0^{-1}]_{ii}} \end{aligned} \tag{3.14}$$

The predictive probability can also be written as a LOOCV version with the Geisser's surrogate Predictive Probability (GPP). It is expressed as:

$$\begin{aligned} GPP \equiv & \frac{1}{N}\sum_{i=1}^{N}\frac{(y_i - \bar{y}_{-i})^2}{\sigma^2_{y_{-i}}} + \frac{1}{N}\sum_{i=1}^{N}\ln\left(\sigma^2_{y_{-i}}\right) + \ln(2\pi) \\ = & \frac{1}{N\alpha^2}\sum_{i=1}^{N}\frac{[\mathbf{C}_0^{-1}(\vec{y}(\mathbf{X}) - \vec{\mu}(\mathbf{X}))]_i^2}{[\mathbf{C}_0^{-1}]_{ii}} + \ln\left(\alpha^2\right) - \frac{1}{N}\sum_{i=1}^{N}\ln\left([\mathbf{C}_0^{-1}]_{ii}\right) + \ln(2\pi) \end{aligned} \tag{3.15}$$

In this work, a modification to the *LOO* is made that gives it a better uncertainty prediction without changing the prediction mean. The original *LOO* is independent of the prefactor hyperparameter, and the prediction uncertainty is therefore not optimized. The modification is an analytical determination of the prefactor hyperparameter expressed as:

$$\alpha_{mod}^2 = \left(\frac{1}{N} \sum_{i=1}^{N} \frac{[\mathbf{C}_0^{-1}\,(\vec{y}(\mathbf{X}) - \vec{\mu}(\mathbf{X}))]_i^2}{[\mathbf{C}_0^{-1}]_{ii}} \right) - \left(\frac{1}{N} \sum_{i=1}^{N} \frac{[\mathbf{C}_0^{-1}\,(\vec{y}(\mathbf{X}) - \vec{\mu}(\mathbf{X}))]_i}{\sqrt{[\mathbf{C}_0^{-1}]_{ii}}} \right)^2 \tag{3.16}$$

A derivation of Eq. 3.16 is given in Section A.6. The modification has no extra computational cost.

The analytical solution of the prefactor hyperparameter from the LL (Eq. 2.32) can also be modified after the maximization. The prefactor hyperparameter can be changed to an unbiased estimate of the variance as:

$$\alpha_{\text{mod}}^2 = \frac{1}{N - D_\theta} (\vec{y} - \vec{\mu})^\top \mathbf{C}_0^{-1} (\vec{y} - \vec{\mu}) \tag{3.17}$$

where D_θ is the number of optimized hyperparameters.

The LP is also tested with prior distributions of the length-scale and relative-noise hyperparameters. The prior distribution of the length-scale hyperparameter is a normal distribution in the logarithmic space with a mean of 2.0 and a standard deviation of 3.0. The prior distribution of the relative-noise hyperparameter is also a normal distribution in the logarithmic space with a mean of −9.0 and a standard deviation at 3.0.

3.3 Results & Discussion

3.3.1 Optimization challenges

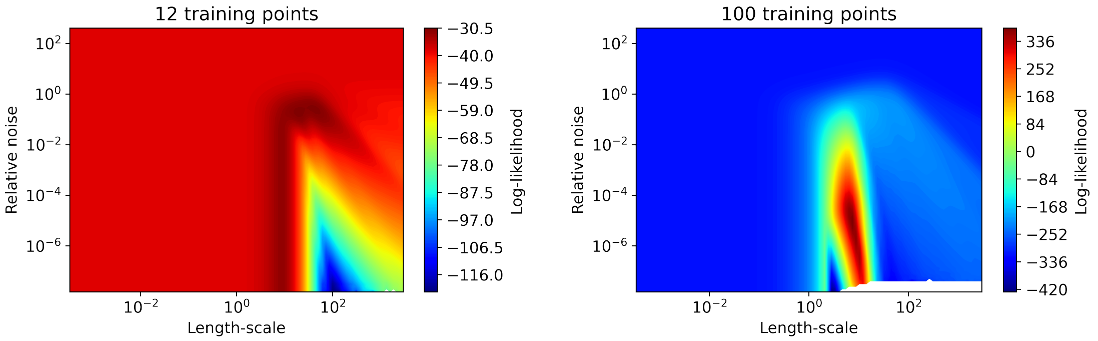
The optimization of the three most common hyperparameters by local maximization of the LL can easily fail. Multiple problems can cause the local optimization to be unsuccessful.

Inversion problems are common since the covariance matrix becomes singular when is the case at high length-scale hyperparameters and small relative-noise hyperparameters. In this region, the covariance matrix becomes an all-ones matrix. Therefore, a noise correction, δ_n, has been introduced as:

$$\delta_n = \frac{\mathrm{Tr}(\mathbf{K}_0)^2}{(\epsilon_M)^{-1} - N_K^2} \tag{3.18}$$

where ϵ_M is the machine precision and N_K is the number of diagonal elements in the covariance matrix. A derivation of Eq. 3.18 is given in Section A.7. The noise correction is crucial since the local optimization is terminated immediately if an error occurs. The noise correction is a constant when the same number of training points is used and the derivatives of the energies are not used due to the definition of the relative-noise hyperparameter instead of the noise hyperparameter.

Another problem is the large region with no gradients wrt. to the hyperparameters at low length-scales and low relative-noise hyperparameters (see Fig. 3.1). A local optimization initialized in this region will immediately converge without finding the global maximum. The region corresponds to GPes that overfits the data. The length-scale hyperparameter depends on the feature distances and the variation of the targets as a function of the features. Therefore, the magnitude of the best length-scale hyperparameter is not known in advance. The region is flat due to the numerical precision of the exponential term in the



(a) 12 training points with random seed 7

(b) 100 training points with random seed 7

Figure 3.1: The log-likelihood surface of a Gaussian Process with analytical maximized prefactor hyperparameter. No noise correction is used. 12 training points are used in figure (a) and 100 training points are used in figure (b). The test system is the analytical one-dimensional system.

covariance function ($\exp(-750) \approx 0$). The SEC matrix becomes the identity matrix when the length-scale hyperparameter is 0.025 times the smallest feature distance. It can also be seen from the region with low length-scale hyperparameters that the posterior distribution of the hyperparameter is an ill-posed problem when uniform prior distributions are chosen. This problem is seen since the integration of the LL over the length-scale hyperparameter is not converging towards a finite value.

The region at high relative-noise hyperparameters is also flat. Therefore, local optimization is not a feasible option for finding the global maximum in this region. The GPes from this region corresponds to completely regularized predictions, where the targets are treated as noise.

The LL surface can also have multiple maxima (see Fig. 3.1 (a)). Therefore, a local optimizer can easily end up at the wrong local maximum. Multiple reasonable GPes can be constructed from the training data. E.g. for the one-dimensional test function at 12 training points, a regularized model that identifies the underlying sine function has the largest LL value, and another model that fits through the points is not as likely. The prediction mean and uncertainty of the GP can be quite different for local maxima of the LL. It is not necessarily the global maximum of the LL that gives the best predictions. Therefore, it is essential to express a prior expectation in the form of a prior distribution of the hyperparameter, e.g. if a low noise of the targets is expected. More training data often gives a distinct maximum in LL. The MLE is a good approximation when a large training set size is applied.

The factorization of the covariance matrix to be independent of the prefactor hyperparameter stabilizes the inversion of the covariance matrix. This is a consequence of the condition number of the covariance matrix being dependent on the machine precision. Naturally, the inverse covariance matrix is the same for all prefactor hyperparameter values.

It is beneficial to optimize the hyperparameters in the logarithmic space (see Fig. 3.2). The hyperparameters must be scaling invariant. Therefore, the values of the hyperparameters can have any magnitude within machine precision. Furthermore, the success rates of finding the global maxima of the LL in the logarithmic space are higher than in the linear space. Therefore, the hyperparameters are defined in the logarithmic scale in the code.

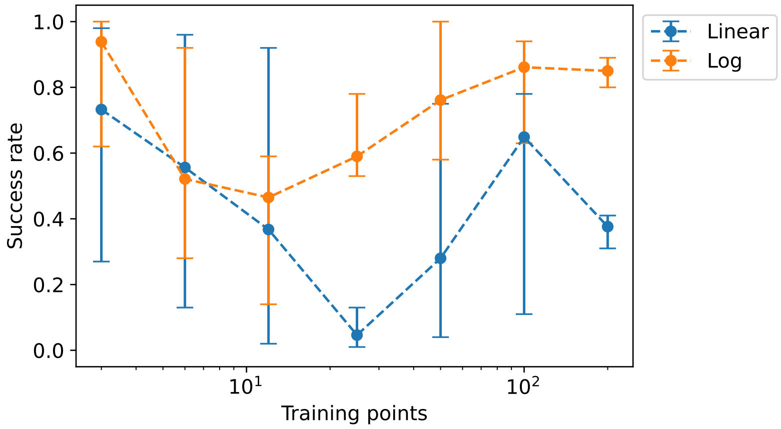


Figure 3.2: A comparison in success rate between hyperparameters in the linear (orange curve) and logarithmic space (blue curve). 100 initial sets of hyperparameters for every eight random seeds at each training set size are locally optimized with L-BFGS-B from Scipy. The test system is the analytical one-dimensional system.

The large flat regions of the LL surface make it essential to define a limited search space or BC for the hyperparameters. The influence of the area of the BC can be seen in Fig. 3.3. The success rates of finding the global maxima of the LL with local optimizations are noticeably higher for more restricted BC. The success rates decrease as a function of the area of BC.

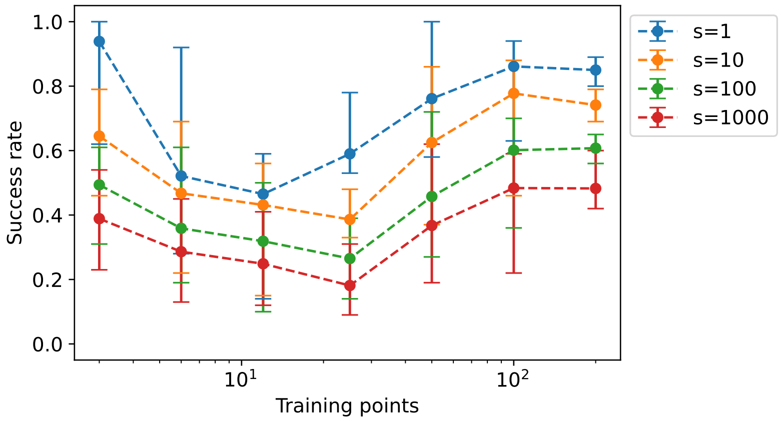
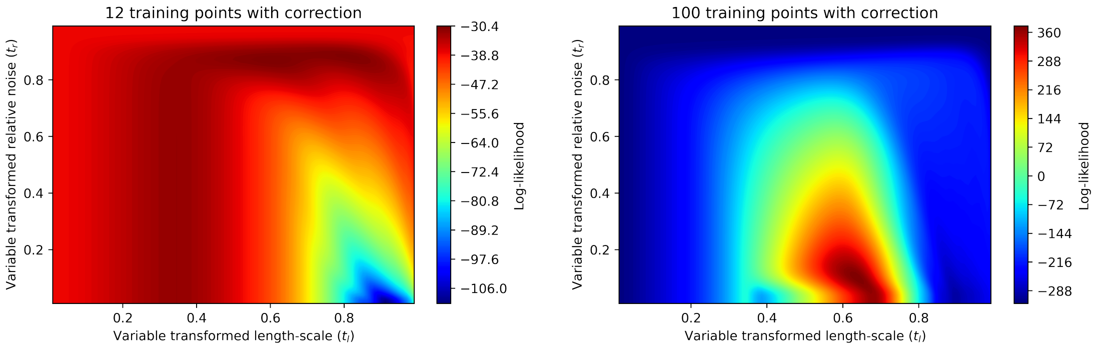


Figure 3.3: A comparison in success rate between hyperparameters sampled in 1 time (blue curve), 10 times (orange curve), 100 times (green curve), and 1000 times (red curve) the educated guess boundary condition interval of the length-scale and prefactor hyperparameter. 100 initial sets of hyperparameters for every eight random seeds at each training set size are locally optimized with L-BFGS-B from Scipy. The test system is the analytical one-dimensional system.

The variable transformation of the hyperparameters enlarges the most important region of the hyperparameter space without restricting the hyperparameter search (see Fig. 3.4). Therefore, it is more likely to sample a good initial set of hyperparameters in the variable transformed space.



(a) 12 training points with random seed 7

(b) 100 training points with random seed 7

Figure 3.4: The log-likelihood surface of a Gaussian Process with analytical maximized prefactor hyperparameter. Noise correction and the variable transformation of the hyperparameters are used. 12 training points are used in figure (a) and 100 training points are used in figure (b). The test system is the analytical one-dimensional system.

3.3.2 Parameter tuning of optimizers

The local optimization methods have many parameters that change the average success rates and the number of performed iterations (see Fig. 3.5). Unsurprisingly, the non-gradient-based local optimizers use on average a larger number of LL evaluations than the gradient-based local optimizers. On average, the non-gradient-based local optimizers have larger success rates. The Powell and TNC local optimizers have a parameter set each with the largest success rate for finding the global maximum of the LL. However, the L-BFGS-B has almost as high a success rate for a single parameter set (0.71 vs. 0.73 for TNC and Powell). The TNC local optimizer strongly depends on its parameter. The L-BFGS-B uses significantly fewer iterations on average. Therefore, L-BFGS-B is Pareto-optimal in terms of success rate and iterations. Thus, L-BFGS-B is selected as the default local optimizer.

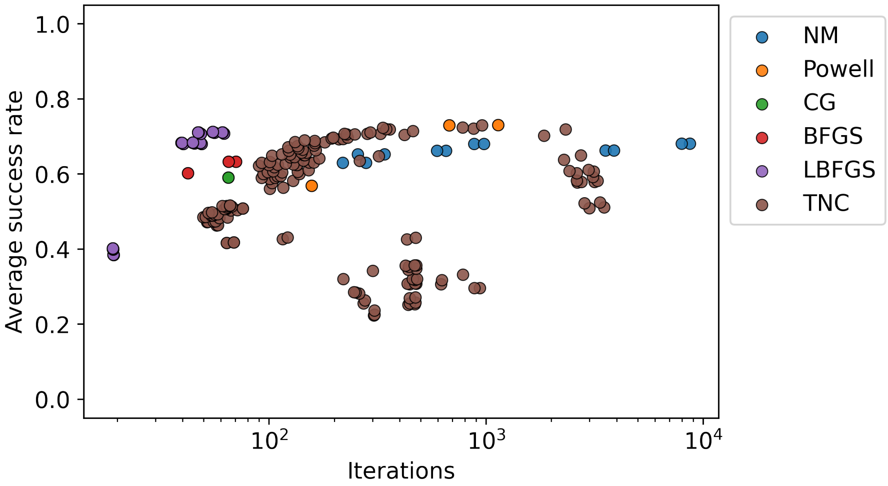


Figure 3.5: The average success rates as a function of the average iterations for finding the global maximum of the log-likelihood. Different local optimizers are used. The average of 7 training set sizes and 8 random seeds on the one-dimensional test system is used.

The parameters of the global optimizers change their performance significantly (see Fig. 3.6). Especially the grid and iterative line search methods are extremely parameter dependent. The factorization method has an average success rate of 1.0 as the only optimizer and is the global optimizer that uses the fewest LL evaluations. Whereas the factorization method thus is Pareto-optimal in terms of iterations and success rate, the random sampling method consistently performs well in terms of success rate, although using more iterations. Hence, a more detailed investigation of the global optimization methods considering different test systems is required.

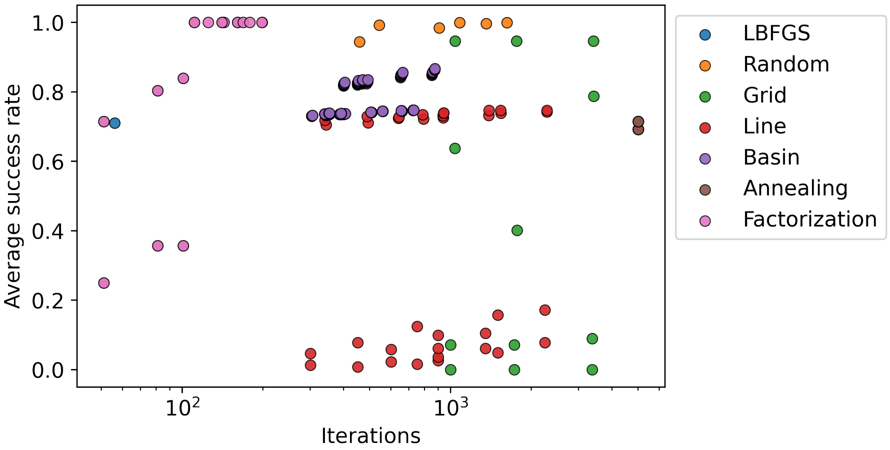


Figure 3.6: The average success rates as a function of the average iterations for finding the global maximum of the log-likelihood. Different global optimizers are used. The average of 7 training set sizes and 8 random seeds on the one-dimensional test system is used.

3.3.3 Global optimization methods

The optimization methods are used on all 9 test systems with 7 training set sizes each and 8 random seeds each. The results from the global optimizations are observed in Fig. 3.7. The factorization method finds the global maximum of the LL for all test systems (success rate of 1.0). It is the only method that locates the global maximum for all test systems every time. The eigendecomposition of the covariance matrix is more computationally expensive than the Cholesky decomposition. However, the computational time of the factorization method is still less than the rest of the global optimizers except for Basin-hopping since it requires less iterations.

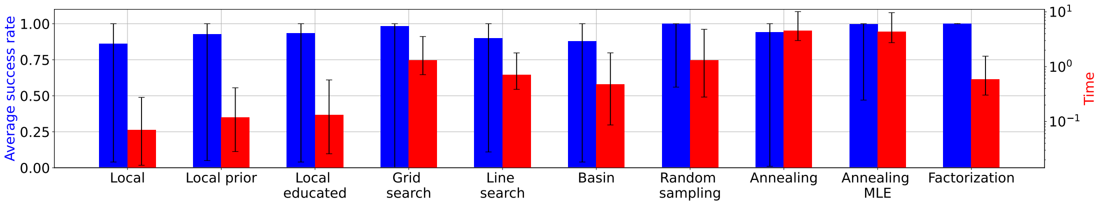


Figure 3.7: The average success rate of finding the global maxima of the log-likelihood for 9 test systems each with 7 training set sizes and 8 random seeds with different optimizers. The average time is also shown. The error bars show the smallest and largest value observed. Here, the derivatives of the targets are not used.

The random sampling and the simulated annealing with analytical prefactor hyperparameter methods have almost a 100 % average success rate (see Section A.5) for locating the global maximum of LL for all test systems every time. Nonetheless, they identify the global maximum for all test systems with different training set sizes within the 100 initial sets of hyperparameters. Therefore, it is a probabilistic problem that both methods are built on.

The local optimization is improved by using prior distributions or educated guesses of the hyperparameters. However, neither prior distributions nor educated guesses of the hyperparameters give a robust method for obtaining the global maximum for all systems. The success rates change significantly for the local optimization methods. Therefore, it can be challenging to determine the number of random samplings to use if multiple local optimizations are performed.

The same trends are observed when the derivatives of the targets are applied (see Section A.5). The factorization method also finds the global maximum of the LL when derivatives of the targets are used.

3.3.4 Objective function optimization

Different objective functions are tested on all the test systems with different training set sizes and random seeds, and the geometric mean of their prediction means and uncertainties are compared.

The geometric means of the prediction means and uncertainties from LOOCV show that the performance of the GP is strongly dependent on the objective function optimized (see Fig. 3.8). The LOOCV is used to understand how well the objective functions optimize

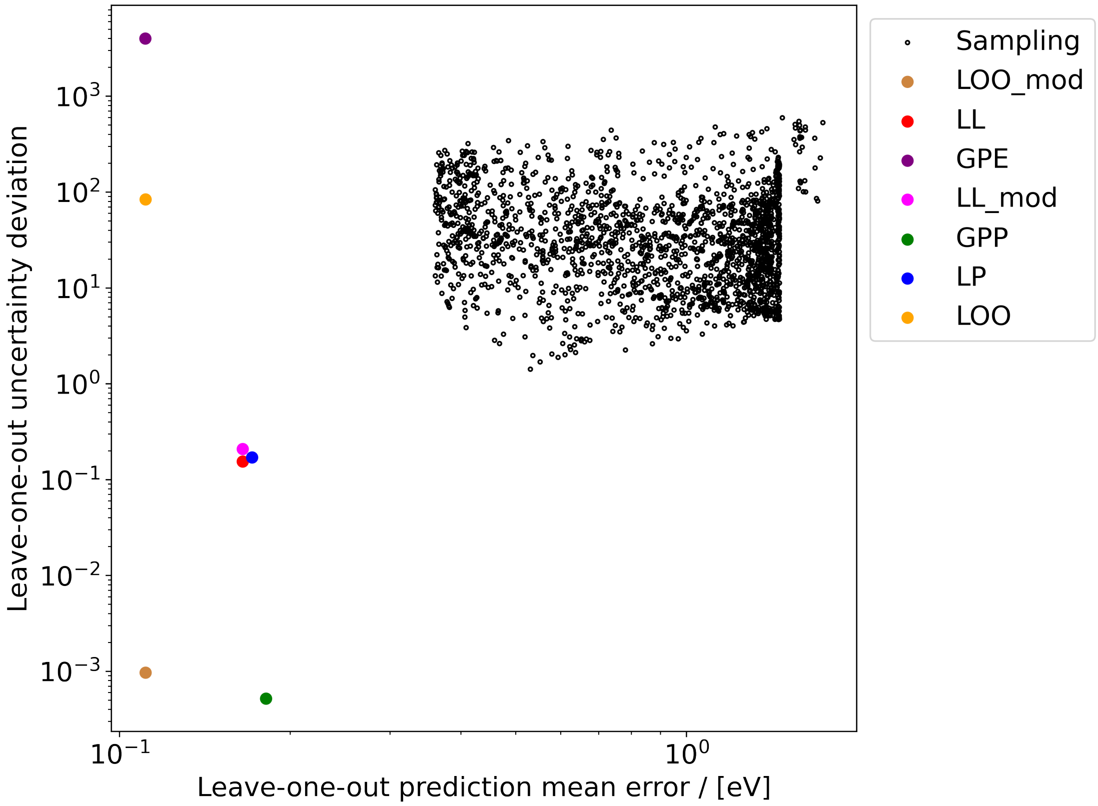


Figure 3.8: Geometric mean of prediction means and uncertainties for 9 test systems with 7 different training set sizes and 8 random seeds. The prediction means and uncertainties are for leave-one-out cross-validations of the training sets. Different objective functions are tested. Samplings of fixed hyperparameter sets are also used.

the interpolation of the training points without including all of them simultaneously. Accordingly, the exploitation of the training data within the given information is evaluated by LOOCV. The sampling of 2000 sets of fixed hyperparameters from a uniform distribution has also been used for predictions. The uniform distribution is in the (natural) logarithmic space of the hyperparameters. Uniform distributions ranging from -4.0 to 4.0, from -18.0 to 4.0 and from -4.0 to 4.0 have been applied for the length-scale, relative-noise and prefactor hyperparameters, respectively. The sampled fixed hyperparameter sets show that the optimization of an objective function is essential for the LOOCV error. The GPE objective function has a low prediction mean error for the LOOCV. However, the quality of prediction uncertainty from the GPE objective function is poor. Unsurprisingly, the poor prediction uncertainty from the GPE objective function is due to the term that gives a penalty for the magnitude of the prediction variances in the objective function (see Eq. 3.14). The LOO objective function likewise has a low LOOCV prediction mean error. The prediction mean error is the same as for GPE since GPE includes the LOO as a term. The LOO objective function performs well on the prediction mean error which shows that the analytical expression for the LOOCV in a GP works well. However, the uncertainty prediction is poor since it is random due to the independence of the prefactor hyperparameter. The modification to the LOO objective function gives a good LOOCV prediction uncertainty error while keeping the same good prediction mean. The GPP objective function has a larger LOOCV prediction mean error but has a better uncertainty prediction than the modified LOO. The maximization of the LL gives a larger LOOCV prediction uncertainty error than for GPP and the modified LOO objective functions. The LOOCV prediction mean errors for the LL solutions are in general larger than for LOO, but smaller than for GPP. The modification of the LL does not improve the LOOCV prediction uncertainty. Thus, the GPP and the modified LOO give a good interpolation of the known data and are Pareto-optimal for the LOOCV prediction errors. However, the methods are also tuned to be good at LOOCV.

Optimizing the hyperparameters is important for the prediction quality of unseen test sets (see Fig. 3.9). The optimization of the GPE results in poor prediction uncertainties and good prediction means for the test sets too. The modification of the LOO gives good prediction means and uncertainties. The prediction qualities of the GPP objective function on the test sets are slightly worse than the modified LOO. However, the predictions from the modified LL objective function perform better than the GPE, GPP, and the modified LOO on the test sets. The modification to LL improves significantly the prediction uncertainty on the test sets. The LL objective function without the modification has a better prediction mean but a slightly worse prediction uncertainty compared to the modified LOO. The LL with the modification is the Pareto-optimal objective function for the prediction of the test sets. The LL without the modification and the LOO with the modification perform almost as well as LL with the modification. The change in the prediction errors from LOOCV to the test sets is small for the LL objective functions compared to the other objective functions. Therefore, LL is more consistent in its prediction error estimation and can be more robust.

The same trends are seen when the derivatives of the targets are applied (see Section A.8). However, in this case the GPP performs better than LOO with the modification in terms of prediction errors. Maximizing the LL objective function gives better prediction errors than the GPP and LOO with the modification.

3.3.5 Student's T Process

The new TP is introduced, and its prediction errors when its hyperparameters are maximized with LL are compared with the results for the GP (see Table 3.2). The TP has the

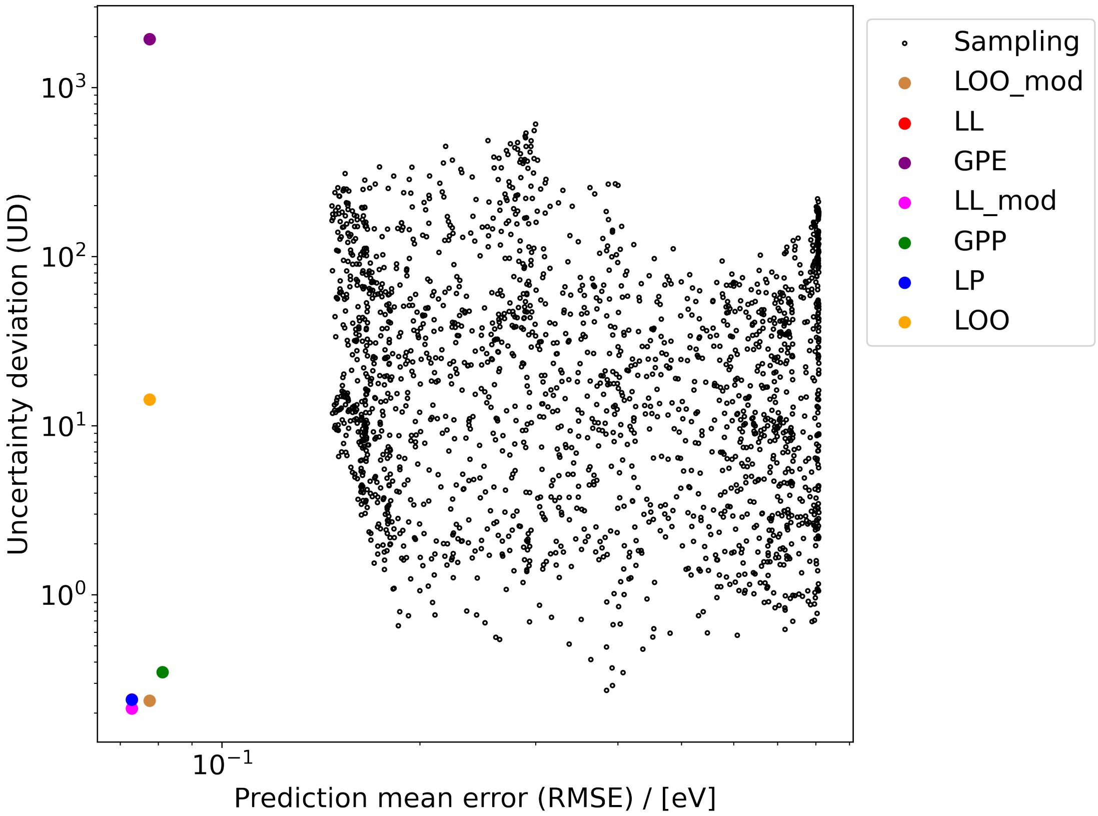


Figure 3.9: Geometric mean of prediction means and uncertainties for 9 test systems with 7 different training set sizes and 8 random seeds. The prediction means and uncertainties are for test sets. Different objective functions are tested. Samplings of fixed hyperparameter sets are also used.

Method	RMSE/ [eV]	UD	NLPP
GP LOO	7.76e-02 (1.11e-06,1.61e+02)	2.38e-01 (1.56e-06,1.93e+02)	4.46e+05
GP LL	7.29e-02 (8.07e-07,8.50e+00)	2.41e-01 (3.38e-07,1.10e+02)	2.06e+04
GP LP	7.29e-02 (8.10e-07,8.77e+00)	2.40e-01 (1.48e-06,8.46e+01)	7.18e+03
TP LL	7.29e-02 (8.07e-07,8.50e+00)	2.15e-01 (8.07e-07,8.85e+01)	6.43e+03
TP LP	7.29e-02 (8.10e-07,8.77e+00)	2.17e-01 (2.03e-05,6.56e+01)	1.91e+03
FBMGP weak.	7.65e-02 (8.53e-07,7.88e+00)	2.12e-01 (1.93e-06,7.07e+01)	1.67e+03
FBMGP info.	7.50e-02 (8.31e-07,7.69e+00)	1.60e-01 (1.16e-09,4.90e+01)	-1.57e+02

Table 3.2: Table of the prediction mean (RMSE) and uncertainty (UD) errors for the Gaussian process and the Student's t process. The Student's t process is optimized with log-likelihood and log-posterior. The Gaussian process is optimized with log-likelihood, log-posterior, and the modified leave-one-out objective function. The fully Bayesian mimicking Gaussian process (FBMGP) is calculated with weak and informative prior distributions. The errors are geometric means over 9 test systems each with 7 training set sizes and 8 random seeds. However, the average is used for the Negative Log Predictive Probability. The brackets identify the smallest and largest value observed.

same prediction mean errors of the test systems as the GP. The TP and GP have the same prediction mean expressions, and therefore the prediction mean errors are the same when the hyperparameters are identical. The geometric mean of the prediction uncertainty error of the TP is smaller than for the GP. Furthermore, the largest predicted uncertainty error is smaller for the TP. The NLPP measure is also smaller for the TP. The prediction uncertainty is especially improved for the TP compared to the GP at few training points. The prediction uncertainty of the TP goes towards the prediction uncertainty of the GP as a function of training points like the unbiased estimation of the variance. However, one or two training points will lead to huge uncertainty predictions for the TP.

The computational time of optimizing the TP is less or equal to the GP. Hence, the TP is an improvement to GP with better uncertainty predictions that comes with no additional computational cost.

3.3.6 Fully Bayesian mimicking Gaussian Process

In this section, weakly informative and informative prior distributions of the hyperparameters are used for the FBMGPes. The weak informative prior distribution of the length-scale hyperparameter is a normal distribution in the logarithmic scale with a mean of 0.0 and a standard deviation of 35.0. The weak informative prior distribution of the relative-noise hyperparameter is also a normal distribution in the logarithmic scale with a mean of -9.0 and a standard deviation of 18.0. The weak informative prior distribution of the prefactor hyperparameter is an inverse-gamma distribution with parameters $a = b = 10^{-20}$, which was also used for deriving the TP (see Section 2.4). The informative prior distributions are the same as the prior distributions used for the LP of the GP and TP (see Section 3.2.4) together with the aforementioned weakly informative prior distribution of the prefactor.

The FBMGP is derived from a grid of hyperparameters with LP values. The grid is constructed in the space of the variable transformed hyperparameters. The space of the variable transformed hyperparameters provides a better description of the important parts of the hyperparameter space. It is important that LP is used and not the LL. This is because the integration of the LL surface does not integrate to a finite number due to the large flat regions. The influence of the flat regions can be decreased by using informative prior distributions.

The geometric mean of the prediction uncertainty errors for the FBMGPes with both prior distributions is even better than for the TP (see Table 3.2). However, the geometric mean of the prediction mean errors are slightly worse for the FBMGPes due to the contributions of the flat regions. The prediction means tend to be more overfitted when few training points are used to have a larger uncertainty instead (see in Fig. 3.10). The larger uncertainties for the good prediction means are due to the contribution of multiple maxima. When a larger number of training points is used, the same prediction mean and uncertainty are predicted as for the MLE as expected. Noticeably, the largest prediction mean errors are decreased when using the FBMGPes. Informative prior distributions of the hyperparameters significantly improve the FBMGP since the influence of the flat regions is reduced and the LP distribution is well-behaved. The NLPP measure also shows that the FBMGP with informative prior distributions is a significant improvement to the MLE or MAP.

The fully Bayesian solution is approximated as a mixture model of all calculated GPes with different sets of hyperparameters from the grid weighted by their LP values. The FBMGP solution is closely related to the fully Bayesian solution.

The computational cost of the FBMGP solutions is larger than the cost of the MLE and MAP solutions (2.6 times on average). However, the FBMGP can be used for any new test point without being retrained as generally required for a fully Bayesian solution.

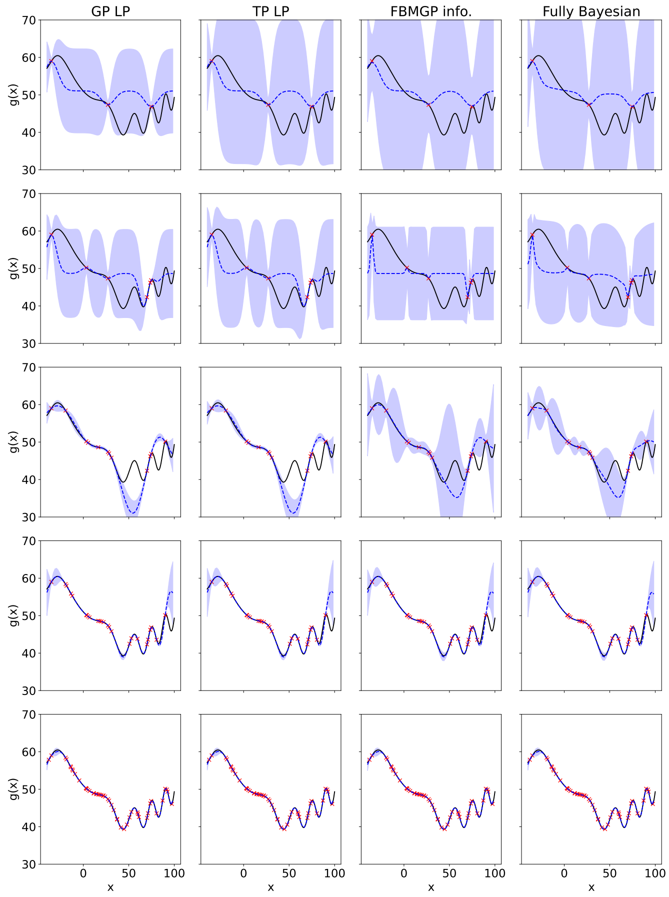


Figure 3.10: Predictions of a one-dimensional test function from Gaussian processes and a Student's t process. The different rows show the use of 3, 6, 12, 25, and 50 training points. The blue areas show two times the uncertainty predictions. The Gaussian processes are optimized by either maximizing the posterior distribution or mimicking the fully Bayesian solution. The Student's t process is optimized by maximizing the posterior distribution.

3.4 Conclusion

A robust optimization of the hyperparameters is the most important prerequisite for obtaining a reliable GP model. However, this optimization is not straightforward and will often lead to errors in the final results. In this section, several improvements to the optimization process have been developed. Furthermore, educated guesses of BC for the hyperparameters have been defined for restricting the search to a reasonable region of GPes. The variable transformation of the hyperparameters enlarges the EGBC without restricting the hyperparameters from values outside the BC. Thus, the problematic large flat regions of the LL are reduced in the hyperparameter space.

A new method (the factorized line search or factorization method) for finding the three most common hyperparameters have been implemented. The method is robust, and it finds the global maximum of LL for all studied test systems with different training sizes. The computational cost of the factorization method is smaller than other global optimization searches that do not guarantee finding the global maximum. A finer grid for the factorized line search can easily be constructed if the basin of attraction is unlikely small.

The LL objective function is confirmed to be the optimal objective function for MLE in terms of both the prediction mean and uncertainty. A modification to the solution of the prefactor hyperparameter obtained from LL maximization gives a significantly better uncertainty prediction due to the unbiased estimate of the prediction variance. The LP can be an optimal objective function for MLE if the user has good prior knowledge of the system. The modification to LOO objective function makes it a competitive method to LL.

A new type of process is derived as the TP. The TP is similar to the GP but does not include the prefactor hyperparameter. The prediction uncertainty is improved by the TP compared to the GP from a Bayesian approach of removing the prefactor hyperparameter. The prediction means of the TP is identical to the GP. The TP has no extra computational cost.

At last, an approach for estimating the fully Bayesian solution of the posterior predictive distribution is derived. Usually, the fully Bayesian solution is only obtainable by mixture models or Monte Carlo simulations[72]. In this work, a single GP is obtained that mimics the fully Bayesian solution. The GP can be used to predict new test points without being retrained. Informative prior distributions of the hyperparameters significantly improve the fully Bayesian solution.

4 Machine Learning Accelerated Global Optimization method

The developed method in this chapter is in collaboration with Kirsten Winther at SUNCAT, SLAC National Accelerator Laboratory.

4.1 Introduction

When a chemical reaction is studied, the Global Minimum Energy Structures (GMESs) of reactants and products are always needed. This is a consequence of the GMES being the most stable and likely structure to appear in a reaction. Therefore, the probabilities of other structures must be compared to the GMES. Often, multiple local minima are present in a reaction where the energy difference is small. A heterogeneous catalytic reaction has multiple local minima in the forms of adsorption sites, and it is not trivial to know intuitively what site corresponds to the GMES. Particularly, when complicated surfaces are involved, it is hard to recognize the symmetry[96].

Many approaches exist to finding the GMES. Some of the approaches are genetic algorithms[97, 98], basin-hopping[88], minima hopping[99, 96], random sampling with local relaxations, and a range of educated guesses with local relaxations. The educated guesses can be significantly faster than the other approaches if the surface and adsorbate are simple and the user has good chemical intuition. However, the surface and adsorbate easily become too complicated, and educated guesses of the GMES will lead to biased results. The other global search methods are often unbiased if the sampling is stochastic. However, they are computationally expensive since they require many iterations. Furthermore, the global optimization methods are never guaranteed to find the GMES. Thus, global optimization methods are often kept running for a given number of iterations without convergence criteria. The brute force grid search will be robust if the grid is dense. Unfortunately, the computational evaluation method is too costly, and the grid search suffers from the curse of dimensionality.

Constructing a workflow that considers different surfaces for a heterogeneous catalysis reaction can be complicated because the surfaces can have different sites and different forms.

ML has shown to be able to accelerate minimum energy structure searches considerably [17, 100, 44, 47, 48, 101]. A variety of algorithms have been introduced that performs GMES searches of metal clusters on surrogate surfaces. A surrogate surface is a PES predicted by ML model.

In this study, a global search method for adsorption structures is introduced. The method does not aim to optimize all structures and clusters but is restricted to finding the best adsorption site. To the author's knowledge, a standard method for finding the best adsorption site does not exist. The method uses a global search for moving the adsorbate around the surface on a surrogate surface. The global search aims to find a compromise between exploration and exploitation. The exploration part is important for learning new regions of the feature space and finding new candidates for the GMES[100]. The exploitation is important for obtaining an accurate prediction of the GMES with low uncertainty. Afterward, the combined structure is relaxed on the surrogate surface to obtain the optimal structure.

4.2 Methods

The method developed in this thesis, Machine Learning Accelerated Global Adsorption Optimization method (MLGO), is implemented in Python with the same format as the ASE[102]. The MLGO is built on the assumption that the energy difference is between the adsorbate placed with fixed bond lengths on a fixed surface and the GMES is small. Therefore, it is beyond the scope of the method to consider surfaces that undergo reconstruction during the adsorption step. Furthermore, it is assumed that the adsorbate does not change significantly after adsorbing on the surface.

The MLGO algorithm is initialized by providing a surface and an adsorbate. The surface and adsorbate must have the same cell sizes and periodic boundary conditions. The fixed-atom constraints of the surface or adsorbate specified by the user are also present in the MLGO simulation. MLGO also provides the option to perform a global optimization of two adsorbates simultaneously. Furthermore, an ASE calculator that calculates the true potential energy surface has to be provided. A default ML calculator is available in the code, however, another ML calculator implemented as a subclass of the ASE calculator can be specified by the user.

The global optimization method is dual simulated annealing[90, 91, 92] from *SciPy*[77]. The simulated annealing moves the adsorbate with fixed bond lengths within some boundary conditions. The boundary conditions for the global optimization search can be passed to the MLGO object. The boundary conditions consist of 6 ranges (list of lower and upper bounds for the optimized variables) for one adsorbate and 12 ranges for two adsorbates. The first boundary corresponds to the scaled first unit cell vector of the geometric center of the adsorbate. Similarly, the second and third boundary ranges correspond to the scaled second and third unit cell vectors of the geometric center of the adsorbate, respectively. The last three boundary ranges are rotation angles of the adsorbate. Hence, the adsorbate atom or molecule with fixed bond lengths can be placed at all positions and angles in the defined boundary conditions. The default boundary conditions are the entire cell of the surface structure and all angles. However, it is beneficial to only search the top layer if a surface is studied.

The MLGO algorithm starts by calculating a number of initial structures given by the `initial_points` argument before the surrogate surface is used (see the pseudo-code of the MLGO algorithm at 1). The initial structures are sampled by moving the center of the adsorbate with the global optimization method, where the energies are calculated as the repulsive potential energy. Two initial structures are calculated in this work. After the initial structures are calculated with the ASE calculator, the ML model is trained and optimized. Then, the global minimization of an acquisition function is performed. The acquisition function determines how much the ML model is exploring and exploiting in the global search. An acquisition function object can also be specified for the MLGO algorithm. The suggested and used acquisition function is the lower confidence bound expressed as:

$$a(\vec{x}_i) = E(\vec{x}_i) - \kappa\sigma_*(\vec{x}_i) \tag{4.1}$$

where $a(\vec{x}_i)$ is the acquisition function value of the test point with coordinates $\vec{x}_i$ and κ corresponds to the number of standard deviations in a Gaussian distribution. A good value for κ is 3.0, and it is used in this study if not specified otherwise. The global optimization is performed in parallel if multiple Central Processing Units (CPUs) are used. The number of global searches that are performed in parallel is set by `ml_chains`, which is 10 in this work. Multiple global searches are performed to ensure that the best candidate from the global search is suggested without using more computational time. Each global search

Algorithm 1 MLGO

```
Require: Surface, adsorbate, ASE calculator, ML calculator, second adsorbate (optional).
Ensure: Global minimum energy structure
  Calculate initial structures from simulated annealing
  converged ← False
  while converged ≠ True do
      Train ML calculator
      Simulated annealing on acquisition function in parallel
      if Number of structures in ML calculator ≥ norelax_points then
          while $\max(\sigma_*(\vec{x}_i)) \leq$ max_unc do
              Local relaxation on the surrogate surface in parallel
          end while
      end if
      Chose candidate from acquisition function
      Evaluation of the candidate with ASE calculator
      if Number of structures in ML calculator ≥ min_steps then
          if $|F_i| \leq$ fmax then
              if $\max(\sigma_*(\vec{x}_i)) \leq$ unc_convergence then
                  if $|E_* - E| \leq 2 *$ unc_convergence then
                      if $|E_* - E_{min}| \leq$ unc_convergence then
                          converged ← True
                      end if
                  end if
              end if
          end if
      end if
  end while
```

is doing `ml_steps`, which is set to 2000. Subsequently, the structures obtained from the global searches are locally relaxed on the predicted PES. The local optimization method can be specified by the user, in this work the MDMin optimizer from ASE is used. All atoms that are not affected by the fixed-atom constraints are relaxed. However, the number of training points used to train the ML calculator must be greater than or equal to `norelax_points` before the local relaxations can be executed. `norelax_points` is set to 10 as default since the global environment must be well determined before a local relaxation in case the uncertainty prediction is underestimated. The local relaxations can also be deselected if the argument `relax` is set to `False`. Furthermore, the uncertainty of the final structure obtained from the global search part must be lower than or equal to `max_unc` to start the relaxation. The `max_unc` is set to 0.05 eV in this work. The local relaxations are also parallelized for each final structure if multiple Central Processing Units (CPUs) are used. The uncertainty is checked for each step in the local relaxation, and if it exceeds `max_unc` then the local relaxation is stopped. The best structure of the candidates from the local relaxations or global searches is chosen from the acquisition function. In this work, the candidate with the largest uncertainty is chosen if it is greater than `max_unc` or else it will be the candidate with the lowest acquisition function value from Eq. 4.1. The chosen candidate is evaluated with the ASE calculator with a relatively computationally expensive method. Convergence of the algorithm requires five criteria. Firstly, the training set for the ML algorithm has to be greater than or equal to `ml_steps`.

Secondly, the maximum absolute force of an atom in the suggested structure must be smaller than `fmax`, which is set to 0.05 eV/Å in this study. Thirdly, the uncertainty prediction must be smaller than `unc_convergence`, which is set to 0.025 eV. Fourthly, the absolute energy difference between the predicted energy and true energy must be smaller than two times *unc_convergence*. Lastly, the true energy of the suggested final structure and the lowest energy observed energy, E_{min}, while performing the MLGO simulation, must be smaller than `unc_convergence`.

The used ML model is a GP. The GP uses a SEC function with derivatives and therefore trains on energies and forces. The hyperparameters are optimized by maximizing the LP with the factorization method. As a consequence, the hyperparameters are optimized robustly, but the same relative-noise hyperparameter is used for energies and forces. The hyperparameters are optimized in the logarithmic space and have prior normal distributions. The prior distribution of the length-scale has a mean of 0.0 and a standard deviation of 3.0 in the logarithmic space. The mean is -11.0, and the standard deviation is -4.0 for the prior distribution of the relative-noise hyperparameter. The prior mean is the maximum of the energies observed in the training set. Furthermore, a repulsive potential energy[47, 101] is applied as the baseline with the expression:

$$\mu(\vec{x}_*) = \mu + \sum_{i=1}^{N} \sum_{j \neq i} \left(R_c \frac{R_{c,i} + R_{c,j}}{|\vec{R}_i - \vec{R}_j|} \right)^{12} \tag{4.2}$$

where $R_c = 0.7$ is a displacement of the repulsion, $R_{c,i}$ is the covalent radius of atom i, and $\vec{R}_i$ is the Cartesian coordinates of atom i. The prior mean and the repulsive potential energy baseline ensure that atoms do not get too close in the structure search.

A simple fingerprint is also introduced. A fingerprint is essential due to the global consideration of the search. The fingerprint vector consists of blocks of pairs of chemical species. The fingerprint element in each block is the distance between two atoms within the atomic pair combination scaled with the sum of the atom's covalent radii. Each block is sorted after size. A simplification of the fingerprint is seen below for oxygen at a palladium surface:

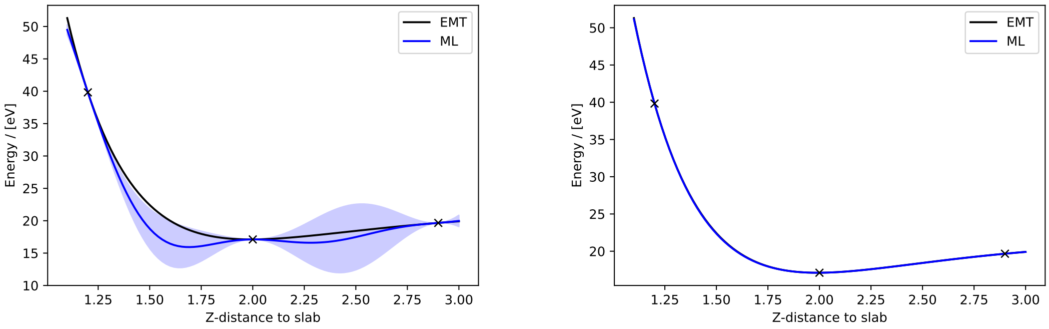
$$\phi(\vec{x}_*) = \left[\text{sort}\left(\left[\tfrac{2R_{c,Pd}}{|\vec{R}_{Pd1} - \vec{R}_{Pd2}|} \quad \tfrac{2R_{c,Pd}}{|\vec{R}_{Pd1} - \vec{R}_{Pd3}|} \quad \cdots \right] \right) \quad \text{sort}\left(\left[\tfrac{R_{c,O} + R_{c,Pd}}{|\vec{R}_{O} - \vec{R}_{Pd1}|} \quad \tfrac{R_{c,O} + R_{c,Pd}}{|\vec{R}_{O} - \vec{R}_{Pd2}|} \quad \cdots \right] \right) \right] \tag{4.3}$$

The derivatives of the inverse distance fingerprint are also needed for training and predicting forces.

Different adsorption systems are considered in this study. The GPAW code[103, 104] is used for calculating the potential energy. The BEEF-vDW[105] XC functional is used in this study if not otherwise mentioned. Plane waves are used with an energy cutoff of 500 eV, and 4 × 4 × 1 k-points are used.

4.3 Results & Discussion

The inverse distance fingerprint (see Eq. 4.3) gives chemical information on the energy dependence of the inverse distances. Therefore, fewer training points are needed to obtain the chemical information compared to the Cartesian coordinates (see Fig. 4.1). Besides learning the potential energy surface faster, the inverse distance fingerprint is also global, and therefore it is invariant to translations, rotations, and permutations. However, it must be noted that the sorting of the elements in each block may lead to jumps in the derivatives. This will not be a problem for simple adsorbates since they consist of a few identical



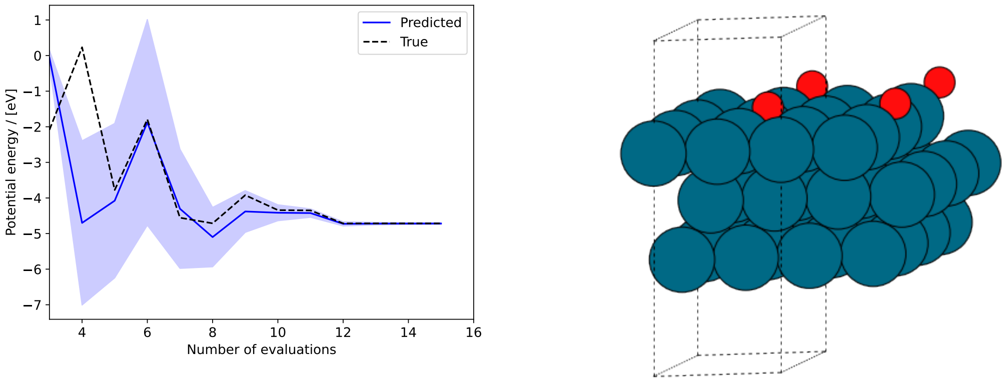
(a) Cartesian fingerprint

(b) Inverse distance fingerprint

Figure 4.1: The potential energy of an oxygen atom adsorbing at an ontop site of a palladium(111) surface. An oxygen atom is already adsorbed on another ontop site. The black curve shows the potential energy from an effective-medium theory calculation, and the blue curve shows the Gaussian process predicted energy. The blue-scaled regions indicate two times the uncertainty from the Gaussian process.

chemical elements. Different tags are used for each adsorbate and the surface. The inverse distance fingerprint understands different tags as different chemical elements. The size of the inverse distance fingerprint makes it possible to get deeper learning compared to using the sum of each block. Utilizing the forces and the fingerprint makes the GP able to understand the potential energy surface of adsorptions quickly.

The GMES of oxygen adsorption on a fixed palladium(111) surface is easily obtained with the MLGO (see Fig. 4.2). The predicted potential energies are close to the true potential

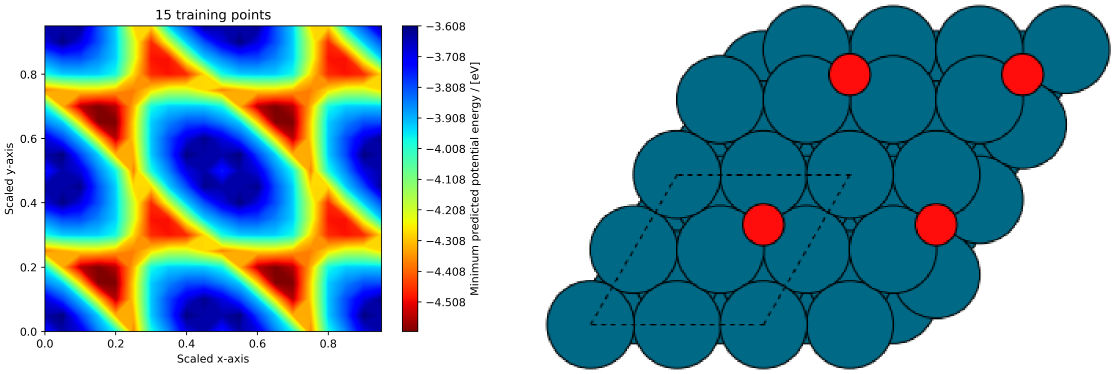


(a) Energy evaluations and predicted energies and uncertainties.

(b) Final structure from the optimization.

Figure 4.2: The global optimization log of oxygen adsorption on a fixed palladium(111) surface. Oxygen is restricted to the top half of the cell. (a) is the potential energy difference of the system as a function of the number of density functional theory evaluations. The black curve shows the true potential energy, and the blue curve shows the predicted energy. The blue-scaled regions are two times the uncertainty from the Gaussian process. (b) is the final structure of the oxygen adsorbed on the Pd(111) surface. The unit cell is repeated twice in the 1. and 2. unit cell vector directions.

energies, and the uncertainties take the prediction errors into account. However, it is observed that the training set needs to contain at least four structures for the uncertainties to be correct. 15 DFT evaluations are required to find the GMES of the oxygen adsorption. This is a small number of iterations that could be expected for a single local optimization. It requires 33 DFT evaluations in total to perform local relaxations (BFGS[82, 83, 84, 85] with fmax=0.05 eV/Å) on the ontop, fcc-hollow, hcp-hollow, and bridge sites from a good educated guess of oxygen adsorption (1.7 Å from the surface). In this case with the simple nature of the surface, it is straightforward to make educated guesses for the adsorption sites, however this is often not the case for more complex surfaces. The ML algorithm and the fingerprint learn the symmetry of the surface itself. The GMES is predicted to be the adsorption of oxygen at the fcc-hollow site, which is also the result of the local relaxations of the different sites. After the MLGO simulation, the ML calculator can be used to predict the complete potential energy surface of the adsorption of oxygen on the palladium(111) surface (see Fig. 4.3). The ML algorithm has learned that the fcc-hollow site is the most stable site, but also that the hcp-hollow site is a stable local minimum. It can also locate the bridge site and recognize that the ontop site is not a stable adsorption site. The time of the prediction of the full potential energy surface is negligible.



(a) Minimum predicted energy surface.

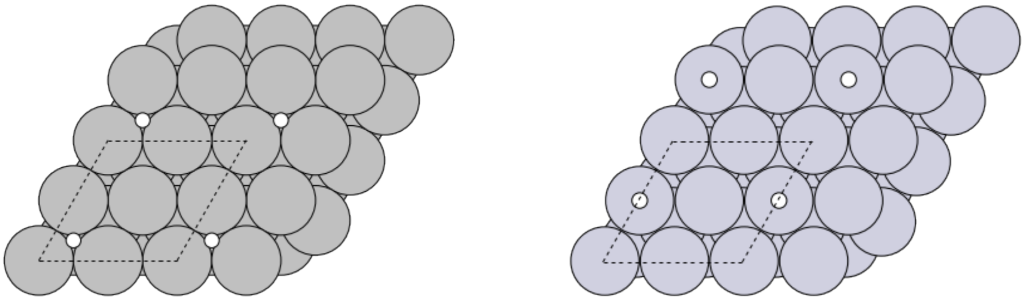
(b) Global minimum energy structure.

Figure 4.3: (a) The minimum predicted energy surface of oxygen adsorption on a fixed palladium(111) surface. The minimum potential energy is obtained from the minimum energy in the z-direction of the oxygen atom. The x-direction of the oxygen atom is scaled with cell size. The y-direction of the oxygen atom is also scaled with the second unit cell vector. (b) The global minimum energy structure (fcc-hollow site) in a top view.

The simulations of the oxygen adsorption are executed with 10 random seeds, and the true GMES is obtained in all 10 simulations. The same results are obtained with $\kappa = 2$, requiring 12 DFT evaluations. However, the exploration is important to ensure that the GMES is obtained for more complicated systems.

The error of the forces can not be too large compared to the energies when a single relative-noise hyperparameter is used for both the energies and the forces. The forces should be neglected if the errors are too large due to the concept of ”garbage in, garbage out”.

Different species can also be adsorbed on different surfaces, like hydrogen adsorption on fixed silver(111) and platinum(111) surfaces (see Fig. 4.4). The GMES of hydrogen adsorption on the silver surface is obtained after 10 DFT evaluations. The most stable adsorption site for hydrogen is the fcc-hollow site. Local relaxations of the four likely sites also show that the fcc-hollow site is the most stable. On a platinum(111) surface, the MLGO predicts the most stable adsorption site for hydrogen to be at the ontop site.



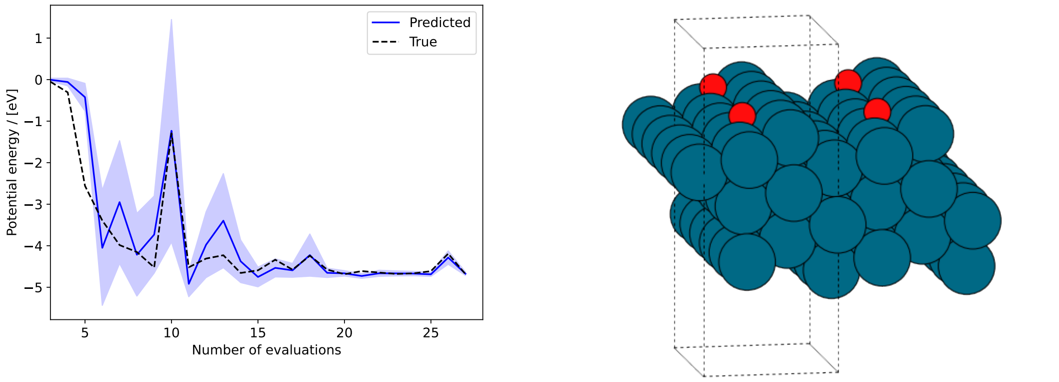
(a) Hydrogen on silver

(b) Hydrogen on platinum

Figure 4.4: The global minimum energy structures of hydrogen adsorption on a fixed silver(111) surface (a) and a platinum(111) surface (b). The unit cell is repeated twice in the 1. and 2. unit cell vector directions.

Local relaxations of hydrogen in the ontop, fcc-hollow, hcp-hollow, and bridge sites with the same ASE calculator show that the ontop site is the most stable structure. The GMES is achieved in 15 DFT evaluations. The different sites have small differences in the local minimum energies (< 0.2 eV). Thus, the MLGO shows that it can locate the GMES and learn the true potential energy surface for the adsorbate even for small energy changes. The exploration part of the MLGO is crucial if there are closely competing local minimum structures. With $\kappa = 3$, the GMES of hydrogen on Pt(111) is achieved 10 out of 10 times with different random seeds. However, the GMES is only obtained 7 out of 10 times if $\kappa = 2$.

The MLGO can easily be applied on more complicated surfaces like a stepped surface. In this study, an oxygen atom is adsorbed on a palladium(211) surface. The identification of the GMES is accomplished in 27 evaluations (see Fig. 4.5). Surprisingly, the more



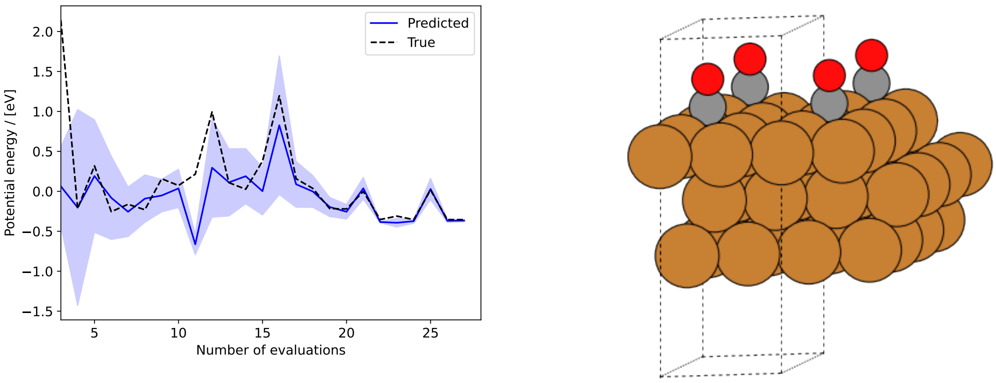
(a) Energy evaluations and predicted energies and uncertainties.

(b) Final structure from the optimization.

Figure 4.5: The global optimization log of oxygen adsorption on a fixed palladium(211) surface. Oxygen is restricted to the top half of the cell. (a) is the potential energy difference of the system as a function of the number of density functional theory evaluations. The black curve shows the true potential energy, and the blue curve shows the predicted energy. The blue-scaled regions are two times the uncertainty from the Gaussian process. (b) is the final structure of the adsorption. The unit cell is repeated twice in the 1. and 2. unit cell vector directions.

complicated surface with many nonequivalent sites does not require a large increase in MLGO iterations. The true potential energy surface is quickly learned by the ML model, and the prediction uncertainty accounts for the prediction errors. The location of the local minima will be harder to search manually since the adsorption sites for more complex surfaces are not implemented in ASE. Hence, the user needs to make educated guesses for the adsorption sites. On the other hand, the code setup of MLGO is unaffected by the surface complexity. Therefore computational and setting-up time is saved using MLGO compared to manual methods.

A GMES of adsorption of a simple molecule can also be located with the current version of MLGO. Here, a carbon monoxide molecule (CO) is adsorbed on a fixed copper(111) surface (see Fig. 4.6). The CO molecule is free to relax its bond length in the local relaxation part. The GMES is obtained in 27 evaluations. The CO molecule is adsorbed at the fcc-hollow site with the carbon end chemisorbed to the copper surface. It is also the most stable site observed from local optimizations. The ML model also learns the potential energy of the molecule and its interaction with the surface from 27 training points and with a simple fingerprint. It has also learned the most stable orientation of the CO molecule at the surface. CO in the gas phase has also been considered in the MLGO simulation.

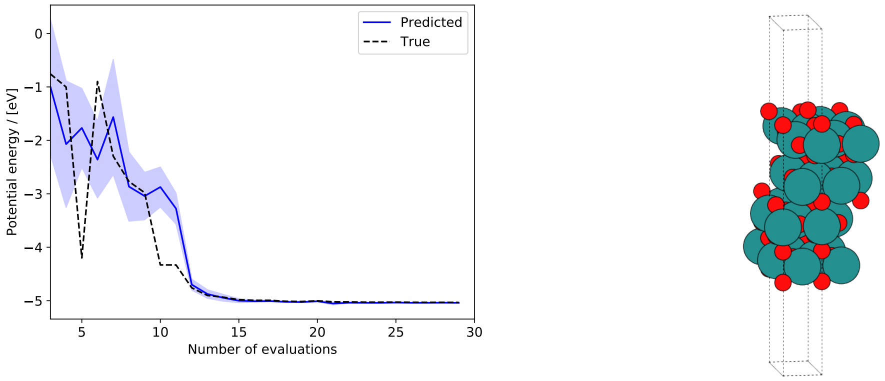


(a) Energy evaluations and predicted energies and uncertainties.

(b) Final structure from the optimization.

Figure 4.6: The global optimization log of carbon monoxide adsorption on a fixed copper(111) surface. The carbon monoxide molecule is restricted to the top half of the cell. (a) is the potential energy difference of the system as a function of the number of density functional theory evaluations. The black curve shows the true potential energy, and the blue curve shows the predicted energy. The blue-scaled regions show two times the uncertainty from the Gaussian process. (b) is the final structure of the adsorption. The unit cell is repeated twice in the first and second unit cell vector directions.

Oxygen adsorption on a ruthenium dioxide (RuO_2) surface is also investigated for testing the MLGO on more complicated systems (see Fig. 4.7). The two top layers of the RuO_2 surface are free to move, and the two lower layers are fixed. The system is calculated with GPAW with the XC functional PBE[63], 8 × 4 × 1 k-points, and an energy cutoff of 500 eV. The final structure from the MLGO simulation is the oxygen adsorbed on top of a ruthenium atom. 29 DFT evaluations have been performed to find the GMES observed, where the oxygen adsorbate and the two top layers are optimized. The adsorption of the oxygen atom on a fully fixed RuO_2 surface requires 23 DFT evaluations. Hence, the MLGO can learn the potential energy surface of adsorption on oxides even when the adsorbate is



(a) Energy evaluations and predicted energies and uncertainties.

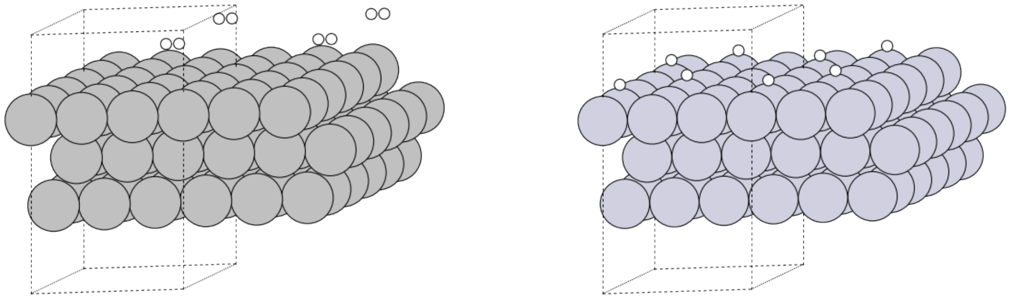
(b) Final structure from the optimization.

Figure 4.7: The global optimization log of oxygen adsorption on a ruthenium dioxide surface. The oxygen is restricted to the top half of the cell. (a) is the potential energy difference of the system as a function of the number of density functional theory evaluations. The black curve shows the true potential energy, and the blue curve shows the predicted energy. The blue-scaled regions are two times the uncertainty from the Gaussian process. (b) is the final structure of the adsorption. The unit cell is repeated twice in the 1. and 2. unit cell vector directions.

identical to atoms in the surface.

The GMES of hydroxide adsorption on the RuO_2 surface is also obtainable. The hydroxide adsorption on the fixed RuO_2 requires 54 DFT evaluations, and on the RuO_2 with two moving layers requires 45 DFT evaluations. The hydroxide is also adsorbed on top of a ruthenium atom and has a bond angle of 110.6°.

It is also possible to globally optimize two adsorbates on a surface simultaneously with MLGO. Here, two hydrogen atoms are globally optimized on a fixed silver(111) or platinum(111) surface (see Fig. 4.8). The surfaces consist of 3 × 3 × 3 atoms. Therefore, finding the GMES is a combinatorially complicated task. The hydrogen atoms can be adsorbed on different sites and different site combinations. Furthermore, the hydrogen atoms can interact, and therefore the same type of sites are nonequivalent. The hydrogen



(a) 2 hydrogen atoms silver(111) surface.

(b) 2 hydrogen atoms on platinum(111) surface.

Figure 4.8: The global minimum energy structures of two hydrogen atoms adsorption on a silver(111) surface (a) and a platinum(111) surface (b). The unit cell is repeated twice in the 1. and 2. unit cell vector directions.

atoms can also form molecular bonds, which introduces an orientational consideration. After 64 DFT evaluations, the GMES is obtained from the MLGO of two hydrogen atoms adsorbed on the silver(111) surface. It is unfavorable for the hydrogen atoms to adsorb on the silver surface when it is possible to form hydrogen gas. Thus, the final structure and the GMES is a hydrogen molecule with an optimized bond length above the silver surface. On platinum(111), on the other hand, it is found by the MLGO, that the hydrogen atoms prefer to adsorb in the ontop sites far from each other. This result is obtained with MLGO within 66 DFT evaluations. Therefore, the MLGO significantly reduces the number of iterations required for finding the GMES. It is also possible to choose an adsorption site for one of the hydrogen atoms and then optimize the second hydrogen atom with the MLGO. This approach would further reduce the number of evaluations required.

$\kappa = 3$ has shown to be a good parameter that balances the exploration and exploitation of the active learning approach. The GMESs are observed in all the systems considered, and the accelerations of the global searches are significant (see Section **??**). The repulsive potential energy baseline and the maximum energy as the prior mean have been shown to be successful in avoiding structures with too short distances.

Systems with larger cell sizes will not complicate the optimization task further for the MLGO since the fingerprint is global and will learn the same information. However, the computational time of the fingerprint and its derivatives would increase. On the other hand, the computational time for the DFT evaluation would also increase so that the relative speed-up would be similar.

4.4 Conclusion

The MLGO is a global optimization method that focuses on finding the global adsorption structure, which is a fundamental task that is always considered when a catalytic reaction is studied. By not aiming to be a general tool for all kinds of structural optimizations, the MLGO is specifically tailored to solve problems related to catalytic reactions. It has been shown how the global optimizations of common adsorbates on surfaces of varying complexity have been significantly accelerated (a reduction factor up to 40), showing that MLGO is a promising method to substantially reduce computational costs of material screenings. It is expected that all kinds of heterogeneous systems with simple adsorbates are feasible to optimize with MLGO assuming that the individual adsorbate and surface structures do not change significantly after the adsorption.

The MLGO has been shown to accelerate the global adsorption search significantly. It has especially been shown to reduce the computational cost for complicated surfaces, where the manual setup of adsorption sites is non-trivial. Furthermore, it is advantageous that the process and code do not change depending on the surface or the adsorbate considered making the method directly applicable to screening studies of different kinds of adsorbates and surfaces. The tailored training data is generated by the active learning approach, ensuring that the user is not limited by the availability of existing training data. The method is expected to significantly improve the computational cost and setup time of workflows studying changing adsorbates and surfaces.

The data and/or the ML calculator can also be reused to study the adsorptions on the surface in more detail since the potential energy surface is learned within an uncertainty. Furthermore, the ML calculator can also be used for a pre-trained Machine Learning Accelerated Nudged Elastic Band method (MLNEB) (see Section 5.1).

5 Machine Learning Accelerated Nudged Elastic Band method

5.1 Introduction

The NEB[22, 24] is the most used method for finding the activation energies and the MEPs for surface reactions. The activation energies are needed for all reaction kinetic calculations. Therefore, they are essential in catalysis simulations. However, the NEB is a computationally expensive method, and a large part of the computational time is used on those calculations.

ML has shown the potential to significantly accelerate the NEB calculations[42, 43]. In this chapter, a new implementation of the Machine Learning Accelerated Nudged Elastic Band method (MLNEB) code developed in Ref. [43] is presented. The new implementation includes the new and more robust ML model in the form of the TP. The new MLNEB code is tested on a range of different catalytic reactions with the EMT and DFT calculators. The energy barriers obtained with MLNEB are compared to the corresponding results from the NEB.

5.2 Method

MLNEB is written in Python with the same class structure as the NEB code implemented in ASE[102]. The MLNEB requires the initial and final state structures with an ASE calculator as input like a regular NEB. Furthermore, it requires a ML calculator with the same form as an ASE calculator that can be trained, predicts energies, predicts forces, and estimates uncertainties. A default ML calculator is applied if a ML calculator is not given. Furthermore, it requires an acquisition function object, an interpolation method, a specified number of images, and a local optimizer for the NEB simulation. The MLNEB uses One-Image-Evaluation method (OIE), which means that only a single image is evaluated with the ASE calculator for each MLNEB iteration. The image that maximizes an acquisition function is evaluated. The acquisition object calculates the chosen acquisition function for each image and returns the image with the largest acquisition function value. A useful acquisition function that is used in this work is the uncertainty if the uncertainty is greater than a selected uncertainty convergence criterion, `unc_convergence`, and else it will be the upper confidence bound as the uncertainty times a value, κ, added to the energy:

$$a(\vec{x}_i) = \begin{cases} \sigma_*(\vec{x}_i) & \text{if } \sigma_*(\vec{x}_i) \geq \texttt{unc\_convergence} \\ E(\vec{x}_i) + \kappa\sigma_*(\vec{x}_i) & \text{otherwise} \end{cases} \tag{5.1}$$

The interpolation method constructs the initial path that is optimized. The interpolation method can be a linear interpolation between the initial and final state, Image Dependent Pair Potential (IDPP) that makes a good initial guess of the path using pairwise distances[106], or a manually constructed initial path. The number of images, `n_images`, in the MEP is set to 11 if it is not specified. The local optimizer is set to the MDMin optimizer implemented in ASE[102]. The MDMin uses MD to relax the structure. The time step must be small since the energy and structure can be unstable. Especially at the beginning of the MLNEB simulation, unstable structures can be suggested when the ML calculator is not trained fully. The MLNEB also takes the regular NEB input parameters as arguments. The NEB input parameters include the method, which is the improved tangent method[22] in this study, and the spring constant, k_s. The spring constant is given as $k_s = 2\sqrt{\texttt{n\_images}}/D_{IF}$ where D_{IF} is the distance between the initial and final states if a spring constant is not given. When the MLNEB simulation is initialized, a set of parameters can also be specified. `fmax` is the argument that specifies the convergence criterion for the maximum absolute force of an atom in the last iteration, which is set to 0.05 eV/Å as default. `unc_convergence` is the maximum uncertainty, $\max(\sigma_*(\vec{x}_i))$, an image can have on the last iteration to converge. Finally, `max_unc` sets the maximum uncertainty an image can have and continues a NEB simulation on the predicted PES.

The pseudo-code for the MLNEB can be seen in Alg. 2. The run of MLNEB is initialized

Algorithm 2 MLNEB

```
Require: Initial state, final state, ASE calculator, ML calculator.
Ensure: MEP images
  Calculate a third structure
  converged ← False
  $i \leftarrow 0$
  while converged ≠ True do
      $i \leftarrow i + 1$
      Train ML calculator
      Construct initial path
      max_u ← $\frac{\texttt{max\_unc}(i-1)+\texttt{unc\_convergence}}{i}$
      while $\max(\sigma_*(\vec{x}_i)) \leq$ max_u do
          NEB step with ML calculator
      end while
      if NEB converged then
          if $\max(\sigma_*(\vec{x}_i)) \leq$ max_u then
              CI-NEB with ML calculator
          end if
      end if
      Chose candidate from acquisition function
      Evaluation of candidate with ASE calculator
      if $|F_i| \leq$ fmax then
          if $\max(\sigma_*(\vec{x}_i)) \leq$ unc_convergence then
              converged ← True
          end if
      end if
  end while
```

by calculating a third training point, besides from the initial and final states, if a training set is not given in advance. The third training point is selected from the initial path. Then, the ML calculator is trained. The initial path is constructed with the ML calculator as the calculator for each image. The NEB simulation is performed, and the predicted uncertainties are checked for each NEB iteration. The NEB simulation will stop if a single uncertainty is greater than `max_u`. `max_u` is a scaled uncertainty criterion of `max_unc` and `unc_convergence` that ensure that the uncertainties are small at the beginning of the simulation. A Climbing Image Nudged Elastic Band method (CI-NEB)[24] simulation is performed if the NEB simulation converges. Whether or not the CI-NEB simulation is performed, the next candidate for evaluation with the ASE calculator is decided by the acquisition object. At last, a convergence check is performed. The MLNEB simulation is converged if the maximum force of the atoms in the last candidate is less than `fmax` and the uncertainties on all the images are less than `unc_convergence`. Furthermore, the difference between the true and predicted energy of the last evaluated candidate has to be less than in `unc_convergence`. Therefore, the acquisition object must decide on the image with the largest energy at the end.

The ML model used as default, and in this study, is the TP. The TP is especially good at prediction uncertainties at few training points compared to a GP, which is important at the beginning of the MLNEB simulation. The maximum energy in the training set is used as the prior mean constant for the TP. The hyperparameters of the TP are tuned by maximizing the LP with the factorization method every time the ML calculator is trained. The prior distribution of the length-scale hyperparameter in logarithmic space is a normal distribution with a mean value of 0.0 and a standard deviation of 3.0. Similarly, the normal distribution is used for the relative-noise hyperparameter in the logarithmic space with a mean value of -11.0 and a standard deviation of -4.0. The prior distributions are chosen to give a reasonable estimate of the hyperparameter when atomic systems are considered with a small noise in data at a few training points. The Cartesian coordinates of the moving atoms are used as the fingerprints. The Cartesian coordinates of the fixed atoms are not included in the fingerprints. All evaluated data through the MLNEB simulation are used as training data.

5.2.1 Test systems

The MLNEB is tested on 9 test systems. The test systems are:

1. A diffusion of a gold atom from a hollow site to a neighboring hollow site of a fixed aluminum(100) surface[43] (AuAl)

2. A heptamer island of platinum atoms that diffuse on a fixed platinum(111) surface[43] (Heptamer)

3. Adsorption and dissociation of a hydrogen molecule onto the fcc sites of a fixed copper(111) surface (H2Cufcc)

4. Adsorption and dissociation of a hydrogen molecule onto the hcp sites of a fixed copper(111) surface (H2Cuhcp)

5. The Müller-Brown test system[75] (MB)

6. Adsorption and dissociation of a nitrogen molecule onto the fcc sites of a fixed copper(111) surface (N2Cufcc)

7. Adsorption and dissociation of a nitrogen molecule onto the hcp sites of a fixed copper(111) surface (N2Cuhcp)

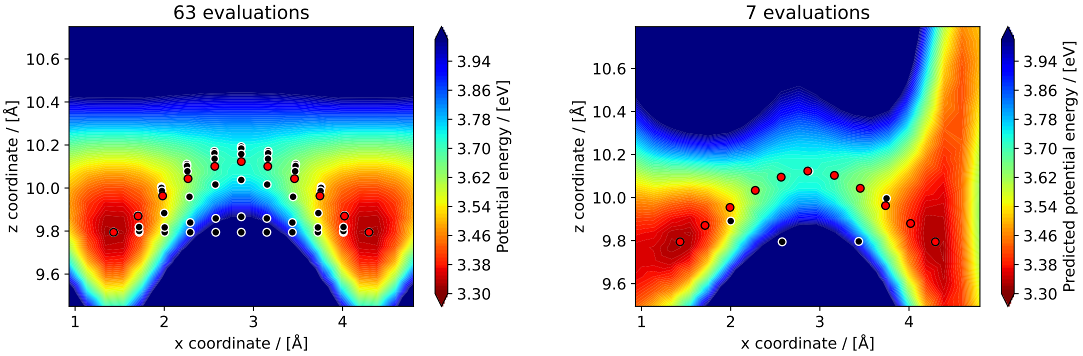
8. Oxadiazoline molecule formation from ethene and Nitrous oxide molecules (Oxad)

9. Diffusion of a platinum atom on a platinum terrace surface [43] (TerPt)

The potential energies and forces of the test systems AuAl, Heptamer, and TerPt are calculated with EMT[107, 108]. GPAW with the XC functional RPBE[109] and plane waves are used to calculate the potential energies and forces for H2Cufcc, H2Cuhcp, N2Cufcc, and N2Cuhcp. GPAW with the XC functional PBE[63] and double zeta linear combinations of atomic orbitals are used as a calculator for Oxad. Computational inexpensive methods are chosen for the test systems for proof of concept and speed rather than accurate results of the activation energies of the test systems.

5.3 Results & Discussion

A clear advantage of the MLNEB compared to the regular NEB is that it can use the OIE. The OIE therefore saves `n_images` − 3 evaluations for each iteration compared to the All-Image-Evaluation method (AIE) method. An example of the influence of the OIE can be seen in Fig. 5.1. Only the data points with the most information are evaluated. The predicted PES is not perfect in all regions, but it is correct and has a small uncertainty in the regions of interest. The obtained activation energy and the MEP from the MLNEB is the true path. 7 data points are evaluated and required for the AuAl test system when the MLNEB is used. Thus, a significant reduction in the number of evaluations is obtained compared to the NEB that requires 63 evaluations on the AuAl test system. The seven evaluated data points are all in the region of interest.



(a) NEB

(b) MLNEB

Figure 5.1: The minimum energy path and the potential energy surface for the x- and z-coordinate of a gold atom diffusing on an aluminum(100) surface. The path in figure (a) is the result of the regular nudged elastic band method that requires 63 evaluations. The path in figure (b) is from the machine learning accelerated nudged elastic band method that requires 7 evaluations. The red dots are the final minimum energy path, and the black dots are the evaluated points.

The acquisition function and the stability of the MLNEB are dependent on the quality of the uncertainty prediction. More evaluations of the true PES are required if the uncertainties are wrong. The stability of the MLNEB depends on the quality of the uncertainty predictions since the images could move too far into unknown regions. To remedy this, a trust radius could be applied instead of uncertainty predictions. Though the trust radius is simpler, the trust radius does not account for the information from the forces. Furthermore, a random evaluation of one of the images or the AIE method must be applied instead if the trust radius is used within a simple approach.

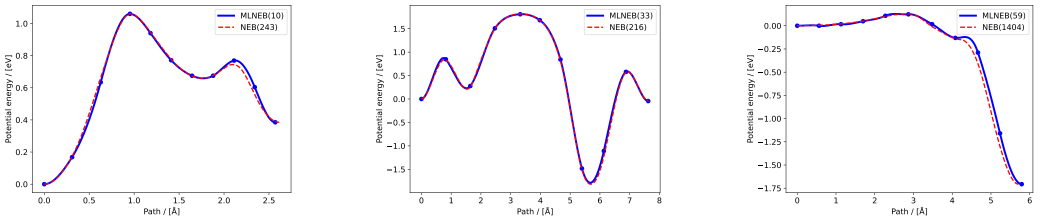
The initial guess of the MEP is very important for a regular NEB and MLNEB. A poor initial guess gives rise to a lot of extra NEB iterations on the true and predicted PES. Furthermore, a poor initial path will also force the ML algorithm to use evaluations and time to learn an unnecessary region of the chemical space. It is also not certain that the NEB can find the MEP if the initial path is poor. The NEB simulation is a local optimization, and it is therefore dependent on the initial guess. Small step sizes in the local optimization are necessary for a stable NEB simulation and especially for the MLNEB. The MLNEB has the same disadvantages and advantages as the regular NEB with respect to the local optimization.

Systems	True barrier / [eV]	Pred. barrier / [eV]	Max. unc. / [eV]	NEB evaluations	MLNEB evaluations
AuAl	0.40	0.40	0.00	63	7
Heptamer	0.91	0.89	0.03	441	40
H2Cufcc	0.13	0.12	0.03	1404	59
H2Cuhcp	0.13	0.13	0.01	1305	54
MB	1.06	1.06	0.05	243	10
N2Cufcc	3.75	3.75	0.05	7353	38
N2Cuhcp	3.86	3.86	0.02	4850	37
Oxad	0.65	0.65	0.04	1665	55
TerPt	1.82	1.81	0.03	216	33

Table 5.1: The true activation barriers calculated with the nudged elastic band method compared to the predicted activation barriers calculated with the machine learning accelerated nudged elastic band method together. The required numbers of evaluations for both methods are also listed. The maximum uncertainty at the last iteration is also listed.

All the predicted activation energies from MLNEB matches the true activation energies from NEB within the pre-defined uncertainty criterion (smaller than 0.05 eV) with the largest error being 0.02 eV (see Table 5.1). The required number of evaluations for finding the MEP is reduced with a factor from 6 to 189 with the MLNEB compared to the corresponding NEB results. The adsorption and dissociation of the hydrogen and nitrogen molecules required small time steps in the local optimization to converge.

However, the training of the ML model also takes time and especially when the training set becomes large. The training time of the ML model is larger than the computational cost of the analytical PES from EMT and the MB potential energy. However, the training time is not as computationally expensive as DFT evaluations and especially not with parameters that give more accurate results. The aim of this study is to test the method to illustrate the robustness of very different systems. Hence, higher accuracy of the DFT calculations with higher computational cost would be redundant. The DFT data shows that the MLNEB also works for DFT data, which can have noises in the energies and forces. However, the noises of the forces can not be too large since a single relative-noise hyperparameter is used for energy and forces.



(a) MB

(b) TerPt

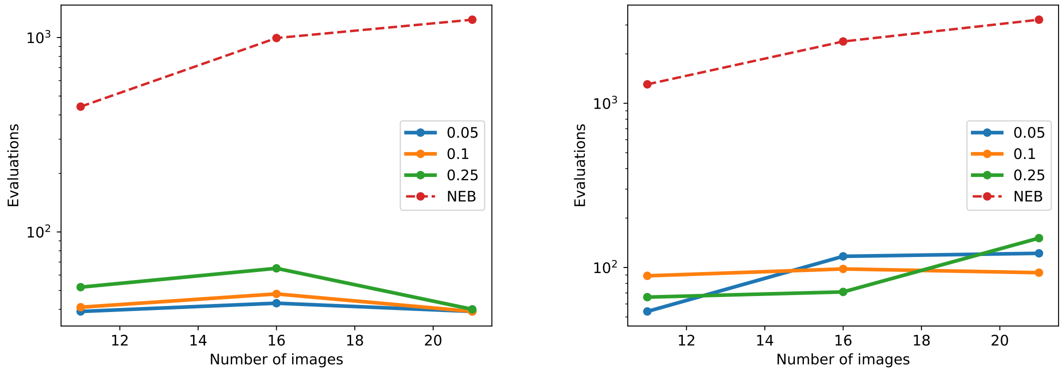
(c) H2Cufcc

Figure 5.2: The minimum energy paths obtained from regular nudged elastic band method simulations (red dashed curves) and machine learning accelerated nudged elastic band method simulations (blue curves with the images as dots). The number of evaluations for each method is shown in the brackets.

The MEP from the MLNEB is also similar to the MEP from the NEB (see Fig. 5.2). The structure of the SP is represented correctly and therefore also the activation energy. Small deviations can be observed in the MEP. Fortunately, the deviations can be avoided with stricter uncertainty criterion if the precise MEP is important.

The MLNEB requires an activation barrier larger than 0.0 eV since the maximum force of an atom in the last iteration must be less than the `fmax` convergence criterion. `fmax` in the MLNEB is not the same as the one often used in the NEB that uses the maximum force along the MEP.

The number of evaluations required for a regular NEB scale with the number of images due to the AIE (see Fig. 5.3). Therefore, a higher resolution of the MEP requires more evaluations with the NEB. Contrary, the number of evaluations in the MLNEB does not scale with the number of images due to OIE. However, the number of evaluations needed for achieving convergence can change nonlinearly with the number of images. Thus, a high resolution of the MEP is obtainable at the same computational cost. Furthermore, the ML calculator is reusable after the simulation, and the PES of the region of interest can be predicted with uncertainty.



(a) Heptamer (b) H2Cuhcp

Figure 5.3: The number of evaluations performed as a function of the number of images used. The evaluations from the nudged elastic band method (red curves), the machine learning accelerated nudged elastic band method with `max_unc` = 0.05 (blue curve), `max_unc` = 0.10 (orange curve), and with `max_unc` = 0.25 (blue curve) is compared. Figure (a) shows the number of evaluations performed on the Heptamer test system with `unc_convergence` = 0.025. Figure (b) shows the H2Cuhcp test system with `unc_convergence` = 0.050.

The Cartesian coordinates are sufficient for the MLNEB since the NEB is a local optimization. The structures can change significantly within the MEP, and a general trend can be more complex to learn than the position dependence.

The prior mean constant greatly influences the stability of the MLNEB. It is advantageous if the prior mean constant is greater than the activation energy. However, a prior mean constant with the value of the initial or final state energy does exploration instead of exploitation in the MEP.

5.4 Conclusion

In this study, the implemented MLNEB has been shown to significantly reduce the number of evaluations required for finding the MEP. The evaluation reduction factor is around 5-200 for the test systems studied. The reduction in the number of evaluations comes without a large price of precision since the predicted accuracy can be tuned by an uncertainty criterion that matches the prediction error. Furthermore, the SP structure obtained from MLNEB is within a maximum force criterion that ensures the right activation energy.

Furthermore, it has been demonstrated that MLNEB can successfully predict reaction paths and barriers for very different kinds of systems, ranging from metal diffusion to molecular reactions. This shows, that the MLNEB has the same degree of systemic flexibility as the regular NEB. It should be noted that the MLNEB requires an energy barrier larger than the reaction energy for convergence. The MLNEB can learn from data calculated from analytical EMT and MB PESs with no noise in energies and forces as well as DFT calculated energies and forces with noises. Therefore, it is a robust method that finds the MEP within the same restrictions as the NEB. The number of evaluations required for achieving convergence depends on its parameters.

The number of moving atoms needs to be considered in the MLNEB since its computational cost scales cubically with the coordinates of the moving atoms, similarly to the cubic scaling with the number of atoms of DFT.

The region of interest for the reaction path is then obtained from active learning within the MLNEB simulation without predefining a database. Afterward, the ML calculator can then be used to get a higher resolution of the MEP.

Better scaling of the computational cost as a function of the number of training data and the number of moving atoms is required to study very complex systems. This can be achieved by using one or more reduced databases instead of one extensive database. However, the selection of the data points must be considered. A reoccurring problem is that the active learning process can be trapped in the suggestion of the same training point that is being removed from the reduced database. Furthermore, a mixture or linear combination of models with reduced databases can be used. However, the prediction means and uncertainties need to be combined. This is the focus of future work.

6 Summary

In this thesis, the Gaussian Process (GP) has been optimized to obtain a robust model with good uncertainty predictions. The GP and a newly introduced Machine Learning (ML) model have been implemented in two algorithms. The first new algorithm significantly accelerates global adsorption searches for catalysis simulations, which is advantageous in, e.g. high-throughput screening studies. The Nudged Elastic Band method (NEB) is the most common way of finding the important Minimum Energy Path (MEP) of surface reactions. However, the process is often computationally costly. The second algorithm presented herein is a new implementation of the Machine Learning Accelerated Nudged Elastic Band method (MLNEB), a robust method to significantly accelerate NEB calculations.

In chapter 2, the theory of the GP is explained. Furthermore, useful equations are introduced for a robust GP. The factorization of the covariance matrix is treated as the standard approach. Thus, the covariance matrix is independent of the prefactor hyperparameter, and the relative-noise hyperparameter replaces the noise hyperparameter. The factorization gives a more transparent understanding of the hyperparameters effect and a more robust inversion of the covariance matrix. The Log-Likelihood (LL) is expressed by the analytical solution of the prefactor hyperparameter. Eigendecomposition of the factorized covariance matrix permits variations in the relative-noise hyperparameter without inversion of the covariance matrix for every variation. Furthermore, the new Student's T Process (TP) is derived from a Bayesian approach of the prefactor hyperparameter. At last, a GP that mimics the fully Bayesian solution of the posterior predictive distribution is derived by the use of Kullback–Leibler divergence (KL).

In the most common way of optimizing the GP, several problems often occur, e.g. exception errors in the optimization, overfitting due to plateaus, and underfitting due to other plateaus. In chapter 3, methods and notes for avoiding these problems were discussed. To make optimization of the GP possible without exception error, which is shown to ensure a stable optimization. A description of the hyperparameter space and a variable transformation increases the probability of finding the global maximum of the LL. A new and greatly improved method for the optimization of the hyperparameters is introduced and implemented. The method finds the global maximums of the LLs for all test systems with different training set sizes. The new method gives robustness to the GP that regular local and global optimizers do not achieve. Furthermore, the new method requires a lower computational cost than other global optimizers. A new measure (Uncertainty Deviation (UD)) for evaluating the uncertainty predictions is also established. Different objective functions are evaluated and discussed with measures of the prediction means and uncertainties. New modifications of already existing objective functions improve their predictive qualities. The new TP is also discussed and evaluated as an improvement to the GP. At last, the new Fully Bayesian Mimicking Gaussian Process (FBMGP) is discussed and compared to the Maximum Likelihood Estimation (MLE), Maximum A Posteriori estimation (MAP), and fully Bayesian solutions. The FBMGP has proved significantly better for uncertainty predictions.

In chapter 4, the robust GP was implemented into a new algorithm, called Machine Learning Accelerated Global Adsorption Optimization method (MLGO), for finding the global minimum energy adsorption structure. The MLGO performs a global optimization on the surrogate surface that would be unfeasible on the true Potential Energy Surface (PES).

A simple fingerprint of inverse distances is implemented to learn the PES faster globally. The algorithm was tested on different surface test systems containing various elements and facets as well as different types of adsorbates, including single atoms, small molecules, and simultaneously adsorbed species. The Global Minimum Energy Structure (GMES) is consistently achieved for all the test systems considered. The acceleration of the global search is significantly faster than the standard methods. Furthermore, the setup of the code is independent of the structure and adsorbate, which reduces the manual programming time. Due to the active learning approach, no database is required for the ML model. The advantages of the MLGO algorithm make it well suited for, e.g, automatic workflows.

A new version of the MLNEB is implemented and discussed in chapter 5. The new TP is implemented into the code as a reliable ML model. The MLNEB uses active learning for constructing the most useful database. The complete database is applied to optimize the GP to ensure a stable model. Different catalysis reactions have been investigated to ensure the stability of the method. A substantial reduction factor of 5-200 is obtained in the number of evaluations required for getting the Minimum Energy Path (MEP) compared to the regular Nudged Elastic Band method (NEB). As NEB is the most commonly used method for finding the MEP, which is crucial in estimating reaction products and rates, the improved MLNEB developed in this thesis has the potential to greatly reduce computational time and resources for a wide variety of catalytic surface reactions.

Bibliography

[1] Hannah Ritchie, Max Roser, and Pablo Rosado. “Energy”. In: *Our World in Data* (2022). URL: https://ourworldindata.org/energy.

[2] “Emissions Gap Report 2022: The Closing Window — Climate crisis calls for rapid transformation of societies”. In: *United Nations Environment Programme* (2022). URL: https://www.unep.org/emissions-gap-report-2022.

[3] Nathan J. L. Lenssen et al. “Improvements in the GISTEMP Uncertainty Model”. In: *Journal of Geophysical Research: Atmospheres* 124 (12 June 2019), pp. 6307–6326. ISSN: 2169-897X. DOI: 10.1029/2018JD029522. URL: https://onlinelibrary.wiley.com/doi/10.1029/2018JD029522.

[4] GISTEMP Team. *GISS Surface Temperature Analysis (GISTEMP), version 4*. URL: https://data.giss.nasa.gov/gistemp/.

[5] “Transforming our world: the 2030 Agenda for Sustainable Development”. In: *United Nations* (2015). URL: https://sdgs.un.org/2030agenda.

[6] Kirstin Alberi et al. “The 2019 materials by design roadmap”. In: *Journal of Physics D: Applied Physics* 52 (1 Jan. 2019), p. 013001. ISSN: 0022-3727. DOI: 10.1088/1361-6463/aad926. URL: https://iopscience.iop.org/article/10.1088/1361-6463/aad926.

[7] Helge S. Stein and John M. Gregoire. “Progress and prospects for accelerating materials science with automated and autonomous workflows”. In: *Chemical Science* 10 (42 Nov. 2019), pp. 9640–9649. ISSN: 2041-6520. DOI: 10.1039/C9SC03766G. URL: http://xlink.rsc.org/?DOI=C9SC03766G.

[8] Nathan S. Lewis and Daniel G. Nocera. “Powering the planet: Chemical challenges in solar energy utilization”. In: *Proceedings of the National Academy of Sciences* 103 (43 Oct. 2006), pp. 15729–15735. ISSN: 0027-8424. DOI: 10.1073/pnas.0603395103. URL: https://pnas.org/doi/full/10.1073/pnas.0603395103.

[9] Ivano E. Castelli et al. “Computational screening of perovskite metal oxides for optimal solar light capture”. In: *Energy Environ. Sci.* 5 (2 Jan. 2012), pp. 5814–5819. ISSN: 1754-5692. DOI: 10.1039/C1EE02717D. URL: http://xlink.rsc.org/?DOI=C1EE02717D.

[10] Elina Buitrago, Anna Maria Novello, and Thierry Meyer. “Third-Generation Solar Cells: Toxicity and Risk of Exposure”. In: *Helvetica Chimica Acta* 103 (9 Sept. 2020), e2000074. ISSN: 0018-019X. DOI: 10.1002/hlca.202000074. URL: https://onlinelibrary.wiley.com/doi/10.1002/hlca.202000074.

[11] Ivano E Castelli, Thomas Olsen, and Yunzhong Chen. “Towards photoferroic materials by design: recent progress and perspectives”. In: *Journal of Physics: Energy* 2 (1 Jan. 2020), p. 011001. ISSN: 2515-7655. DOI: 10.1088/2515-7655/ab428c. URL: https://iopscience.iop.org/article/10.1088/2515-7655/ab428c.

[12] Maximilian Fichtner et al. “Rechargeable Batteries of the Future—The State of the Art from a BATTERY 2030+ Perspective”. In: *Advanced Energy Materials* 12 (17 May 2022), p. 2102904. ISSN: 1614-6832. DOI: 10.1002/aenm.202102904. URL: https://onlinelibrary.wiley.com/doi/10.1002/aenm.202102904.

[13] Zhi Wei Seh et al. “Combining theory and experiment in electrocatalysis: Insights into materials design”. In: *Science* 355 (6321 Jan. 2017). ISSN: 0036-8075. DOI: 10.1126/science.aad4998. URL: https://www.science.org/doi/10.1126/science.aad4998.

[14] Rahman Daiyan, Iain MacGill, and Rose Amal. “Opportunities and Challenges for Renewable Power-to-X”. In: *ACS Energy Letters* 5 (12 Dec. 2020), pp. 3843–3847.

ISSN: 2380-8195. DOI: 10.1021/acsenergylett.0c02249. URL: https://pubs.acs.org/doi/10.1021/acsenergylett.0c02249.

[15] Joseph H. Montoya et al. “Materials for solar fuels and chemicals”. In: *Nature Materials* 16 (1 Jan. 2017), pp. 70–81. ISSN: 1476-1122. DOI: 10.1038/nmat4778. URL: https://www.nature.com/articles/nmat4778.

[16] Kevin Tran and Zachary W. Ulissi. “Active learning across intermetallics to guide discovery of electrocatalysts for CO2 reduction and H2 evolution”. In: *Nature Catalysis* 1 (9 Sept. 2018), pp. 696–703. ISSN: 2520-1158. DOI: 10.1038/s41929-018-0142-1. URL: https://www.nature.com/articles/s41929-018-0142-1.

[17] Philomena Schlexer Lamoureux et al. “Machine Learning for Computational Heterogeneous Catalysis”. In: *ChemCatChem* 11 (16 Aug. 2019), pp. 3581–3601. ISSN: 1867-3880. DOI: 10.1002/cctc.201900595. URL: https://onlinelibrary.wiley.com/doi/10.1002/cctc.201900595.

[18] Surya R. Kalidindi, Andrew J. Medford, and David L. McDowell. “Vision for Data and Informatics in the Future Materials Innovation Ecosystem”. In: *JOM* 68 (8 Aug. 2016), pp. 2126–2137. ISSN: 1047-4838. DOI: 10.1007/s11837-016-2036-5. URL: http://link.springer.com/10.1007/s11837-016-2036-5.

[19] E. Schrödinger. “An Undulatory Theory of the Mechanics of Atoms and Molecules”. In: *Physical Review* 28 (6 Dec. 1926), pp. 1049–1070. ISSN: 0031-899X. DOI: 10.1103/PhysRev.28.1049. URL: https://link.aps.org/doi/10.1103/PhysRev.28.1049.

[20] Chunyang Peng and H. Bernhard Schlegel. “Combining Synchronous Transit and Quasi-Newton Methods to Find Transition States”. In: *Israel Journal of Chemistry* 33 (4 Jan. 1993), pp. 449–454. ISSN: 00212148. DOI: 10.1002/ijch.199300051. URL: https://onlinelibrary.wiley.com/doi/10.1002/ijch.199300051.

[21] 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 Oct. 1999), pp. 7010–7022. ISSN: 0021-9606. DOI: 10.1063/1.480097. URL: http://aip.scitation.org/doi/10.1063/1.480097.

[22] Graeme Henkelman and Hannes Jónsson. “Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points”. In: *The Journal of Chemical Physics* 113 (22 Dec. 2000), pp. 9978–9985. ISSN: 0021-9606. DOI: 10.1063/1.1323224. URL: http://aip.scitation.org/doi/10.1063/1.1323224.

[23] HANNES JÓNSSON, GREG MILLS, and KARSTEN W. JACOBSEN. “Nudged elastic band method for finding minimum energy paths of transitions”. In: WORLD SCIENTIFIC, June 1998, pp. 385–404. ISBN: 978-981-02-3498-0. DOI: 10.1142/9789812839664_0016. URL: http://www.worldscientific.com/doi/abs/10.1142/9789812839664_0016.

[24] 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 Dec. 2000), pp. 9901–9904. ISSN: 0021-9606. DOI: 10.1063/1.1329672. URL: http://aip.scitation.org/doi/10.1063/1.1329672.

[25] Zachary W. Ulissi et al. “To address surface reaction network complexity using scaling relations machine learning and DFT calculations”. In: *Nature Communications* 8 (1 Mar. 2017), p. 14621. ISSN: 2041-1723. DOI: 10.1038/ncomms14621. URL: https://www.nature.com/articles/ncomms14621.

[26] R. Car and M. Parrinello. “Unified Approach for Molecular Dynamics and Density-Functional Theory”. In: *Physical Review Letters* 55 (22 Nov. 1985), pp. 2471–2474. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.55.2471. URL: https://link.aps.org/doi/10.1103/PhysRevLett.55.2471.

[27] Alessandro Laio and Michele Parrinello. “Escaping free-energy minima”. In: *Proceedings of the National Academy of Sciences* 99 (20 Oct. 2002), pp. 12562–12566. ISSN: 0027-8424. DOI: 10.1073/pnas.202427399. URL: https://pnas.org/doi/full/10.1073/pnas.202427399.

[28] Stefan Grimme. “Exploration of Chemical Compound, Conformer, and Reaction Space with Meta-Dynamics Simulations Based on Tight-Binding Quantum Chemical Calculations”. In: *Journal of Chemical Theory and Computation* 15 (5 May 2019), pp. 2847–2862. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.9b00143. URL: https://pubs.acs.org/doi/10.1021/acs.jctc.9b00143.

[29] Mads Koerstz, Maria H. Rasmussen, and Jan H. Jensen. “Fast and automated identification of reactions with low barriers: the decomposition of 3-hydroperoxypropanal”. In: *SciPost Chemistry* 1 (1 Oct. 2021), p. 003. ISSN: 2772-6762. DOI: 10.21468/SciPostChem.1.1.003. URL: https://scipost.org/10.21468/SciPostChem.1.1.003.

[30] Maria H. Rasmussen and Jan H. Jensen. “Fast and automated identification of reactions with low barriers using meta-MD simulations”. In: *PeerJ Physical Chemistry* 4 (Mar. 2022), e22. ISSN: 2689-7733. DOI: 10.7717/peerj-pchem.22. URL: https://peerj.com/articles/pchem-22.

[31] Xin Yang et al. “Neural Network Potentials for Accelerated Metadynamics of Oxygen Reduction Kinetics at Au-Water Interfaces”. In: (Nov. 2022). DOI: https://doi.org/10.26434/chemrxiv-2022-b1pt5. URL: https://chemrxiv.org/engage/chemrxiv/article-details/63848f3e0949e1dd4f58c527.

[32] Ivano E. Castelli et al. “New Light-Harvesting Materials Using Accurate and Efficient Bandgap Calculations”. In: *Advanced Energy Materials* 5 (2 Jan. 2015), p. 1400915. ISSN: 16146832. DOI: 10.1002/aenm.201400915. URL: https://onlinelibrary.wiley.com/doi/10.1002/aenm.201400915.

[33] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. “Inverse molecular design using machine learning: Generative models for matter engineering”. In: *Science* 361 (6400 July 2018), pp. 360–365. ISSN: 0036-8075. DOI: 10.1126/science.aat2663. URL: https://www.science.org/doi/10.1126/science.aat2663.

[34] Marc H. Garner et al. “The Bicyclo[2.2.2]octane Motif: A Class of Saturated Group 14 Quantum Interference Based Single-Molecule Insulators”. In: *The Journal of Physical Chemistry Letters* 9 (24 Dec. 2018), pp. 6941–6947. ISSN: 1948-7185. DOI: 10.1021/acs.jpclett.8b03432. URL: https://pubs.acs.org/doi/10.1021/acs.jpclett.8b03432.

[35] Felix T. Bölle et al. “Autonomous Discovery of Materials for Intercalation Electrodes”. In: *Batteries Supercaps* 3 (6 June 2020), pp. 488–498. ISSN: 2566-6223. DOI: 10.1002/batt.201900152. URL: https://onlinelibrary.wiley.com/doi/10.1002/batt.201900152.

[36] Mads Koerstz et al. “High throughput virtual screening of 230 billion molecular solar heat battery candidates”. In: *PeerJ Physical Chemistry* 3 (Feb. 2021), e16. ISSN: 2689-7733. DOI: 10.7717/peerj-pchem.16. URL: https://peerj.com/articles/pchem-16.

[37] Joerg Schaarschmidt et al. “Workflow Engineering in Materials Design within the BATTERY 2030 <b>+</b> Project”. In: *Advanced Energy Materials* 12 (17 May 2022), p. 2102638. ISSN: 1614-6832. DOI: 10.1002/aenm.202102638. URL: https://onlinelibrary.wiley.com/doi/10.1002/aenm.202102638.

[38] Zeyu Deng et al. “Towards autonomous high-throughput multiscale modelling of battery interfaces”. In: *Energy Environmental Science* 15 (2 Feb. 2022), pp. 579–594. ISSN: 1754-5692. DOI: 10.1039/D1EE02324A. URL: http://xlink.rsc.org/?DOI=D1EE02324A.

[39] Jörg Behler and Michele Parrinello. "Generalized neural-network representation of high-dimensional potential-energy surfaces". In: *Physical Review Letters* 98 (14 Apr. 2007), p. 146401. ISSN: 00319007. DOI: 10.1103/PhysRevLett.98.146401. URL: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.98.146401.

[40] Venkatesh Botu and Rampi Ramprasad. "Adaptive machine learning framework to accelerate ab initio molecular dynamics". In: *International Journal of Quantum Chemistry* 115 (16 Aug. 2015), pp. 1074–1083. ISSN: 00207608. DOI: 10.1002/qua.24836. URL: https://onlinelibrary.wiley.com/doi/10.1002/qua.24836.

[41] Alireza Khorshidi and Andrew A. Peterson. "Amp: A modular approach to machine learning in atomistic simulations". In: *Computer Physics Communications* 207 (Oct. 2016), pp. 310–324. ISSN: 00104655. DOI: 10.1016/j.cpc.2016.05.010. URL: https://linkinghub.elsevier.com/retrieve/pii/S0010465516301266.

[42] Olli-Pekka Koistinen et al. "Nudged elastic band calculations accelerated with Gaussian process regression". In: *The Journal of Chemical Physics* 147 (15 Oct. 2017), p. 152720. ISSN: 0021-9606. DOI: 10.1063/1.4986787. URL: http://aip.scitation.org/doi/10.1063/1.4986787.

[43] José A. Garrido Torres et al. "Low-Scaling Algorithm for Nudged Elastic Band Calculations Using a Surrogate Machine Learning Model". In: *Physical Review Letters* 122 (15 Apr. 2019), p. 156001. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.122.156001. URL: https://link.aps.org/doi/10.1103/PhysRevLett.122.156001.

[44] Estefanía Garijo del Río, Jens Jørgen Mortensen, and Karsten Wedel Jacobsen. "Local Bayesian optimizer for atomic structures". In: *Physical Review B* 100 (10 Sept. 2019), p. 104103. ISSN: 2469-9950. DOI: 10.1103/PhysRevB.100.104103. URL: https://link.aps.org/doi/10.1103/PhysRevB.100.104103.

[45] Sebastian Dick and Marivi Fernandez-Serra. "Machine learning accurate exchange and correlation functionals of the electronic density". In: *Nature Communications* 11 (1 July 2020), p. 3509. ISSN: 2041-1723. DOI: 10.1038/s41467-020-17265-7. URL: https://www.nature.com/articles/s41467-020-17265-7.

[46] Chiara Panosetti et al. "Learning to Use the Force: Fitting Repulsive Potentials in Density-Functional Tight-Binding with Gaussian Process Regression". In: *Journal of Chemical Theory and Computation* 16 (4 Apr. 2020), pp. 2181–2191. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.9b00975. URL: https://pubs.acs.org/doi/10.1021/acs.jctc.9b00975.

[47] Malthe K. Bisbo and Bjørk Hammer. "Efficient Global Structure Optimization with a Machine-Learned Surrogate Model". In: *Physical Review Letters* 124 (8 Feb. 2020), p. 086102. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.124.086102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.124.086102.

[48] Sami Kaappa, Casper Larsen, and Karsten Wedel Jacobsen. "Atomic Structure Optimization with Machine-Learning Enabled Interpolation between Chemical Elements". In: *Physical Review Letters* 127 (16 Oct. 2021), p. 166001. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.127.166001. URL: https://link.aps.org/doi/10.1103/PhysRevLett.127.166001.

[49] Christopher M Bishop. *Pattern Recognition and Machine Learning.* 1st ed. Springer New York, NY, 2006. ISBN: 978-0-387-31073-2. URL: https://link.springer.com/book/9780387310732%20papers2://publication/uuid/05A8B4CF-0248-4692-8B1D-DCC065B79465.

[50] Anders Krogh and Jesper Vedelsby. "Neural Network Ensembles, Cross Validation, and Active Learning". In: *Advances in neural information processing systems* 7 (1994). URL: https://proceedings.neurips.cc/paper/1994/file/b8c37e33defde51cf91e1e03e51657da-Paper.pdf.

[51] Juhwan Noh et al. “Active learning with non- <i>ab initio</i> input features toward efficient CO <sub>2</sub> reduction catalysts”. In: *Chemical Science* 9 (23 June 2018), pp. 5152–5159. ISSN: 2041-6520. DOI: 10.1039/C7SC03422A. URL: http://xlink.rsc.org/?DOI=C7SC03422A.

[52] A. O’Hagan. “Curve Fitting and Optimal Design for Prediction”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 40 (1 Sept. 1978), pp. 1–24. ISSN: 00359246. DOI: 10.1111/j.2517-6161.1978.tb01643.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1978.tb01643.x.

[53] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Nov. 2005. ISBN: 9780262256834. DOI: 10.7551/mitpress/3206.001.0001. URL: https://direct.mit.edu/books/book/2320/gaussian-processes-for-machine-learning.

[54] 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 Dec. 2019), pp. 6738–6751. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.9b00692. URL: https://pubs.acs.org/doi/10.1021/acs.jctc.9b00692.

[55] M. Born and R. Oppenheimer. “Zur Quantentheorie der Molekeln”. In: *Annalen der Physik* 389 (20 Jan. 1927), pp. 457–484. ISSN: 00033804. DOI: 10.1002/andp.19273892002. URL: https://onlinelibrary.wiley.com/doi/10.1002/andp.19273892002.

[56] L. H. Thomas. “The calculation of atomic fields”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 23 (5 Jan. 1927), pp. 542–548. ISSN: 0305-0041. DOI: 10.1017/S0305004100011683. URL: https://www.cambridge.org/core/product/identifier/S0305004100011683/type/journal_article.

[57] P. Hohenberg and W. Kohn. “Inhomogeneous Electron Gas”. In: *Physical Review* 136 (3B Nov. 1964), B864–B871. ISSN: 0031-899X. DOI: 10.1103/PhysRev.136.B864. URL: https://link.aps.org/doi/10.1103/PhysRev.136.B864.

[58] W. Kohn and L. J. Sham. “Self-Consistent Equations Including Exchange and Correlation Effects”. In: *Physical Review* 140 (4A Nov. 1965), A1133–A1138. ISSN: 0031-899X. DOI: 10.1103/PhysRev.140.A1133. URL: https://link.aps.org/doi/10.1103/PhysRev.140.A1133.

[59] F. Bloch. “Bemerkung zur Elektronentheorie des Ferromagnetismus und der elektrischen Leitfuhigkeit”. In: *Zeitschrift fur Physik* 57 (7-8 July 1929), pp. 545–555. ISSN: 1434-6001. DOI: 10.1007/BF01340281. URL: http://link.springer.com/10.1007/BF01340281.

[60] P. A. M. Dirac. “Note on Exchange Phenomena in the Thomas Atom”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 26 (3 July 1930), pp. 376–385. ISSN: 0305-0041. DOI: 10.1017/S0305004100016108. URL: https://www.cambridge.org/core/product/identifier/S0305004100016108/type/journal_article.

[61] A. D. Becke. “Density-functional exchange-energy approximation with correct asymptotic behavior”. In: *Physical Review A* 38 (6 Sept. 1988), pp. 3098–3100. ISSN: 0556-2791. DOI: 10.1103/PhysRevA.38.3098. URL: https://link.aps.org/doi/10.1103/PhysRevA.38.3098.

[62] Chengteh Lee, Weitao Yang, and Robert G. Parr. “Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density”. In: *Physical Review B* 37 (2 Jan. 1988), pp. 785–789. ISSN: 0163-1829. DOI: 10.1103/PhysRevB.37.785. URL: https://link.aps.org/doi/10.1103/PhysRevB.37.785.

[63] John P. Perdew, Kieron Burke, and Yue Wang. “Generalized gradient approximation for the exchange-correlation hole of a many-electron system”. In: *Physical*

Review B 54 (23 Dec. 1996), pp. 16533–16539. ISSN: 0163-1829. DOI: 10.1103/PhysRevB.54.16533. URL: https://link.aps.org/doi/10.1103/PhysRevB.54.16533.

[64] Axel D. Becke. “A new mixing of Hartree–Fock and local density-functional theories”. In: *The Journal of Chemical Physics* 98 (2 Jan. 1993), pp. 1372–1377. ISSN: 0021-9606. DOI: 10.1063/1.464304. URL: http://aip.scitation.org/doi/10.1063/1.464304.

[65] D. R. Hartree. “The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part II. Some Results and Discussion”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 24 (1 1928), pp. 111–132. ISSN: 1469-8064. DOI: 10.1017/S0305004100011920. URL: https://www.cambridge.org/core/journals/mathematical-proceedings-of-the-cambridge-philosophical-society/article/wave-mechanics-of-an-atom-with-a-noncoulomb-central-field-part-ii-some-results-and-discussion/5916E7A0DEC0A051B435688BE2ACD57E.

[66] D R Hartree and W Hartree. “Self-consistent field, with exchange, for beryllium”. In: *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences* 150 (869 May 1935), pp. 9–33. ISSN: 0080-4630. DOI: 10.1098/rspa.1935.0085. URL: https://royalsocietypublishing.org/doi/10.1098/rspa.1935.0085.

[67] J. C. Slater. “Note on Hartree’s Method”. In: *Physical Review* 35 (2 Jan. 1930), pp. 210–211. ISSN: 0031-899X. DOI: 10.1103/PhysRev.35.210.2. URL: https://link.aps.org/doi/10.1103/PhysRev.35.210.2.

[68] V. Fock. “Näherungsmethode zur Lösung des quantenmechanischen Mehrkörperproblems”. In: *Zeitschrift für Physik* 61 (1-2 Jan. 1930), pp. 126–148. ISSN: 14346001. DOI: 10.1007/BF01340294/METRICS. URL: https://link.springer.com/article/10.1007/BF01340294.

[69] Karsten W. Jacobsen. *Probabilistic machine learning.* Jan. 2022.

[70] Benoit ’ Commandant. “Note Sur Une Méthode de Résolution des équations Normales Provenant de L’Application de la MéThode des Moindres Carrés a un Système D’équations Linéaires en Nombre Inférieur a Celui des Inconnues. — Application de la Méthode a la Résolution D’un Système Defini D’éQuations LinéAires”. In: *Bulletin Géodésique* 2 (1 Apr. 1924), pp. 67–77. ISSN: 0007-4632. DOI: 10.1007/BF03031308. URL: http://link.springer.com/10.1007/BF03031308.

[71] Estefanía Garijo del Río et al. “Machine learning with bond information for local structure optimizations in surface science”. In: *The Journal of Chemical Physics* 153 (23 Dec. 2020), p. 234116. ISSN: 0021-9606. DOI: 10.1063/5.0033778. URL: http://aip.scitation.org/doi/10.1063/5.0033778.

[72] Vidhi Lalchand and Carl Edward Rasmussen. “Approximate Inference for Fully Bayesian Gaussian Process Regression”. In: (Dec. 2019), pp. 1–12. DOI: https://doi.org/10.48550/arXiv.1912.13440. URL: http://arxiv.org/abs/1912.13440.

[73] Fergus Simpson, Vidhi Lalchand, and Carl Edward Rasmussen. “Marginalised Gaussian Processes with Nested Sampling”. In: *Advances in Neural Information Processing Systems* 17 (Oct. 2020), pp. 13613–13625. ISSN: 10495258. DOI: 10.48550/arxiv.2010.16344. URL: http://arxiv.org/abs/2010.16344.

[74] S. Kullback and R. A. Leibler. “On Information and Sufficiency”. In: *The Annals of Mathematical Statistics* 22 (1 Mar. 1951), pp. 79–86. ISSN: 0003-4851. DOI: 10.1214/aoms/1177729694. URL: http://projecteuclid.org/euclid.aoms/1177729694.

[75] Klaus Müller and Leo D. Brown. “Location of saddle points and minimum energy paths by a constrained simplex optimization procedure”. In: *Theoretica Chimica Acta* 53 (1 Mar. 1979), pp. 75–93. ISSN: 0040-5744. DOI: 10.1007/BF00547608. URL: http://link.springer.com/10.1007/BF00547608.

[76] S. Sundararajan and S. S. Keerthi. “Predictive Approaches for Choosing Hyperparameters in Gaussian Processes”. In: *Neural Computation* 13 (5 May 2001), pp. 1103–1118. ISSN: 0899-7667. DOI: 10.1162/08997660151134343. URL: https://direct.mit.edu/neco/article/13/5/1103-1118/6513.

[77] Pauli Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17 (3 Mar. 2020), pp. 261–272. ISSN: 1548-7091. DOI: 10.1038/s41592-019-0686-2. URL: http://www.nature.com/articles/s41592-019-0686-2.

[78] J. A. Nelder and R. Mead. “A Simplex Method for Function Minimization”. In: *The Computer Journal* 7 (4 Jan. 1965), pp. 308–313. ISSN: 0010-4620. DOI: 10.1093/comjnl/7.4.308. URL: https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/7.4.308.

[79] Fuchang Gao and Lixing Han. “Implementing the Nelder-Mead simplex algorithm with adaptive parameters”. In: *Computational Optimization and Applications* 51 (1 Jan. 2012), pp. 259–277. ISSN: 09266003. DOI: 10.1007/S10589-010-9329-3/METRICS. URL: https://link.springer.com/article/10.1007/s10589-010-9329-3.

[80] M. J. D. Powell. “An efficient method for finding the minimum of a function of several variables without calculating derivatives”. In: *The Computer Journal* 7 (2 Jan. 1964), pp. 155–162. ISSN: 0010-4620. DOI: 10.1093/COMJNL/7.2.155. URL: https://academic.oup.com/comjnl/article/7/2/155/335330.

[81] M.R. Hestenes and E. Stiefel. “Methods of conjugate gradients for solving linear systems”. In: *Journal of Research of the National Bureau of Standards* 49 (6 Dec. 1952), p. 409. ISSN: 0091-0635. DOI: 10.6028/jres.049.044. URL: https://nvlpubs.nist.gov/nistpubs/jres/049/jresv49n6p409_A1b.pdf.

[82] C. G. BROYDEN. “The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations”. In: *IMA Journal of Applied Mathematics* 6 (1 Mar. 1970), pp. 76–90. ISSN: 0272-4960. DOI: 10.1093/imamat/6.1.76. URL: https://academic.oup.com/imamat/article-lookup/doi/10.1093/imamat/6.1.76.

[83] FLETCHER R. “A new approach to variable metric algorithms”. In: *The Computer Journal* 13 (3 Jan. 1970), pp. 317–322. ISSN: 0010-4620. DOI: 10.1093/COMJNL/13.3.317. URL: https://academic.oup.com/comjnl/article/13/3/317/345520.

[84] Donald Goldfarb. “A family of variable-metric methods derived by variational means”. In: *Mathematics of Computation* 24 (109 1970), pp. 23–26. ISSN: 0025-5718. DOI: 10.1090/S0025-5718-1970-0258249-6. URL: https://www.ams.org/mcom/1970-24-109/S0025-5718-1970-0258249-6/.

[85] D. F. Shanno. “Conditioning of quasi-Newton methods for function minimization”. In: *Mathematics of Computation* 24 (111 1970), pp. 647–656. ISSN: 0025-5718. DOI: 10.1090/S0025-5718-1970-0274029-X. URL: https://www.ams.org/mcom/1970-24-111/S0025-5718-1970-0274029-X/.

[86] Richard H. Byrd et al. “A Limited Memory Algorithm for Bound Constrained Optimization”. In: *SIAM Journal on Scientific Computing* 16 (5 Sept. 1995), pp. 1190–1208. ISSN: 1064-8275. DOI: 10.1137/0916069. URL: http://epubs.siam.org/doi/10.1137/0916069.

[87] Ron S. Dembo and Trond Steihaug. “Truncated-newtono algorithms for large-scale unconstrained optimization”. In: *Mathematical Programming* 26 (2 June 1983), pp. 190–212. ISSN: 0025-5610. DOI: 10.1007/BF02592055. URL: http://link.springer.com/10.1007/BF02592055.

[88] 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 July 1997), pp. 5111–

5116. ISSN: 1089-5639. DOI: 10.1021/jp970984n. URL: https://pubs.acs.org/doi/10.1021/jp970984n.

[89] Constantino Tsallis. "Possible generalization of Boltzmann-Gibbs statistics". In: *Journal of Statistical Physics* 52 (1-2 July 1988), pp. 479–487. ISSN: 0022-4715. DOI: 10.1007/BF01016429. URL: http://link.springer.com/10.1007/BF01016429.

[90] Constantino Tsallis and Daniel A. Stariolo. "Generalized simulated annealing". In: *Physica A: Statistical Mechanics and its Applications* 233 (1-2 Nov. 1996), pp. 395–406. ISSN: 03784371. DOI: 10.1016/S0378-4371(96)00271-3. URL: https://linkinghub.elsevier.com/retrieve/pii/S0378437196002713.

[91] Y Xiang et al. "Generalized simulated annealing algorithm and its application to the Thomson model". In: *Physics Letters A* 233 (3 Aug. 1997), pp. 216–220. ISSN: 03759601. DOI: 10.1016/S0375-9601(97)00474-X. URL: https://linkinghub.elsevier.com/retrieve/pii/S037596019700474X.

[92] Y. Xiang and X. G. Gong. "Efficiency of generalized simulated annealing". In: *Physical Review E* 62 (3 Sept. 2000), pp. 4473–4476. ISSN: 1063-651X. DOI: 10.1103/PhysRevE.62.4473. URL: https://link.aps.org/doi/10.1103/PhysRevE.62.4473.

[93] J. Kiefer. "Sequential minimax search for a maximum". In: *Proceedings of the American Mathematical Society* 4 (3 1953), pp. 502–506. ISSN: 0002-9939. DOI: 10.1090/S0002-9939-1953-0055639-3. URL: https://www.ams.org/proc/1953-004-03/S0002-9939-1953-0055639-3/.

[94] Seymour Geisser. "The Predictive Sample Reuse Method with Applications". In: *Journal of the American Statistical Association* 70 (350 June 1975), p. 320. ISSN: 01621459. DOI: 10.2307/2285815. URL: https://www.jstor.org/stable/2285815?origin=crossref.

[95] Seymour Geisser and William F. Eddy. "A Predictive Approach to Model Selection". In: *Journal of the American Statistical Association* 74 (365 Mar. 1979), pp. 153–160. ISSN: 0162-1459. DOI: 10.1080/01621459.1979.10481632. URL: http://www.tandfonline.com/doi/abs/10.1080/01621459.1979.10481632.

[96] Andrew A. Peterson. "Global Optimization of Adsorbate–Surface Structures While Preserving Molecular Identity". In: *Topics in Catalysis* 57 (1-4 Feb. 2014), pp. 40–53. ISSN: 1022-5528. DOI: 10.1007/s11244-013-0161-8. URL: http://link.springer.com/10.1007/s11244-013-0161-8.

[97] Bernd Hartke. "Global geometry optimization of clusters using genetic algorithms". In: *The Journal of Physical Chemistry* 97 (39 Sept. 1993), pp. 9973–9976. ISSN: 0022-3654. DOI: 10.1021/j100141a013. URL: https://pubs.acs.org/doi/abs/10.1021/j100141a013.

[98] 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 July 2014), p. 044711. ISSN: 0021-9606. DOI: 10.1063/1.4886337. URL: http://aip.scitation.org/doi/10.1063/1.4886337.

[99] Stefan Goedecker. "Minima hopping: An efficient search method for the global minimum of the potential energy surface of complex molecular systems". In: *The Journal of Chemical Physics* 120 (21 June 2004), pp. 9911–9917. ISSN: 0021-9606. DOI: 10.1063/1.1724816. URL: http://aip.scitation.org/doi/10.1063/1.1724816.

[100] Mathias S. Jørgensen et al. "Exploration versus Exploitation in Global Atomistic Structure Optimization". In: *The Journal of Physical Chemistry A* 122 (5 Feb. 2018), pp. 1504–1509. ISSN: 1089-5639. DOI: 10.1021/acs.jpca.8b00160. URL: https://pubs.acs.org/doi/10.1021/acs.jpca.8b00160.

[101] Sami Kaappa, Estefanía Garijo del Río, and Karsten Wedel Jacobsen. "Global optimization of atomic structures with gradient-enhanced Gaussian process regres-

sion". In: *Physical Review B* 103 (17 May 2021), p. 174114. ISSN: 2469-9950. DOI: 10.1103/PhysRevB.103.174114.

[102] Ask Hjorth Larsen et al. "The atomic simulation environment—a Python library for working with atoms". In: *Journal of Physics: Condensed Matter* 29 (27 July 2017), p. 273002. ISSN: 0953-8984. DOI: 10.1088/1361-648X/aa680e. URL: https://iopscience.iop.org/article/10.1088/1361-648X/aa680e.

[103] J. J. Mortensen, L. B. Hansen, and K. W. Jacobsen. "Real-space grid implementation of the projector augmented wave method". In: *Physical Review B* 71 (3 Jan. 2005), p. 035109. ISSN: 1098-0121. DOI: 10.1103/PhysRevB.71.035109. URL: https://link.aps.org/doi/10.1103/PhysRevB.71.035109.

[104] J Enkovaara 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 June 2010), p. 253202. ISSN: 0953-8984. DOI: 10.1088/0953-8984/22/25/253202. URL: https://iopscience.iop.org/article/10.1088/0953-8984/22/25/253202.

[105] Jess Wellendorff et al. "Density functionals for surface science: Exchange-correlation model development with Bayesian error estimation". In: *Physical Review B - Condensed Matter and Materials Physics* 85 (23 June 2012), p. 235149. ISSN: 10980121. DOI: 10.1103/PHYSREVB.85.235149/FIGURES/9/MEDIUM. URL: https://journals.aps.org/prb/abstract/10.1103/PhysRevB.85.235149.

[106] Søren Smidstrup et al. "Improved initial guess for minimum energy path calculations". In: *The Journal of Chemical Physics* 140 (21 June 2014), p. 214106. ISSN: 0021-9606. DOI: 10.1063/1.4878664. URL: http://aip.scitation.org/doi/10.1063/1.4878664.

[107] J. K. Nørskov and N. D. Lang. "Effective-medium theory of chemical binding: Application to chemisorption". In: *Physical Review B* 21 (6 Mar. 1980), pp. 2131–2136. ISSN: 0163-1829. DOI: 10.1103/PhysRevB.21.2131. URL: https://link.aps.org/doi/10.1103/PhysRevB.21.2131.

[108] Zhigang Xi et al. "An effective-medium theory approach to ordering in Cu-Au alloys". In: *Journal of Physics: Condensed Matter* 4 (35 Aug. 1992), pp. 7191–7202. ISSN: 0953-8984. DOI: 10.1088/0953-8984/4/35/005. URL: https://iopscience.iop.org/article/10.1088/0953-8984/4/35/005.

[109] B. Hammer, L. B. Hansen, and J. K. Nørskov. "Improved adsorption energetics within density-functional theory using revised Perdew-Burke-Ernzerhof functionals". In: *Physical Review B* 59 (11 Mar. 1999), pp. 7413–7421. ISSN: 0163-1829. DOI: 10.1103/PhysRevB.59.7413. URL: https://link.aps.org/doi/10.1103/PhysRevB.59.7413.

[110] Martin Hangaard Hansen et al. "An Atomistic Machine Learning Package for Surface Science and Catalysis". In: (Apr. 2019). DOI: 10.48550/arxiv.1904.00904. URL: https://arxiv.org/abs/1904.00904v1.

[111] H. J. C. Berendsen et al. "Molecular dynamics with coupling to an external bath". In: *The Journal of Chemical Physics* 81 (8 Oct. 1984), pp. 3684–3690. ISSN: 0021-9606. DOI: 10.1063/1.448118. URL: http://aip.scitation.org/doi/10.1063/1.448118.

A Appendix

A.1 Fully Bayesian Mimicking Gaussian Process derivation

The Kullback–Leibler divergence (KL) of a single GP to the fully Bayesian solution can be derived by using that the posterior predictive distribution of a GP is a Gaussian distribution. Two important equations before the derivation are the expected mean and variance of a Gaussian distribution:

$$\mu = \int_{-\infty}^{\infty} y\mathcal{N}(y|\mu,\sigma^2)\ \mathrm{d}y \tag{A.1}$$

$$\sigma^2 = \int_{-\infty}^{\infty} (y-\mu)^2\mathcal{N}(y|\mu,\sigma^2)\ \mathrm{d}y \tag{A.2}$$

Thereby, the derivation of Eq. 2.49 in Section 2.5 is performed as:

$$\begin{aligned} D_{FB} =& \int_{-\infty}^{\infty} p(y_* \mid \vec{y}) \ln\left(\frac{p(y_* \mid \vec{y})}{p(y_* \mid \vec{y}, \vec{\theta_0})}\right)\mathrm{d}y_* \\ &\propto - \int_{-\infty}^{\infty} p(y_* \mid \vec{y}) \ln\Big(p(y_* \mid \vec{y}, \vec{\theta_0})\Big)\mathrm{d}y_* \\ &=\frac{1}{2N_c}\sum_{i,j,r}\tilde{c}([\mathbf{\Theta}]_{ijr},\vec{y}) \int_{-\infty}^{\infty} \mathcal{N}(y_*|\bar{y}_*([\mathbf{\Theta}]_{ijr}),\sigma_*^2([\mathbf{\Theta}]_{ijr}))\frac{(y_*-\bar{y}_*(\vec{\theta_0}))^2}{\sigma_*^2(\vec{\theta_0})}\mathrm{d}y_* \\ &\quad+\frac{1}{2N_c}\sum_{i,j,r}\tilde{c}([\mathbf{\Theta}]_{ijr},\vec{y}) \int_{-\infty}^{\infty} \mathcal{N}(y_*|\bar{y}_*([\mathbf{\Theta}]_{ijr}),\sigma_*^2([\mathbf{\Theta}]_{ijr}))\ln\left(2\pi\sigma_*^2(\vec{\theta_0})\right)\mathrm{d}y_* \\ &=\frac{1}{2N_c}\sum_{\vec{\theta_i}}\tilde{c}(\vec{\theta_i},\vec{y})\left(\frac{\sigma_*^2(\vec{\theta_i})+(\bar{y}_*(\vec{\theta_i})-\bar{y}_*(\vec{\theta_0}))^2}{\sigma_*^2(\vec{\theta_0})}+\ln\left(2\pi\sigma_*^2(\vec{\theta_0})\right)\right) \\ &=\frac{1}{2N_c\sigma_*^2(\vec{\theta_0})}\sum_{\vec{\theta_i}}\tilde{c}(\vec{\theta_i},\vec{y})\Big(\sigma_*^2(\vec{\theta_i})+(\bar{y}_*(\vec{\theta_i})-\bar{y}_*(\vec{\theta_0}))^2\Big)+\frac{1}{2}\ln\left(2\pi\sigma_*^2(\vec{\theta_0})\right) \end{aligned} \tag{A.3}$$

A.2 The test systems

The test systems used in Chapter 3 are described here.

A.2.1 Optimization of hyperparameters

The test systems used in Chapter 3 are described here.

Müller-Brown

The MB test system[75] is an analytical energy surface calculated with the implementation in *CatLearn*[110]. Therefore, it has no noise in its energy calculations. The test system is constructed from a linear grid in two dimensions with 30 points in each. The grid in the x-direction is from -1.4 Å to 0.2 Å. The grid in the y-direction extends from 0.0 Å to 1.9 Å.

Au at Al

The AuAl test systems is calculated with the ASE [102] and the EMT calculator[107, 108]. The aluminum(100) fcc slab consists of $3 \times 3 \times 4$ fixed aluminum atoms with a gold atom located in different positions above the surface. The locations of the gold atom are constructed from a grid of 12 points in each of the x- and y-dimension and 6 points in the z-dimension. The x-grid ranges from 0.0 Å to 5.0 Å. The y-grid ranges from 0.0 Å to 3.2 Å. The z-grid ranges from 0.0 Å to 2.0 Å. The grid points are relative movements of the gold atom initially located at $(1.0, 3.0, 12.65)$.

CO at Ni with EMT
The CONi test system consists of a nickel(100) fcc slab with $3 \times 3 \times 5$ fixed nickel atoms, a carbon atom adsorbed in the hollow site at 1.8 Å above the surface, and an oxygen atom adsorbed at the on-top site at 1.7 Å above the surface. A molecular dynamics (MD)[26] calculation in ASE with the EMT calculator within the canonical ensemble (NVT) is performed. The Berendsen thermostat[111] at 800 K for 800 steps of 0.5 fs is performed.

CO at Ni with PBE
The system is the same as the CONi with EMT test system except the oxygen atom is initialized in another hollow site than the carbon atom. Furthermore, the energies and forces are calculated with DFT performed in GPAW[103, 104] with RPBE[109]. The default parameters are used in GPAW.

Copper clusters
Two clusters of copper atoms are constructed by MD simulations with the same parameters as for CONi. The copper clusters consist of 5 (Cu5) and 13 copper (Cu13) atoms. The energies and forces are calculated with EMT. The initial structure of Cu5 is built from fcc of Cu(111) with size $2 \times 2 \times 1$ with a bridged copper atom at 2 Å above the four atoms. The Cu13 is also an fcc structure of Cu(111) with size $2 \times 2 \times 3$ and a bridged copper atom.

O_2 at platinum
Two oxygen atoms adsorbed at on-top sites of a platinum(100) surface are also as a database. The platinum atoms are fixed and are constructed as fcc surface with $3 \times 3 \times 3$ atoms. The energies and forces are calculated with PBE in GPAW. A MD is performed with the same parameters as for the CONi database.

Water molecules at platinum
A database of four water molecules on a platinum(111) surface with $3 \times 2 \times 3$ atoms is constructed with a MD. All atoms move. The energies and forces are calculated with PBE in GPAW. The same MD parameters are used as in the CONi database.

A.3 Local optimization parameters

All the parameters tested for the local optimizers can be seen in Table A.1. The parameters were discussed in Section 3.2.2 and Section 3.3.2.

Parameters	**Local optimizer**	**Values**
Tolerance (`tol`)	All	[$1.0 \cdot 10^{-3}$, $1.0 \cdot 10^{-8}$, $1.0 \cdot 10^{-12}$]
Maximum iterations (`maxiter`)	All	[500, 5000]
Adapting the parameters to dimensionality (`adaptive`)	Nelder-Mead	[False, True]
Metric corrections for the limited memory matrix (`maxcor`)	L-BFGS-B	[5, 10, 15]
Maximum line search steps (`maxls`)	L-BFGS-B	[10, 20, 30]
Hessian times vector evaluations per iteration (`maxCGit`)	TNC	[-1, 0, 4]
Quality of line search (`eta`)	TNC	[0.1, 0.25, 0.5]
Scaling factor in log10 (`rescale`)	TNC	[0.1, 1.3, 3]

Table A.1: The parameters investigated for getting the best local optimizer implemented in *SciPy* are listed here[77].

A.4 Global optimization parameters

All the parameters tested for the global optimizers can be seen in Table A.2. The parameters were discussed in Section 3.2.3 and Section 3.3.2.

Parameters	**Global optimizer**	**Values**
Use educated guess boundary conditions (`use_bound`)	Random sampling, Grid Search, Line search, Factorization	[False, True]
Number of random sampled points (`npoint`)	Random sampling	[10, 20, 30]
Number of grid points in each dimension (`npoint`)	Grid search	[10, 12, 15]
Local optimize in the endpoint (`optimize`)	Grid search, Line search, Factorization	[False, True]
Number of grid points in each dimension (`npoint`)	Line search	[50, 100, 150]
Number of loops over all dimensions (`loop`)	Grid search	[False, True]
Number of jumps (`niter`)	Basin	[10, 15, 20]
How often to update stepsize (`interval`)	Basin	[5, 10, 15]
Temperature (`T`)	Basin	[0.1, 1.0, 10.0]
Maximum stepsize (`stepsize`)	Basin	[0.01, 0.01, 1.0]
Number for convergence (`niter_success`)	Basin	[5,20]
Initial temperature (`initial_temp`)	Annealing	[1000, 5230, 10000]
Ratio for restart temperature (`restart_temp_ratio`)	Annealing	[1e-8, 2e-5, 1e-2]
Visiting distribution (`visit`)	Annealing	[1.1, 2.62, 2.9]
Local optimizations (`no_local_search`)	Annealing	[False,True]
Number of grid points in each dimension (`ngrid`)	Factorization	[50, 80, 100]
Search for multiple maxima (`multiple_max`)	Factorization	[False,True]

Table A.2: The parameters[77] investigated for getting the best global optimizer.

A.5 Global optimization of hyperparameters

The average success rates, times, and iterations of the tested optimizers in Section 3.3.3. The average success rates when derivatives are not used are shown in Table A.3.

Method	Success rate	Time / [s]	Iterations
Local	0.863 (0.040,1.000)	0.071 (0.016,0.275)	52.813
Local prior	0.926 (0.050,1.000)	0.120 (0.028,0.412)	88.521
Local educated	0.933 (0.040,1.000)	0.132 (0.026,0.573)	97.827
Grid search	0.982 (0.000,1.000)	1.314 (0.716,3.550)	1760.612
Line search	0.900 (0.110,1.000)	0.709 (0.386,1.783)	940.271
Basin	0.877 (0.040,1.000)	0.484 (0.086,1.787)	345.061
Random sampling	0.999 (0.560,1.000)	1.304 (0.279,4.808)	1022.864
Annealing	0.940 (0.010,1.000)	4.586 (3.000,10.115)	5006.558
Annealing MLE	0.998 (0.470,1.000)	4.331 (2.744,9.671)	5016.133
Factorization	1.000 (1.000,1.000)	0.587 (0.303,1.555)	111.657

Table A.3: Table of the average success rate of finding the global maxima of the log-likelihood for 8 test systems each with 7 training set sizes and 8 random seeds with different optimizers. The average time and iterations are also shown. The brackets identify the smallest and largest value observed. The derivatives of the targets (forces) are not used.

The average success rates when derivatives are used are shown in Table A.4.

Method	Success rate	Time / [s]	Iterations
Local	0.745 (0.120,1.000)	0.254 (0.017,5.423)	53.019
Local prior	0.839 (0.150,1.000)	0.399 (0.038,8.471)	87.262
Local educated	0.908 (0.120,1.000)	0.484 (0.027,9.382)	107.854
Grid search	0.918 (0.000,1.000)	3.777 (0.662,57.362)	1763.690
Line search	0.753 (0.070,1.000)	1.891 (0.356,28.222)	940.151
Basin	0.768 (0.130,1.000)	1.513 (0.106,28.798)	357.300
Random sampling	0.999 (0.870,1.000)	4.477 (0.311,91.681)	1044.934
Annealing	0.906 (0.090,1.000)	9.685 (2.671,112.999)	5007.056
Annealing MLE	0.997 (0.770,1.000)	9.339 (2.454,110.617)	5017.380
Factorization	1.000 (1.000,1.000)	2.717 (0.293,41.304)	197.175

Table A.4: Table of the average success rate of finding the global maxima of the log-likelihood for 8 test systems each with 7 training set sizes and 8 random seeds with different optimizers. The average time and iterations are also shown. The brackets identify the smallest and largest value observed. The derivatives of the targets (forces) are used.

A.6 Modification for Leave-one-out object function

The LOOCV prediction mean error scaled with the prediction uncertainty, z_{-i}, is defined as:

$$z_{-i} = \frac{y_{-i} - \overline{y}_{-i}}{\sigma_{-i}} \tag{A.4}$$

z_{-i} will be called the LOO scaled prediction error. The variance of the LOO scaled prediction errors, σ^2_{-z}, can be expressed as:

$$\begin{aligned}\sigma^2_{-z} =& \frac{1}{N}\sum_{i=1}^{N}(z_{-i}-\overline{z})^2 = \overline{z^2}-\overline{z}^2 \\ =& \left(\frac{1}{N}\sum_{i=1}^{N}\frac{(y_{-i}-\overline{y}_{-i})^2}{\sigma^2_{-i}}\right)-\left(\frac{1}{N}\sum_{i=1}^{N}\frac{y_{-i}-\overline{y}_{-i}}{\sigma_{-i}}\right)^2 \end{aligned} \tag{A.5}$$

σ^2_{-z} can also be factorized as $\sigma^2_{-z} = \alpha^{-2}\sigma^2_{-z0}$. Since the best prediction uncertainty is obtained if $\sigma^2_{-z} = 1$, the prefactor hyperparameter can be derived analytically as:

$$\alpha^2_{mod} = \sigma^2_{-z0} = \left(\frac{1}{N}\sum_{i=1}^{N}\frac{[\mathbf{C}_0^{-1}\left(\vec{y}(\mathbf{X})-\vec{\mu}(\mathbf{X})\right)]_i^2}{[\mathbf{C}_0^{-1}]_{ii}}\right)-\left(\frac{1}{N}\sum_{i=1}^{N}\frac{[\mathbf{C}_0^{-1}\left(\vec{y}(\mathbf{X})-\vec{\mu}(\mathbf{X})\right)]_i}{\sqrt{[\mathbf{C}_0^{-1}]_{ii}}}\right)^2 \tag{A.6}$$

A.7 Noise correction

In this section, the noise correction introduced in Section 3.3.1 is derived.

A matrix is invertible if the condition number is not infinity. The condition number of a matrix $\mathbf{A}$ can be expressed of the minimum and maximum eigenvalues. As a consequence of that, the matrix is singular if the minimum eigenvalue is zero. Numerically, the machine precision, $\varepsilon_M \approx 2.3 \cdot 10^{-16}$, is the limit before infinity. Therefore, the ratio between the maximum and minimum eigenvalues for a matrix must not be larger than the inverse machine precision if the matrix must be invertible as:

$$\mathrm{cond}(\mathbf{A}) = \frac{\lambda_{max}}{\lambda_{min}} \leq \frac{1}{\varepsilon_M} \tag{A.7}$$

The problem of inverting the covariance matrix arises within the regime of a large length-scale hyperparameter, where the covariance matrix becomes the all-ones matrix. Therefore, a small correction (the noise correction) can be added to the diagonal to make the covariance matrix invertible:

$$\mathbf{K}_c(\mathbf{X},\mathbf{X}) = \mathbf{K}_0(\mathbf{X},\mathbf{X}) + \delta_n\mathbf{I} \tag{A.8}$$

The largest possible eigenvalue of the covariance matrix is the trace of the covariance matrix, $\lambda_{max} \leq \mathrm{Tr}(\mathbf{K}_0)$. This is due to the eigenvalues being positive and the sum of the eigenvalues being equal to the trace of the matrix, $\mathrm{Tr}(\mathbf{K}_0) = \sum_{i=1}^{N_K}\lambda_i$. In the limit of the length-scale hyperparameter going towards infinity, the covariance matrix becomes the all-ones matrix, $\mathbf{J}$, as:

$$\lim_{l\to\infty}\mathbf{K}_c(\mathbf{X},\mathbf{X}) = \mathbf{J}_{NN}\alpha^2 + \delta_n\mathbf{I} \tag{A.9}$$

Then, the minimum eigenvalue will be the noise correction. Therefore, the conditional number is:

$$\mathrm{cond}(\mathbf{K}_c) \leq \frac{\sum_{i=1}^{N_K}[\mathbf{K}_0]_{ii} + \delta_n N_K}{\delta_n} \leq \frac{1}{\varepsilon_M} \tag{A.10}$$

where N_K is the number of diagonal elements of $\mathbf{K}_0$. Now, the noise correction can be analytically derived as:

$$\delta_n = \frac{\mathrm{Tr}(\mathbf{K}_0)}{c_\epsilon\epsilon_M^{-1} - N_K} \tag{A.11}$$

where $c_\epsilon \in (0, 1]$ is a constant to ensure that the condition number is lower than the machine precision.

Unfortunately, the eigendecomposition algorithm requires a larger noise correction than the analytical noise correction. A noise correction that also works for the eigendecomposition is:

$$\delta_n = \frac{\mathrm{Tr}(\mathbf{K})^2}{c_\epsilon \epsilon_M^{-1} - N_K^2} \tag{A.12}$$

The noise correction is tested on all-ones matrices with different sizes on a local MacBook and Simple Linux Utility for Resource Management (SLURM) cluster with nodes of Xeon 16 and Xeon 40 (see Fig. A.1).

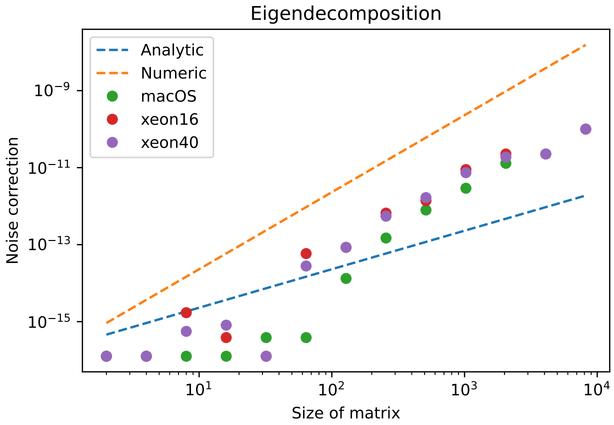


Figure A.1: Analytic noise corrections from Eq. A.11 (blue dashed line) and Eq. A.12 (orange dashed line) plotted as a function of the size of all-ones matrices. The scatter plots are the minimum noise correction needed for inverting the all-ones matrices on a MacBook (green points), a Xeon 16 node (red points), and a Xeon 40 node (purple points) as a function of matrix size.

The derivative of the corrected covariance matrix wrt. the length-scale hyperparameter is:

$$\frac{\mathrm{d}\mathbf{K}_c(\mathbf{X},\mathbf{X})}{\mathrm{d}l} = \frac{\mathrm{d}\mathbf{K}_0(\mathbf{X},\mathbf{X})}{\mathrm{d}l} + \mathbf{I}\frac{2\mathrm{Tr}(\mathbf{K})}{c_\epsilon \epsilon_M^{-1} - N_K^2}\sum_{i=1}^{N_K}\frac{\mathrm{d}[\mathbf{K}_0(\mathbf{X},\mathbf{X})]_{ii}}{\mathrm{d}l} \tag{A.13}$$

A.8 Objective function error predictions

The results discussed in Section 3.2.4 are displayed here.

The geometric mean of the prediction mean and uncertainty errors when only energies are used are shown in Table A.5.

The geometric mean of the prediction mean and uncertainty errors when energies and forces are used are shown in Table A.6.

Method	RMSE / [eV]	UD	NLPP
GP LOO	7.76e-02 (1.11e-06,1.61e+02)	2.38e-01 (1.56e-06,1.93e+02)	4.46e+05
GP GPP	8.10e-02 (1.70e-06,9.47e+01)	3.48e-01 (1.22e-05,7.43e+01)	7.43e+03
GP GPE	7.76e-02 (1.12e-06,1.61e+02)	1.92e+03 (6.35e+01,2.96e+04)	5.10e+74
GP LL	7.29e-02 (8.07e-07,8.50e+00)	2.41e-01 (3.38e-07,1.10e+02)	2.06e+04
GP LP	7.29e-02 (8.10e-07,8.77e+00)	2.40e-01 (1.48e-06,8.46e+01)	7.18e+03
GP LL mod.	7.29e-02 (8.07e-07,8.50e+00)	2.13e-01 (1.15e-06,1.10e+02)	2.05e+04
TP LL	7.29e-02 (8.07e-07,8.50e+00)	2.15e-01 (8.07e-07,8.85e+01)	6.43e+03
TP LP	7.29e-02 (8.10e-07,8.77e+00)	2.17e-01 (2.03e-05,6.56e+01)	1.91e+03

Table A.5: Table of the geometric mean prediction errors of test systems. The training targets are the energies. The LOO is the leave-one-out object function with modification. LL mod. denotes log-likelihood with modification. The error bars show the smallest and largest value observed. The brackets identify the smallest and largest value observed.

Method	RMSE / [eV]	UD	NLPP
GP LOO	1.61e-01 (1.60e-04,9.70e+01)	5.88e-01 (1.00e-05,1.00e+02)	2.39e+04
GP GPP	1.54e-01 (1.43e-04,3.29e+01)	5.68e-01 (8.86e-06,1.29e+02)	8.49e+04
GP GPE	1.61e-01 (1.60e-04,9.98e+01)	2.36e+03 (1.27e+02,3.00e+04)	2.80e+75
GP LL	1.31e-01 (1.50e-04,7.88e+00)	4.18e-01 (4.46e-04,4.04e+01)	3.76e+02
GP LP	1.30e-01 (1.50e-04,7.66e+00)	4.04e-01 (3.48e-05,3.23e+01)	1.31e+02
GP LL mod.	1.31e-01 (1.50e-04,7.88e+00)	3.54e-01 (1.32e-05,3.84e+01)	2.11e+02
TP LL	1.31e-01 (1.50e-04,7.88e+00)	3.81e-01 (2.60e-06,3.91e+01)	2.65e+02
TP LP	1.30e-01 (1.50e-04,7.66e+00)	3.85e-01 (8.69e-04,3.12e+01)	3.46e+01

Table A.6: Table of the geometric mean prediction errors of test systems. The training targets are the energies and derivatives. The LOO is the leave-one-out object function with modification. LL mod. denotes log-likelihood with modification. The error bars show the smallest and largest value observed. The brackets identify the smallest and largest value observed.

A.9 Machine learning accelerated Global Optimization

The results obtained and discussed in Chapter 4 is shown in Table A.7.

System	Evaluations	Stand. evaluations	Energy deviation / [eV]
2H_Ag_GPAW	64	426	1.018
2H_Pt_GPAW	66	584	0.091
CO_Cu_GPAW	27	78	-0.001
H_Ag_GPAW	10	16	0.000
H_Pt_GPAW	15	28	0.001
OH_RuO2_GPAW	45	387	0.008
O_Pd211_GPAW	27	593	-0.000
O_Pd_GPAW	15	40	-0.000
O_Pd_fix	12	40	-0.001
O_RuO2_fix_GPAW	23	1109	-0.001
O_RuO2_GPAW	29	1009	-0.001

Table A.7: Table of the required evaluations with the new Machine learning accelerated Global Optimization method compared to the standard methods. The energy deviation (new method's energy - standard methods' energy) is also shown.

B Included publications

B.1 Paper I

Best Conventional Gaussian Process

Andreas Lynge Vishart and Thomas Bligaard

To be submitted

Best Conventional Gaussian Process

Andreas Lynge Vishart[1] and Thomas Bligaard[1]

[1]*ASM, Department of Energy Conversion and Storage,*
Technical University of Denmark, Kongens Lyngby, Denmark

Maximum likelihood estimation is a problematic and often unsuccessful task, even though it is the most used approach for tuning a Gaussian process. The posterior distribution of the hyperparameters from a Gaussian process is a multi-modal distribution with large flat regions. The success of the tuning is crucially dependent on the defined area of interest for the hyperparameters. Applying prior distributions to the hyperparameters are beneficial for tuning if the system is well-known. Besides the likelihood, a variety of loss functions are considered. An introduction of the new factorization method can guarantee a global maximization of the likelihood and significantly reduce the computational cost.

INTRODUCTION

Electronic structure calculations are a growing field for explaining and discovering the regime of molecules and condensed phases. However, insightful electronic structure calculations come with a computationally expensive cost. Even though Density Functional Theory[1, 2] (DFT) is considered a great compromise between accuracy and computational cost, the cost is still overwhelming due to the high number of iterations required for the structure search algorithms.

Machine learning (ML) methods have shown to be promising for accelerating the expensive techniques that require electronic structure calculations[3–9]. Especially, molecular dynamics, local structure optimizations, global structure optimizations, and transition state searches with minimum energy paths have been accelerated tremendously by applying ML models. However, a precise and reliable ML model is the foundation of all those accelerated techniques.

The complete potential energy surface calculated with DFT for a specific atomistic system can also be learned by a ML model[10–12]. A high accuracy is then obtained with a low computational cost. However, it requires a large predefined database with structures of the specific atomistic system and their DFT energies.

On the contrary, a structure search can be accelerated by learning the region of interest and interpolating between the calculated points. Active learning gives the benefit of creating a database on the fly, which only contains the expensive computational calculations needed. However, it requires that the ML model can predict uncertainties.

The Gaussian process (GP) [13] is a well-known ML model for accelerating the aforementioned expensive techniques. The GP is especially suited for active learning since it predicts the best fit and corresponding uncertainties. The GP interpolates well even with a few training points. The drawback of the GP is that it scales $\mathcal{O}(N^3)$ with the number of training points, N, and its hyperparameters must also be optimized. The optimization of the hyperparameters is not an easy task even for three hyperparameters, which are often used. The normal procedure for optimizing the hyperparameters of the GP is by using log-likelihood, LL, or log-posterior, LP, maximization. The maximum likelihood estimation (MLE) and the maximum a posteriori estimation (MAP) assume that the posterior distribution of the hyperparameters is practically described by its mode with a single set of hyperparameters, which is a crude approximation and it is only valid when enough training data is applied. Especially the uncertainty prediction is affected by the crude single-point estimation. Often, a simple local optimization is used for maximizing the hyperparameters. A brute-force grid search can also be used although it is computationally costly. Hamiltonian Monte Carlo[14] can also be used to get a complete representation of the hyperparameter's posterior distribution. However, it is extremely computationally expensive, and it has to be recalculated for each new test point if the fully Bayesian predictive distribution has to be used.

The MLE can easily fail especially when an educated guess is not applied. However, the educated guess will change with the number of training points and the system considered. In this work, the typical mistakes when optimizing the hyperparameters are illustrated and explained. Furthermore, a new method that guarantees to find the global maximum for the three most used hyperparameters is introduced. The performance of the new optimization method is shown and compared to usual optimization methods on nine test systems with various training set sizes and random seeds. Different objective functions are tested to verify if LL is the best objective function. Furthermore, the underestimated uncertainty prediction of the GP due to MLE is addressed by introducing a modification.

THEORY

A GP is a conditional multivariate normal distribution. Therefore, a predicted target value, y_*, with a feature $\vec{x}_*$ is given by Bayesian inference from a set of training features, $\mathbf{X}$, and targets, $\vec{y}$, expressed as:

$$p(y_* \mid \vec{x}_*, \vec{y}, \mathbf{X}, \vec{\theta}) = \mathcal{N}\left(y_* \mid \bar{y}_*, \sigma_*^2\right) \quad (1)$$

where $\bar{y}_*$ is the predictive mean, σ_*^2 is the predictive variance, and $\vec{\theta}$ is a set of hyperparameters for the GP. The function dependencies of the features have been disregarded due to simplicity. The predictive mean and variance are analytically determined as:

$$\bar{y}_* = \mu_* + \mathbf{K}_* \mathbf{C}^{-1} \left(\vec{y} - \vec{\mu}\right) \tag{2}$$

$$\sigma_*^2(\vec{x}_*) = \alpha^2 (k_{**} - \mathbf{K}_* \mathbf{C}^{-1} \mathbf{K}_*^\top + \sigma_r^2) \tag{3}$$

where $\mu_*(\vec{x}_*)$ is the prior mean of the predicted point, $\vec{\mu}(\mathbf{X})$ is the prior mean of the training points, $\mathbf{K}_* \equiv \mathbf{K}(\vec{x}_*, \mathbf{X})$ is the covariance matrix between the predicted point and the training points, $k_{**} \equiv k(\vec{x}_*, \vec{x}_*)$ is the covariance matrix element of the test point with itself, and α is the prefactor hyperparameter. Furthermore, $\mathbf{C}$ is the covariance matrix of the training points with noise expressed as:

$$\mathbf{C} = \mathbf{K}(\mathbf{X}, \mathbf{X}) + \delta_n \mathbf{I} + \sigma_r^2 \mathbf{I} \tag{4}$$

where σ_r is the relative-noise hyperparameter and δ_n is the noise correction. The introduced noise correction is expressed as:

$$\delta_n = \frac{\text{Tr}(\mathbf{K})^2}{(\epsilon_M)^{-1} - N^2} \tag{5}$$

The noise correction is the minimum noise required to ensure that the covariance matrix is invertible. The noise correction is essential since the covariance matrix can be singular. The mean of the training targets is used as the prior mean.

The elements of the covariance matrices are kernel function values. It is the kernel function that makes it possible to connect the features to the targets. The kernel function chosen in this work is the well-known squared exponential kernel function:

$$k(\vec{x}_p, \vec{x}_q) = \exp\left(\frac{-|\vec{x}_p - \vec{x}_q|^2}{2l^2}\right) \tag{6}$$

where l is the length-scale hyperparameter. Usually, the prefactor hyperparameter is included in the kernel function. In this work, the prefactor is factorized outside of the kernel function, and a relative-noise hyperparameter is defined as a free hyperparameter instead of the common noise hyperparameter. This factorization gives a better understanding of the hyperparameters effect. The prefactor hyperparameter determines the variance of the targets and affects the prediction uncertainty. However, the prefactor hyperparameter has no effect on the prediction mean. The relative-noise hyperparameter controls the regularization of the predictions. The inversion of the covariance matrix is also stabilized by the factorization due to machine precision.

The distribution of the hyperparameters can be expressed from Bayes' theorem:

$$p(\vec{\theta} \mid \vec{y}, \mathbf{X}) = \frac{p(\vec{y} \mid \vec{\theta}, \mathbf{X}) p(\vec{\theta})}{p(\vec{y} \mid \mathbf{X})} \tag{7}$$

However, often the prior distribution of the hyperparameters, $p(\vec{\theta})$, is chosen to be a uniform prior distribution. As a consequence of that, the posterior distribution of the hyperparameters is the likelihood. The log-likelihood, LL, expression with the factorized prefactor hyperparameter is:

$$\begin{aligned} LL \equiv & \frac{-1}{2\alpha^2} (\vec{y} - \vec{\mu})^\top \mathbf{C}^{-1} (\vec{y} - \vec{\mu}) - \frac{1}{2} \ln\left(|\mathbf{C}|\right) \\ & - \frac{N}{2} \ln\left(\alpha^2\right) - \frac{N}{2} \ln\left(2\pi\right) \end{aligned} \tag{8}$$

Thereby, an analytic solution of the prefactor hyperparameter, α_{MLE}, can be derived from maximizing the LL[15]:

$$\alpha_{\text{MLE}}^2 = \frac{1}{N} (\vec{y} - \vec{\mu})^\top \mathbf{C}^{-1} (\vec{y} - \vec{\mu}) \tag{9}$$

One eigendecomposition of the covariance matrix without the relative-noise hyperparameters, $\mathbf{K} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top$, is enough to search after all values of the relative-noise hyperparameter and to find the prefactor solution for each length-scale hyperparameter. Then, the factorized log-likelihood is given as:

$$\begin{aligned} LL = & \frac{-N}{2} \left(1 + \ln\left(2\pi\right)\right) - \frac{1}{2} \sum_{i=1}^{N} \ln\left([\mathbf{\Lambda}]_{ii} + \sigma_r^2\right) \\ & - \frac{N}{2} \ln\left(\frac{1}{N} \sum_{i=1}^{N} \frac{[\mathbf{U}^\top (\vec{y} - \vec{\mu})]_i^2}{[\mathbf{\Lambda}]_{ii} + \sigma_r^2}\right) \end{aligned} \tag{10}$$

Thereby, the LL is independent of the prefactor hyperparameter in the optimization.

In this work, a modified version of LL is also applied. The modification is performed by changing the analytical prefactor hyperparameter solution in Eq. 9 into an unbiased estimation of the variance after the maximization of LL. The modified solution to the prefactor hyperparameter is expressed as:

$$\alpha_{\text{mod}}^2 = \frac{N}{N - D_\theta} \alpha_{\text{MLE}}^2 \tag{11}$$

where D_θ is the number of hyperparameters optimized.

In this work, the objective functions from Ref. [16] are also tested. The analytical expression for leave-one-out cross-validation (LOO) is obtainable with a GP, which is expressed as:

$$LOO = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{[\mathbf{C}^{-1} \left(\vec{y} - \vec{\mu}\right)]_i}{[\mathbf{C}^{-1}]_{ii}}\right)^2 \tag{12}$$

The posterior predictive distribution can also be expressed in terms of LOO as:

$$\begin{aligned} GPP = & \frac{1}{N\alpha^2} \sum_{i=1}^{N} \frac{[\mathbf{C}^{-1} \left(\vec{y} - \vec{\mu}\right)]_i^2}{[\mathbf{C}^{-1}]_{ii}} + \ln\left(\alpha^2\right) \\ & - \frac{1}{N} \sum_{i=1}^{N} \ln\left([\mathbf{C}^{-1}]_{ii}\right) + \ln\left(2\pi\right) \end{aligned} \tag{13}$$

The LOO expression in Eq. 12 is independent of the prefactor hyperparameter and does not optimize the prediction uncertainty. Therefore, a modification has been implemented that determines the prefactor hyperparameter without changing the prediction mean. The prefactor hyperparameter for the modified LOO is expressed as:

$$\alpha_{mod}^2 = \left(\frac{1}{N} \sum_{i=1}^{N} \frac{\left[\mathbf{C}^{-1}\left(\vec{y}-\vec{\mu}\right)\right]_i^2}{\left[\mathbf{C}^{-1}\right]_{ii}} \right) - \left(\frac{1}{N} \sum_{i=1}^{N} \frac{\left[\mathbf{C}^{-1}\left(\vec{y}-\vec{\mu}\right)\right]_i}{\sqrt{\left[\mathbf{C}^{-1}\right]_{ii}}} \right)^2 \tag{14}$$

METHODS

The test systems

Seven different training set sizes are used for nine different test systems to discuss the performance of the global optimization algorithms. The test system consists of a one-dimensional analytic test function and seven atomistic systems that are treated in the Atomic Simulation Environment[17, 18] python package (ASE), where the potential energies are calculated with either Effective Medium Theory[19, 20] (EMT) or with DFT exchange-correlation functional PBE[21] in GPAW[22, 23]. Eight random seeds are used for each test system, and 100 initial random sets of hyperparameters are sampled for each of the random seeds. For detailed information about the test systems, see SI.

The hyperparameters

A hyperparameter optimization is defined as successful if the objective function value has a relative and absolute error smaller than $1.0 \cdot 10^{-3}$ compared to the global optimum. The global maxima are defined as the greatest object function values observed after all optimization for each of the test systems at each random seed and training set size. The success rate is calculated as the fraction of successful hyperparameter optimizations. Mean values together with minimum and maximum success rates over different test systems and random seeds are used to compare success rates.

Boundary conditions are defined for the hyperparameters to restrict the search space (see Table. I). The initial sets of hyperparameters are sampled from a uniform distribution of the boundary conditions. A process with a shorter length-scale than the lower boundary condition will not describe the interpolation between the points and will be an overfit to the training points. However, a larger length-scale than the median distance will not use the data, but takes an average of the training points and will be an underfit. A lower relative-noise hyperparameter than the lower boundary condition does not have any effect, because of the machine precision and the noise correction. A higher relative-noise hyperparameter than the upper boundary condition will lead to an underfit since the largest possible eigenvalue of the covariance matrix without the prefactor will be N when derivatives of the targets are not used.

Hyperparameter	Min. bound	Max. bound
Length-scale (l)	$\frac{\text{median}(\overrightarrow{\text{NN}})}{5s}$	$4s \cdot \text{median}(\mathbf{D})$
Prefactor (α)	$\frac{1}{10s}\sqrt{\frac{1}{N}\lvert\vec{y}-\vec{\mu}\rvert^2}$	$10s\sqrt{\frac{1}{N}\lvert\vec{y}-\vec{\mu}\rvert^2}$
Relative-noise (σ_r)	$10\sqrt{2\varepsilon_M}$	N

TABLE I: Table of the boundary conditions obtained by the educated guesses of the hyperparameters when using the squared exponential kernel family. s is the scaling factor chosen, $\overrightarrow{\text{NN}}$ is the nearest neighbor distance for each training data in the feature space, $\mathbf{D}$ is the distance matrix in the feature space, and ϵ_M is the machine precision.

A variable transformation of the hyperparameters is introduced to enlarge the region of interest of the hyperparameters and without restricting any values of the hyperparameters. The scaled-logit functions are used for the variable transformation from the new parameters $t_\theta \in (0,1)$ to the old hyperparameters:

$$\ln(\theta) = \mu_\theta + s_\theta \ln\left(\frac{t_\theta}{1-t_\theta}\right) \tag{15}$$

The mean values of the logistic distributions, μ_θ, is the average of the logarithm of the minimum, $b_{\theta,\text{min}}$, and maximum boundary conditions, $b_{\theta,\text{max}}$, (see Table I) of the hyperparameters:

$$\mu_\theta = \frac{1}{2}\left(\ln\left(b_{\theta,\text{min}}\right) + \ln\left(b_{\theta,\text{max}}\right)\right) \tag{16}$$

The scaling of the logistic distributions is set to 0.14 times the difference of the logarithm of the boundary conditions of the hyperparameter:

$$s_\theta = 0.14\left(\ln\left(b_{\theta,\text{max}}\right) - \ln\left(b_{\theta,\text{min}}\right)\right) \tag{17}$$

The value of 0.14 is selected since the boundary conditions of the hyperparameters then corresponds to a 95% percentile of the logistic distribution.

The optimization methods

The different optimization methods used are local, basin-hopping, grid-search, simulated annealing, random sampling, and the new factorized line-search (factorization method) optimization.

The local optimization method uses Scipy's minimizer[24] with the L-BFGS-B method[25] for maximizing the LL (Eq. 8).

The basin-hopping[24, 26] implementation from Scipy is used to optimize Eq. 8 with 15 basin-hopping iterations and initialized from the initial sets of the hyperparameters.

The grid-search method defines a grid with 12 points in each dimension for $\vec{t_\theta}$ and calculates the LL in all points. The point with the greatest value of LL is then maximized with the local optimization method.

The dual-simulated annealing[24, 27–30] method implemented in Scipy with default parameters and 5000 maximum iterations is used as the simulated annealing method. The simulated annealing method uses the $\vec{t_\theta}$ space and optimizes LL with the analytic solution of the prefactor.

The random sampling method, as the name implies, sample 19 different sets of hyperparameters in $\vec{t_\theta}$ space and uses the initial set of hyperparameter given. Then, all the hyperparameter sets are locally optimized.

The factorization method initially makes a grid of 80 points in the t_l space. For each length-scale hyperparameter, a grid of 50 points is constructed in the space of the variable-transformed relative-noise hyperparameter. All LL values of the grid in the relative-noise hyperparameter space are calculated with Eq. 10, which only requires a single eigendecomposition of the covariance matrix with the given length-scale hyperparameter. A golden-section search[31] is performed on the intervals surrounding the maximum of LL values. This process is performed for all length-scale hyperparameters in the grid. All intervals that surround a maximum of LL are identified with the finite difference method and optimized with a golden-section search.

The prediction evaluation

The predictive abilities of the GP with the hyperparameters optimized by global optimization of the different objective functions are evaluated and compared. When the LP is maximized, normal prior distributions are used for the length-scale and relative-noise hyperparameters in the logarithmic space (log-space). The mean of the length-scale prior is 2.0, and the standard deviation is 3.0 since the Cartesian coordinates are used as the fingerprint and potential energy changes around that length-scale of the Cartesian coordinates. The mean of the prior distribution of the relative-noise is -9.0, and the standard deviation is 3.0 since the noise on the potential energy from an EMT or DFT calculation is small.

The evaluation of the prediction quality is based on the root-mean-square error (RMSE) for the prediction mean and new defined uncertainty measure, uncertainty deviation (UD). The UD is the error between the variance from the standardized predicted distribution and the standard normal distribution. The RMSE is:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_{*i} - \overline{y}_{*i})^2} \tag{18}$$

where $\overline{y}_{*i}$ is the prediction of test point i and M is the number of test points. The UD error is expressed as:

$$UD = \ln\left(\left(\left(\frac{1}{M}\sum_{i=1}^{M} z_i^2\right) - \left(\frac{1}{M}\sum_{i=1}^{M} z_i\right)^2\right)^2\right) \tag{19}$$

where $z_i = \frac{y_{*i}-\overline{y}_{*i}}{\sigma_{*i}}$ is the standardized prediction error for test point i. The geometric mean is used to summarise the prediction qualities.

RESULTS & DISCUSSION

Challenges of optimizing hyperparameters

Multiple problems can occur already with three hyperparameters. One major issue is the large flat region at low length-scale and relative-noise values (see Fig. 1) which corresponds to overfitting and where the covariance matrix is going towards an identity matrix. Consequently, the optimization of hyperparameters initialized in the mentioned region will converge immediately. Therefore, a global optimization method must be used. A similar problem is another large flat region at high relative-noise values (larger than the number of training points) that corresponds to all observations being treated as noise. Another critical problem is that the covariance matrix can be singular at large length-scale and low relative-noise hyperparameters if the noise correction is not applied. This problem comes from the covariance matrix going towards an all-ones matrix.

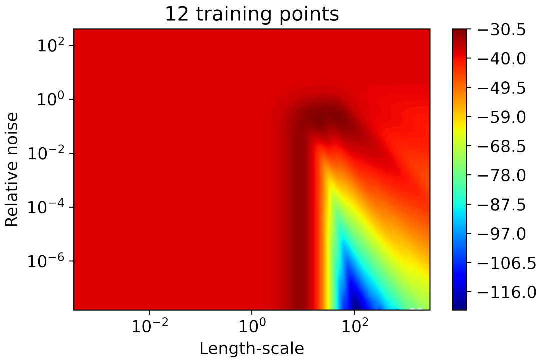


FIG. 1: Log-likelihood with maximized prefactor hyperparameter as a function of the length-scale and relative-noise hyperparameters for the one-dimensional test system.

Furthermore, the LL can be a multimodal distribution and give multiple reasonable and different processes.

It is important to consider the hyperparameters in the log-space since the hyperparameters must be scale-invariant given that the features can have any length-scale and the targets can have any function values. Furthermore, the success rate comparison between local optimization of hyperparameters in the linear- and log-spaces clearly shows an advantage of using hyperparameters in the log-space (see Fig. 2).

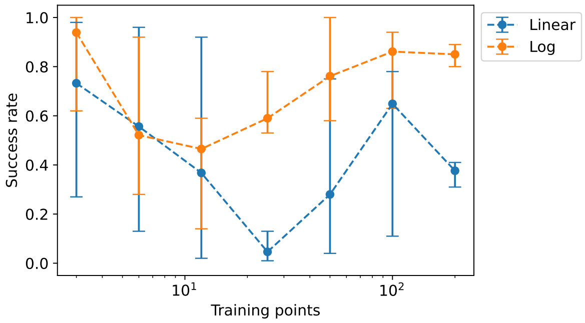


FIG. 2: A comparison in success rate between hyperparameters in the linear- (orange curve) and logarithmic-space (blue curve). 100 initial sets of hyperparameters for every eight random seeds at each training set size are locally optimized with L-BFGS-B from Scipy[24].

The boundary conditions of the hyperparameters are crucial for a good optimization of the hyperparameters due to the large flat regions on the LL surface. A success rate comparison can also illustrate the importance of the boundary conditions (see Fig. 3). The success rate of finding the global maximum of the LL consistently decreases as a function of the increase of the boundary conditions of the length-scale and prefactor hyperparameters.

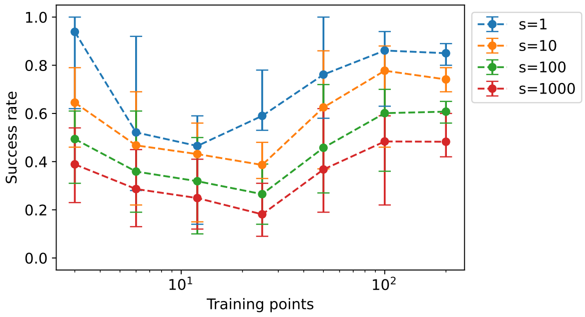


FIG. 3: A comparison in success rate between hyperparameters sampled in the boundary condition interval (blue curve), 10 times (orange curve), 100 times (green curve), and 1000 times (red curve) the boundary conditions of the length-scale and prefactor hyperparameter. 100 initial sets of hyperparameters for every eight random seeds at each training set size are locally optimized with L-BFGS-B from Scipy[24].

The optimization methods

The local optimizer finds the global maximum 86.3 % of the time on average. However, the success rates change drastically depending on the training set and the initial set of hyperparameters.

The factorization method outperforms the other optimization methods (see Fig. 4) in terms of success rate for finding the global maximum of LL for the nine test systems with seven different training set sizes and eight different random seeds each. The factorization method locates the global maximum in all test cases and is therefore a robust method for finding the global maximum of LL. However, it is necessary to state that the basin of attraction in the length-scale hyperparameter has to be larger than the grid spacing. The computational cost of the factorization method is larger than the local optimization due to fewer iterations in the local optimization and the larger computational cost of the eigendecomposition compared to the Cholesky decomposition. However, the local optimization is far from sufficient for finding the global maximum. The factorization method does not depend on probability for finding a reasonable initial hyperparameter set and will give a consistent hyperparameter solution. Furthermore, it is not as computationally expensive as the robust grid-search method, and therefore a finer grid can be achieved.

The grid search method cannot find all global maximums since the grid is not dense enough due to its high computational cost.

The random sampling method is performing well in terms of success rate, but it relies on probability to find the global maximum. Therefore, the global maximum is not guaranteed. The computational cost of the random sampling method with the chosen number of samplings is beyond the factorization method.

The simulated annealing method also has problems locating the global maximums consistently, and it is computationally expensive.

The prediction evaluation

The modification of the LOO significantly improves its uncertainty predictions (see Table II). The uncertainty predictions of the modified LOO are slightly better than the uncertainty predictions from LL and LP. However, the prediction means from LOO are slightly worse than the prediction means from LL and LP. LOO can result in overfitting due to the prediction of only a single point in the cross-validation. GPP does not perform as well as the other methods in terms of prediction mean and uncertainty. The use

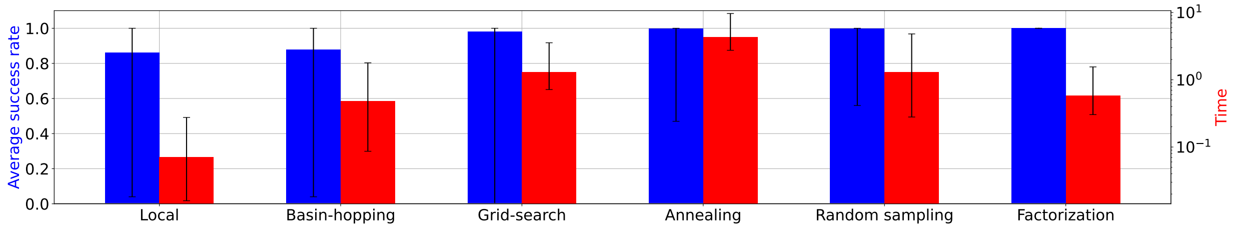


FIG. 4: The average success rate for finding the global maximum of the likelihood for each optimization method. The success rate from 100 initial sets of hyperparameters for each of the eight random seeds at each training set size at each test system is averaged and shown together with error bars from the minimum and maximum success rate for each global optimization method.

Method	RMSE / [eV]	UD
LOO	7.76e-02 (1.11e-06,1.61e+02)	1.43e+01 (1.24e-06,2.85e+03)
LOO mod.	7.76e-02 (1.11e-06,1.61e+02)	2.38e-01 (1.56e-06,1.93e+02)
GPP	8.10e-02 (1.70e-06,9.47e+01)	3.48e-01 (1.22e-05,7.43e+01)
LL	7.29e-02 (8.07e-07,8.50e+00)	2.41e-01 (3.38e-07,1.10e+02)
LP	7.29e-02 (8.10e-07,8.77e+00)	2.40e-01 (1.48e-06,8.46e+01)
LL mod.	7.29e-02 (8.07e-07,8.50e+00)	2.13e-01 (1.15e-06,1.10e+02)

TABLE II: Table of the geometric mean prediction errors of test systems. The training targets are the energies. The LOO is the leave-one-out object function with modification. LL mod. Denotes log-likelihood with modification. The error bars show the smallest and largest value observed. The brackets identify the smallest and largest value observed.

of prior distributions on the relative-noise hyperparameters ensures that the model takes the low noise of the potential energies from the EMT or DFT calculations into account to avoid underfitting when the training sets are small. Similarly, the length-scale prior distributions enforce a small enough length-scale to avoid underfitting, but it also avoids overfitting when the training sets are small. Therefore, the greatest error in the prediction uncertainty is smaller for LP than for LL. The modification to LL improves the prediction uncertainty. Therefore, the LL with modification is the Pareto-optimal solution of the tested objective functions.

CONCLUSION

The LL is verified to be the best of the investigated objective functions since it leads to a good compromise between prediction means and uncertainties. Using prior distributions makes the prediction uncertainty more controlled if prior knowledge is known. A simple modification to the prefactor hyperparameter improves the uncertainty prediction without changing the prediction mean. Therefore, robustly maximizing the LL or LP is essential.

The large flat regions of LL make it challenging to maximize LL with local and global methods. The flat regions lead to overfitted and underfitted models. Thus, enlarging of the important regions of the LL surface without restricting possible hyperparameters is essential. It is possible with a variable transformation that uses defined boundary conditions for the hyperparameters. Then, a grid can be constructed in the entire hyperparameter space. This complete grid permits the factorization method. The new factorization method consistently obtains the maxima of LL for all optimizations of the hyperparameters. Furthermore, the factorization method has a lower computational cost than the other global optimizers. Therefore, a robust and reliable GP can be obtained.

[1] P. Hohenberg and W. Kohn, Physical Review **136**, B864 (1964).

[2] W. Kohn and L. J. Sham, Physical Review **140**, A1133 (1965).

[3] A. Khorshidi and A. A. Peterson, Computer Physics Communications **207**, 310 (2016).

[4] O.-P. Koistinen, F. B. Dagbjartsdóttir, V. Ásgeirsson, A. Vehtari, and H. Jónsson, The Journal of Chemical Physics **147**, 152720 (2017).

[5] J. A. G. Torres, P. C. Jennings, M. H. Hansen, J. R. Boes, and T. Bligaard, Physical Review Letters **122**, 156001 (2019).

[6] C. Panosetti, A. Engelmann, L. Nemec, K. Reuter, and J. T. Margraf, Journal of Chemical Theory and Computation **16**, 2181 (2020).

[7] E. G. del Río, J. J. Mortensen, and K. W. Jacobsen, Physical Review B **100**, 104103 (2019).

[8] M. K. Bisbo and B. Hammer, Physical Review Letters **124**, 086102 (2020).

[9] S. Kaappa, C. Larsen, and K. W. Jacobsen, Physical Review Letters **127**, 166001 (2021).

[10] S. Lorenz, A. Groß, and M. Scheffler, Chemical Physics Letters **395**, 210 (2004).

[11] J. Behler, S. Lorenz, and K. Reuter, The Journal of Chemical Physics **127**, 014705 (2007).

[12] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky,

Nature Communications **13**, 2453 (2022).

[13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, 2005).

[14] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, Physics Letters B **195**, 216 (1987).

[15] E. G. del Río, S. Kaappa, J. A. G. Torres, T. Bligaard, and K. W. Jacobsen, The Journal of Chemical Physics **153**, 234116 (2020).

[16] S. Sundararajan and S. S. Keerthi, Neural Computation **13**, 1103 (2001).

[17] S. Bahn and K. Jacobsen, Computing in Science Engineering **4**, 56 (2002).

[18] 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, Journal of Physics: Condensed Matter **29**, 273002 (2017).

[19] J. K. Nørskov and N. D. Lang, Physical Review B **21**, 2131 (1980).

[20] Z. Xi, B. Chakraborty, K. W. Jacobsen, and J. K. Norskov, Journal of Physics: Condensed Matter **4**, 7191 (1992).

[21] J. P. Perdew, K. Burke, and Y. Wang, Physical Review B **54**, 16533 (1996).

[22] J. J. Mortensen, L. B. Hansen, and K. W. Jacobsen, Physical Review B **71**, 035109 (2005).

[23] 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, Journal of Physics: Condensed Matter **22**, 253202 (2010).

[24] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İlhan Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, A. Vijaykumar, A. P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C. N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D. A. Nicholson, D. R. Hagen, D. V. Pasechnik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G. A. Price, G.-L. Ingold, G. E. Allen, G. R. Lee, H. Audren, I. Probst, J. P. Dietrich, J. Silterra, J. T. Webber, J. Slavič, J. Nothman, J. Buchner, J. Kulick, J. L. Schönberger, J. V. de Miranda Cardoso, J. Reimer, J. Harrington, J. L. C. Rodríguez, J. Nunez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kümmerer, M. Bolingbroke, M. Tartre, M. Pak, N. J. Smith, N. Nowaczyk, N. Shebanov, O. Pavlyk, P. A. Brodtkorb, P. Lee, R. T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlik, T. Oshima, T. J. Pingel, T. P. Robitaille, T. Spura, T. R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y. O. Halchenko, and Y. Vázquez-Baeza, Nature Methods **17**, 261 (2020).

[25] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, SIAM Journal on Scientific Computing **16**, 1190 (1995).

[26] D. J. Wales and J. P. K. Doye, The Journal of Physical Chemistry A **101**, 5111 (1997).

[27] C. Tsallis, Journal of Statistical Physics **52**, 479 (1988).

[28] C. Tsallis and D. A. Stariolo, Physica A: Statistical Mechanics and its Applications **233**, 395 (1996).

[29] Y. Xiang, D. Sun, W. Fan, and X. Gong, Physics Letters A **233**, 216 (1997).

[30] Y. Xiang and X. G. Gong, Physical Review E **62**, 4473 (2000).

[31] J. Kiefer, Proceedings of the American Mathematical Society **4**, 502 (1953).

[32] K. Müller and L. D. Brown, Theoretica Chimica Acta **53**, 75 (1979).

[33] W. Kohn and L. J. Sham, Physical Review **140**, A1133 (1965).

[34] B. Hammer, L. B. Hansen, and J. K. Nørskov, Physical Review B **59**, 7413 (1999).

[35] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, The Journal of Chemical Physics **81**, 3684 (1984).

Supporting Information

Test systems

Nine test systems are used to study the optimization of the hyperparameters. A database is made for each of the nine test systems. Training set sizes of 3, 6, 12, 25, 50, 100, and 200 are sampled from each of the databases with 8 different random seeds. The random seeds are from 1 to 8. The training sets with larger sizes include the same points as training sets with fewer points at each seed. A test set of 400 points is also sampled from the database. The points of the test set are excluded from the training sets. All atomistic structures are considered in the atomic simulation environment[17, 18] (ASE) python package.

Simple test system

The first test system is a simple analytical function with the expression:

$$g(x) = 3\sin\left(\frac{x^2}{20^2}\right) - 9\sin\left(\frac{0.6x}{20}\right) + 17 \qquad (20)$$

The database with the simple test system uses an 1 dimensional input feature x from -40 to 100 with 800 points. The simple test system is used to illustrate the challenges of finding the global maxima of the LL.

Müller-Brown

The Müller-Brown potential energy[32] is used for a 2-dimensional analytical test system. The database is calculated from a feature grid in the x dimension from -1.4 to 0.2 with 30 points and in the y dimension from 0.0 to 1.9 with 30 points.

Au at Al

Another test system is a gold atom at a fixed aluminum(100) surface with $3 \times 3 \times 4$ atoms. The potential energy of the gold atom is calculated from a 3-dimensional grid. The grid points in the x dimension are from 0.0 to 5.0 with 12 points. The grid points in the y dimension are from 0.0 to 3.2 with 12 points. The last dimension has grid points from 0.0 to 2.0 with 6 points. The potential energy is calculated with the effective medium theory[19, 20] (EMT).

CO at Ni

Two databases are constructed from molecular dynamic simulations (MDs) calculation of carbon monoxide at a fixed nickel(100) surface. The first database uses EMT for calculating the potential energy. The nickel surface has $3 \times 3 \times 5$ atoms. The second database uses density functional theory[33**?**] (DFT) with RPBE[34] as the exchange-correlation functional. GPAW[22, 23] is used for the electronic structure calculation. The default parameters are used in GPAW. The nickel surface has $3 \times 3 \times 3$ atoms. The canonical ensemble (NVT) is used for both MDs. The Berendsen thermostat[35] is used to scale the temperature at every step. A time step of 0.5 fs, a temperature of 800 K, and 800 steps are used in both MDs. The initial structure is constructed from the carbon atom and oxygen atoms are adsorbed in a hollow side each.

Copper clusters

Two databases are made from copper clusters. The smallest cluster consist of 5 copper atoms (Cu5). 13 copper atoms are used in the largest cluster (Cu13). The potential energies of both databases are calculated with EMT in the ASE framework. The NVT are used to perform the MDs of the two clusters. The Berendsen thermostat is used at every step. A time step is 0.5 fs, the temperature is 800 K, and 800 steps are used. The initial structure of Cu5 is build from an fcc of Cu(111) with size $2 \times 2 \times 1$ with a bridged copper atom. Similarly, Cu13 is an fcc structure of Cu(111) with size $2 \times 2 \times 3$ and a bridged copper atom.

O_2 at platinum

A database of two oxygen atoms adsorbed both on on-top sides at a platinum(100) is calculated with PBE in GPAW. The fixed platinum(100) surface is constructed of $3 \times 3 \times 3$ atoms. MD is performed with NVT, and the Berendsen thermostat is used at every step. The step size is 0.5 fs, the temperature is 800 K, and 800 steps are used.

Water molecules at platinum

The last database is a MD of four water molecules on a platinum(111) surface with $3 \times 2 \times 3$ atoms. All atoms are able to move. PBE is used as the exchange-correlation functional in GPAW. The NVT method is used with the Berendsen thermostat used at every step, the stepsize is 0.5 fs, the temperature is 300 K, and 800 steps are used.

B.2 Paper II

Machine-learning enabled optimization of atomic structures using atoms with fractional existence

Casper Larsen, Sami Kaappa, Andreas Lynge Vishart, Thomas Bligaard, and Karsten Wedel Jacobsen

Submitted to *Physical Review Letters*

arXiv preprint: https://arxiv.org/abs/2211.10342

Machine-learning enabled optimization of atomic structures using atoms with fractional existence

Casper Larsen,[1] Sami Kaappa,[1, 2] Andreas Lynge Vishart,[3, 4] Thomas Bligaard,[3, 4] and Karsten Wedel Jacobsen[1]

[1] *CAMD, Department of Physics, Technical University of Denmark, Kongens Lyngby, Denmark*
[2] *Computational Physics Laboratory, Tampere University, P.O. Box 692, FI-33014 Tampere, Finland*
[3] *CatTheory, Department of Physics, Technical University of Denmark, Kongens Lyngby, Denmark*
[4] *ASM, Department of Energy Conversion and Storage,*
Technical University of Denmark, Kongens Lyngby, Denmark

(Dated: November 24, 2022)

We introduce a method for global optimization of the structure of atomic systems that uses additional atoms with fractional existence. The method allows for movement of atoms over long distances bypassing energy barriers encountered in the conventional position space. The method is based on Gaussian processes, where the extrapolation to fractional existence is performed with a vectorial fingerprint. The method is applied to clusters and two-dimensional systems, where the fractional existence variables are optimized while keeping the atomic positions fixed on a lattice. Simultaneous optimization of atomic coordinates and existence variables is demonstrated on copper clusters of varying size. The existence variables are shown to speed up the global optimization of large and particularly difficult-to-optimize clusters.

The atomic-scale structure is of critical relevance to the physical and chemical properties of materials and nanoparticles. In the low temperature limit, the most stable atomic configuration is found by minimizing the total energy, but the optimization problem is difficult because of many metastable states, and, in many cases, the total energy evaluations are computationally time consuming.

To address these problems several algorithms of automatized structure prediction have been proposed [1] including random searches [2], genetic searches [3–6], basin hopping [7] and particle swarm optimizations [8]. Central to most of these methods is that they rely on carrying out large numbers of time-consuming calculations with density functional theory (DFT) or other quantum chemistry methods. To circumvent the time-issue of DFT without compromising the accuracy of the calculations, Gaussian processes have shown effective in constructing surrogate potential energy surfaces (PES) [9, 10]. These surfaces can be explored by random searching and updated by Bayesian search methods as demonstrated with the so-called GOFEE ('Global Optimization with First-principles Energy Expression') algorithm in Ref. 11. This methodology is generalized to include training on forces in the BEACON ('Bayesian Exploration of Atomic Configurations for OptimizatioN') code [12]. In Ref. 13, GOFEE is shown to decrease the number of energy evaluations necessary to find the global minimum by up to several orders of magnitude compared to traditional algorithms. Central to GOFEE/BEACON is the representation of atomic configurations by means of a fingerprint, which is invariant under translation, rotation, and inversion, and also under the permutation of atoms of the same chemical element.

It has been shown that the efficiency of random searching can be improved by inclusion of hyperdimensions [14]. The extra dimensions make it possible to circumvent barriers in the usual configuration space. However, the energy function has to be defined for the extra hyper-dimensions. This can be done for some analytic interatomic potentials, but it is not clear how to do this in the case of potential energy surfaces based on quantum mechanical calculations.

An alternative way to increase the dimensionality of configuration space and circumvent barriers is to interpolate between chemical elements ('ICE') as implemented in the ICE-BEACON code [15]. Here, additional dimensions are introduced so that an atom can be a fractional mixture of two chemical elements. The extension of the energy function to the extra dimensions is performed through a Gaussian process with a fingerprint, which allows for fractional chemical identities.

In this paper, we apply the idea of expanded dimensionality in a new way by introducing extra variables, which allow the atoms to have partial existence. The idea is that additional atoms of fractional existence can act as candidate sites for real atoms, allowing existence to be transferred from less to more favorable sites over arbitrarily long distances bypassing energy barriers in the conventional position space. Since some of the atoms end up with very little or no existence we shall refer to the additional atoms as ghost atoms, and we will refer to the approach as Ghost-BEACON.

In the model, a system with N atoms is treated as a surrogate system with $N^* > N$ atoms, where every atom (with index i) is given a fractional existence $q_i \in [0, 1]$ with the constraint that the fractions sum to the number of real atoms $\sum_i^{N^*} q_i = N$. The system is thus characterized by $3N^*$ spatial coordinates and N^* existence variables. The existence variables are incorporated into a structural fingerprint with radial and angular parts that resemble the corresponding distribution functions. The radial part reads

$$\rho^R(r) = \sum_{\substack{i,j \\ i\neq j}} q_i q_j \frac{1}{r_{ij}^2} f_c(r_{ij})\, e^{-|r-r_{ij}|^2/2\delta_R^2} \qquad (1)$$

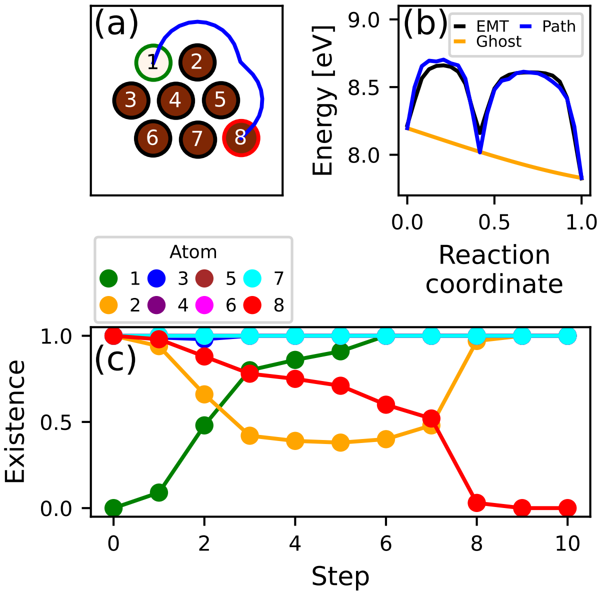


FIG. 1. (a) The 2D test system with 8 atoms, labeled from 1 to 8. In this configuration, atom 1 is a ghost atom, and atoms 2-8 are real. The blue curve shows the real-space minimum-energy path, where atom 8 is moved to the empty site 1. (b) Different energy profiles while moving the atom from site 8 to site 1 in (a). The black curve shows the EMT energies along the minimum-energy path, and the blue curve shows surrogate energies along the same path. The yellow curve shows the energy profile in the case where no atoms are moved, but the existence is transferred from atom 8 to atom 1. (c) The variation of the existence variables during the transfer of existence from atom 8 to 1.

where r is the distance variable, r_{ij} are the interatomic distances, f_c a cutoff function, and δ_R a length parameter. The angular part has a similar form. (Please, see details of the machine learning model and the fingerprint in the Supplemental Material [16]).

The radial fingerprint is in general quadratic in the existence variables. However, let us consider a situation where all atoms either fully exist ($q = 1$) or are completely removed ($q = 0$) except for two atoms, say numbers 1 and 2, whose distance is larger than the cutoff distance. In that case, the fingerprint becomes linear in q_1 and q_2. If we furthermore assume that the surroundings of the two atoms are identical, the transfer of existence from atom 2 to atom 1 ($q_2 = 1 - q_1$) leaves the fingerprint *completely unchanged* during the transfer. This means that any machine-learning model based on the fingerprint shows no energy barrier for the process. This analysis also holds if the angular fingerprint is included. (Shown explicitly in Supplemental Material [16], Fig. S1).

To illustrate the removal of energy barriers further, we show in Fig. 1(a) a system with 7 copper atoms accompanied by a ghost atom with the energies calculated with an effective-medium-theory (EMT) interatomic potential [17, 18]. We investigate the energy profile of moving an atom from a less favourable site (site 8) to a more favourable one (site 1) by following the trajectory shown in blue, which is the minimal-energy path found with a nudged-elastic-band (NEB) calculation [19, 20]. We compare this motion to the alternative path of existence transfer allowed by the new existence variables. A Gaussian-process surrogate model is trained on 8 points along the NEB trajectory. The black curve in Fig. 1(b) shows the EMT energies along the NEB path, while the blue curve is the surrogate energy along the same path. The blue curve roughly matches the black one, as expected, showing two energy barriers in the energy landscape corresponding to atom 8 bypassing atoms 5 and 2. The yellow curve in Fig. 1(b) shows the energy during the transfer of existence from atom 8 to 1 with the reaction coordinate $q_1 = 1 - q_8$ and all other existence variables fixed. The energy is almost linear with no potential barrier which means that the transfer of the atom from site 8 to 1 is favoured and straightforward in the existence space.

Figure 1(c) visualizes the energy minimization process where initially $q_1 = 0$ and $q_i = 1$ for $i = 2, 3, \dots, 8$. During the relaxation, the existence of atom 8 decreases while the existence of atom 1 increases. Interestingly, the process also involves atoms 2 and 3, which temporarily lose some of their existence. At the end of the relaxation, the existence has been completely transferred from atom 8 to atom 1.

We further illustrate the property of the PES when varying the existence variables in Fig. 2. Atom 8 is now moved along the indicated linear path in Fig. 2(c) when having different amounts of existence q_8, where the remaining existence is taken up by atom 1, $q_1 = 1 - q_8$. Atom 8 is seen to be more weakly interacting with the rest of the cluster when its existence is reduced, but the bonding distance remains essentially the same. This means that an atom with a small existence will tend to position itself at similar geometries as real atoms making the transfer of existence more relevant. However, the figure also shows that an atom with vanishing existence does not interact. This also follows from the fact that such an atom does not contribute to the fingerprint. Atoms with zero existence can therefore float freely around making it unlikely that they take part in optimization. For efficient structure optimizations, it is therefore necessary to introduce a lower bound for the existence variables and consequently increase the total existence.

It should be noted that the extension of the machine learning model to the fractional existence space is an extrapolation that cannot be controlled by the addition of data points. The quality of the model therefore depends strongly on the way the existence fractions are included in the fingerprint and the choice of hyperparameters for the machine learning model.

We now turn to structural optimizations where the energies and forces are based on DFT. The DFT calculations are performed using GPAW [21, 22] and the Atomic Simu-

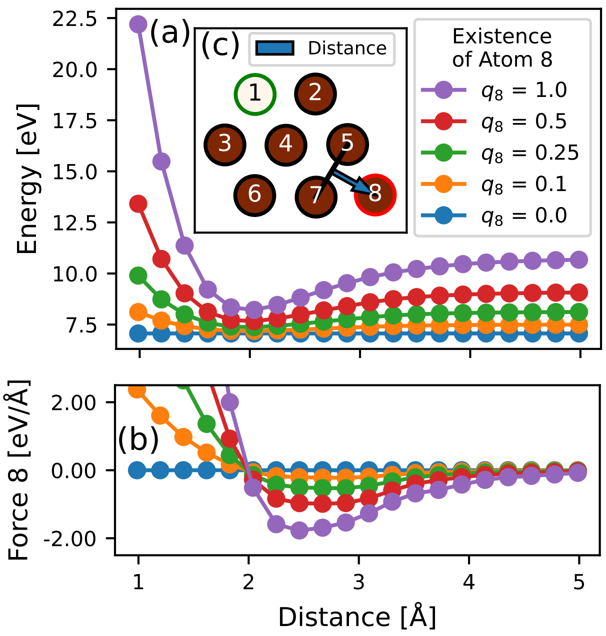


FIG. 2. (a) Energy curve and (ba) force curve of copper atom 8 as a function of the distance between copper atom 8 and the remaining cluster along the direction of the blue arrow depicted in (c) for different existence fractions of atom 8. Training is done with EMT on 10 different distances of atom 8. All existence not carried in atom 8 is placed in atom 1 ($q_1 = 1 - q_8$). The energy curves are seen to exhibit a minimum at approximately the same distance.

lation Environment [23, 24]. We apply the Perdew-Burke-Ernzerhof [25] exchange-correlation functional. The plane wave cutoff is 700 eV and the Fermi temperature is 0.1 eV. Only the Γ-point is used for k-point sampling except for graphene on a dense grid (Fig. 3) where (3,2,1) k-points are used. When performing relaxations with DFT, we use as convergence criterion that all atomic forces are smaller than 0.01 eV/Å.

The optimization algorithm is similar to the one of ICE-BEACON but with existence variables instead of chemical element interpolation: given a database of structures with DFT calculated energies and forces, a surrogate PES is constructed using a Gaussian process where the structures are described by the fingerprint. All systems in the database have N atoms, but the surrogate model can be used to make predictions for systems with N^* atoms with fractional existence. The surrogate PES is explored with random searching, that is with 40 local relaxations based on random initial configurations. The relaxations can be performed in either the atomic coordinates or the fractional existence variables, or both. If the existence variables take on fractional values after relaxation, the N largest fractions are set to 1, and the remaining to 0. The relaxed structures are evaluated with an acquisition function using the predicted energy and its uncertainty, and the structure with the lowest value is added to the DFT database. This procedure is iteratively repeated keeping track of the low energy structures obtained. The full simulation procedure is repeated to obtain statistics of the performance. Details of the algorithm including the computational parameters can be found in Supplemental Material [16].

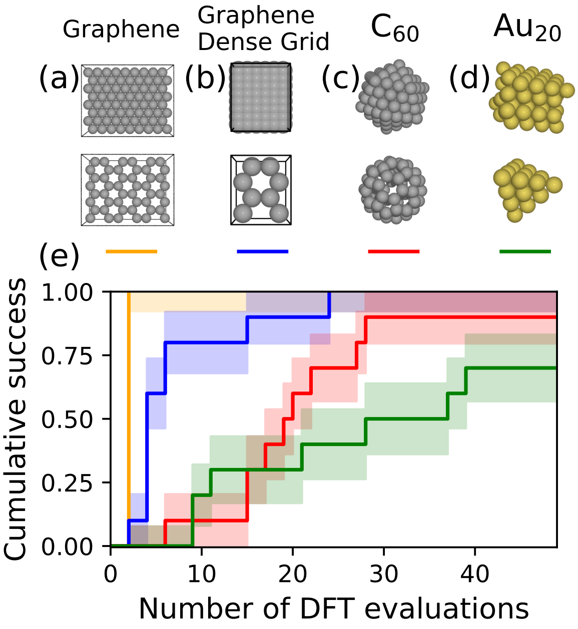


FIG. 3. (a-d) Atomic grids (top) and global minimum energy structures (bottom) of (a) carbon (48 atoms) on a periodic triangular lattice (72 atoms), (b) carbon (8 atoms) on a dense rectangular lattice (48 atoms total), (c) C_{60} on a 147 atoms icosahedral grid, and (d) Au_{20} on a 64 atoms fcc grid. (e) Success curves for finding the global minimum energy structure for each setup shown in (a-d). Only the existence variables are optimized while keeping the atomic positions fixed on the grid. The uncertainties are Bayesian estimates.

We first consider some examples where the atomic positions are fixed and where only the existence variables are optimized. Fig 3(a)-(d) show four different systems, which are (a) a single layer of carbon atoms on a periodic triangular lattice with an equilibrium interatomic distance of 1.42 Å corresponding to the one of graphene. The system contains a total of 72 atoms with 48 real atoms, which is the number of atoms corresponding to a layer of graphene. (b) A dense layer of carbon atoms on a periodic rectangular grid with interatomic distance $a = 0.710$ Å in one direction and $0.5\sqrt{3}a$ in the other direction. The total number of atoms is 48 with 8 real atoms again corresponding to the density of graphene. (c) An icosahedron of carbon atoms with 147 atoms in total and 60 real atoms with an interatomic distance of 1.44 Å between atoms belonging to the same icosahedral layer roughly agreeing with the bond lengths for a Bucky ball. (d) A cluster of fcc gold containing a total of 64 atoms and 20 real atoms.

Each optimization has an initial training set of two random sets of existence variables: one where the atoms are chosen by random and one where the atoms are chosen by random but so that the final structure is connected. The obtained minimum-energy structures for the four systems are shown in the lower panel of Fig. 3(a)-(d) The minimum-energy structure for (a) and (b) is a graphene layer, for (c) it is a C_{60} bucky ball, and for (d) it is the tetrahedral Au_{20} cluster [26]. The statistics of the optimizations are shown in the success curves in Fig 3(e). In all four cases 10 independent simulations have been performed, and the success curves show the fraction of simulations, which have found the lowest-energy structure as a function of the number of DFT calculations being performed.

The algorithm succeeds in finding the global optimum within 50 DFT calculations in 10/10 runs for both grid types of graphene and in 9/10 and 7/10 attempts for C_{60} and Au_{20}, respectively. Finding the structure of graphene on the standard triangular lattice proved to be a particularly easy task for the algorithm, which is probably due to the high degree of regularity of the grid and due to the high N/N^* ratio as compared to the problem of Au_{20}, for example.

The method also allows for simultaneous optimization of atomic coordinates and existence fractions as we shall now illustrate with copper clusters of varying size. We compare the performance of BEACON, which optimizes in only the configuration space of atomic coordinates, and the present approach, Ghost-BEACON, which optimizes in both configuration space and existence variables. We consider clusters of sizes 10, 20, and 30 atoms and in each case we add 50% ghost atoms and perform 20 independent simulations. The resulting minimum-energy structures are shown in Fig. 4 together with the success curves, where success is declared when a structure is within 0.1 eV of the lowest energy encountered across all runs of a give cluster size. Further analysis shows that the declared successful structures for Cu_{10} are all identical, while in the case of Cu_{20} two distinct structures are identified. In the case of Cu_{30} several structures have low energies, most of them slight alterations of the structures shown in (c).

We first note that the number of DFT calculations necessary to determine low energy structures does not vary monotonically with cluster size. The Cu_{10} cluster requires considerably more computational effort than Cu_{20}. This might seem surprising as the number of variables to consider in the optimization of course increases with cluster size. However, it should be recalled that we are doing random searching on the surrogate PES (with or without the existence variables) starting from random initial configurations, and the basin of attraction for the different local minima might vary substantially. This is the case for Cu_{10}, where the 3rd lowest energy structure is found more frequently than the ground state. (Shown with success curves in Supplemental Material [16]Fig. S2).

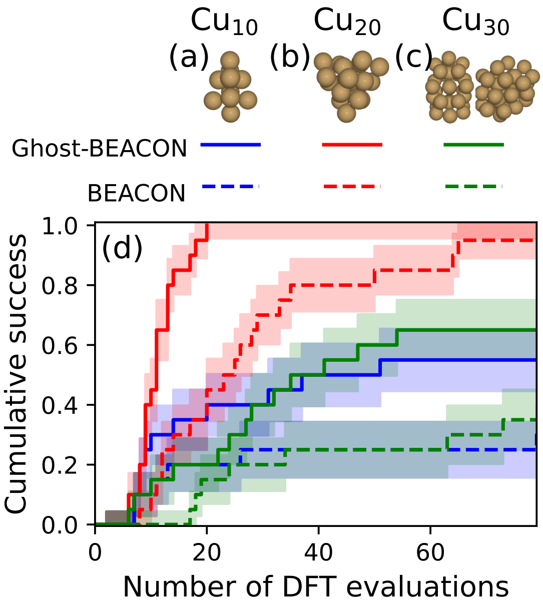


FIG. 4. (a-b) Global minimum structures of Cu_{10} and Cu_{20} and(c) the two lowest energy minima of Cu_{30} being so close in energy that they are almost inseparable. (d) Success curves of 20 independent runs of each 80 DFT-calculations without ghost atoms (BEACON) and with ghost atoms constituting 1/3 of the total number of atoms (Ghost-BEACON) for optimization of Cu_{10} (5 extra atoms), Cu_{20} (10 extra atoms) and Cu_{30} (15 extra atoms). Each iteration of the BEACON cycle was based on 40 surrogate relaxations. Each run had an initial training set of 2 random structures.

The presence of ghost atoms is seen to improve the searches considerably, in particular in the cases where BEACON does not easily identify the ground state.

The structures of Fig. 4(a-c) are different from the ones found using empirical potentials or tight binding molecular dynamics [27–29]. They are also different and lower in energy than the structures found using DFT in Ref. 30 as verified by relaxing all candidate structures with DFT.

The main function of the ghost atoms is to open new relaxation pathways as discussed above. To analyze this more, we construct a surrogate PES for Cu_{30} from a training set consisting of 151 configurations including some of the identified low-energy structures. We perform 1000 relaxations on the potential energy surface from random initial configurations for different choices of ghost atoms. The distributions of the obtained relaxed surrogate energies are shown in Fig. 5. Without any ghost atoms (the blue curve) we get the result that is obtained with BEACON. We see that when ghost atoms are introduced, the distribution is shifted to lower energies as an indication that the relaxations are not trapped as much in higher-lying local minima as is the case for BEACON. The inset in the figure shows the average energies of the distributions. Clearly the main

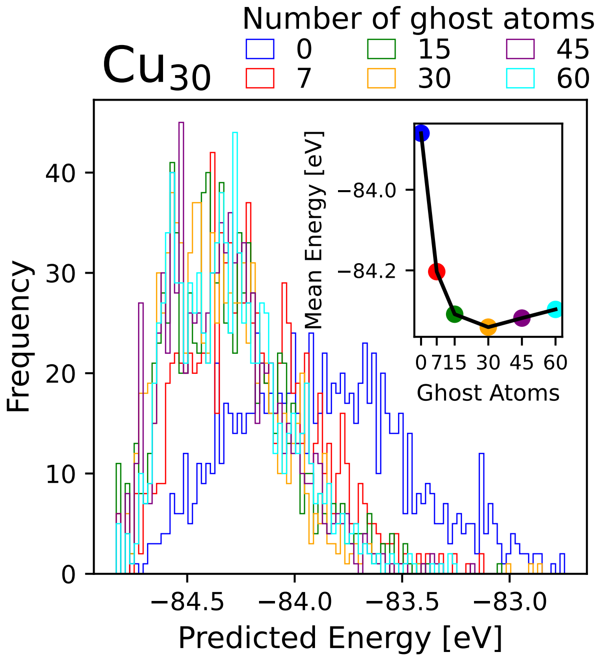


FIG. 5. Distribution of the energies obtained by 1000 relaxations on a surrogate PES for Cu_{30}. The inset shows the variation of the average energies as a function of the number of ghost atoms.

effect comes from introducing just a few ghost atoms into the system, and the effect quickly levels off with the number of ghost atoms. The fact that rather few ghost atoms improve the efficiency is also seen for Cu_{10} and Cu_{20} and is also observed in the success curves (Supplemental Material [16]Figs. S3 and S4).

Several modifications and extensions of the approach presented here are possible. It should be straightforward to combine the method with the ICE-approach. Each atom i would then carry a set of variables $q_i^A \in [0, 1]$, where A indicates the chemical element. The total existence of the atom would then be given by $q_i = \sum_A q_i^A \in [0, 1]$ with the constraint that the number of atoms N_A of element A is $N_A = \sum_i^{N^*} q_i^A$.

The example with graphene on a dense grid points to the possibility of restricting the atomic positions to a finely spaced grid and then only optimize the existence variables. However, this will require the treatment of very many atoms (one per grid point), which is not feasible with the current fingerprint.

In the present implementation, the sum of the existence variables is constrained to be the number of real atoms in the system. However, one could easily generalize this to treat open systems with a variable number of atoms controlled by a chemical potential. This would just correspond to a Lagrange-multiplier implementation of the constraint.

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, International Journal of Quantum Chemistry **121**, 044114 (2020).

[2] C. J. Pickard and R. J. Needs, Journal of Physics: Condensed Matter **23**, 053201 (2011).

[3] L. B. Vilhelmsen and B. Hammer, The Journal of Chemical Physics **141**, 044711 (2014).

[4] S. V. Lepeshkin, V. S. Baturin, Y. A. Uspenskii, and A. R. Oganov, The Journal of Physical Chemistry Letters **10**, 102 (2019).

[5] S. Lysgaard, D. D. Landis, T. Bligaard, and T. Vegge, Topics in Catalysis **57**, 33 (2014).

[6] M. Jäger, R. Schäfer, and R. L. Johnston, Nanoscale **11**, 9042 (2019).

[7] D. J. Wales and J. P. K. Doye, The Journal of Physical Chemistry A **101**, 5111 (1997).

[8] Z. Chen, W. Jia, X. Jiang, S.-S. Li, and L.-W. Wang, Computer Physics Communications **219**, 35 (2017).

[9] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, Physical Review Letters **104**, 136403 (2010).

[10] M. Todorović, M. U. Gutmann, J. Corander, and P. Rinke, npj Computational Materials **5**, 35 (2019).

[11] M. K. Bisbo and B. Hammer, Physical Review Letters **124**, 086102 (2020).

[12] S. Kaappa, E. G. del Río, and K. W. Jacobsen, Phys. Rev. B **103**, 174114 (2021).

[13] M.-P. V. Christiansen, N. Rønne, and B. Hammer, The Journal of Chemical Physics **157**, 054701 (2022).

[14] C. J. Pickard, Physical Review B **99**, 054102 (2019).

[15] S. Kaappa, C. Larsen, and K. W. Jacobsen, Physical Review Letters **127**, 166001 (2021).

[16] See Supplemental Material at [URL will be inserted by publisher] for additional information on the machine-learning model and the Ghost-BEACON algorithm, as well as supporting data on the performance of Ghost-BEACON.

[17] K. W. Jacobsen, J. K. Nørskov, and M. J. Puska, Physical Review B **35**, 7423 (1987).

[18] K. Jacobsen, P. Stoltze, and J. Nørskov, Surface Science **366**, 394 (1996).

[19] G. Mills and H. Jónsson, Physical Review Letters **72**, 1124 (1994).

[20] H. Jónsson, G. Mills, and K. W. Jacobsen, in *Classical and Quantum Dynamics in Condensed Phase Simulations*, Classical and Quantum Dynamics in Condensed Phased Simulations, Proceedings of the International School of Physics "Computer Simulation of Rare Events and Dynamics of Classical and Quantum Condensed-Phased Systems": Lerici, Villa Marigola, 7 July-18 July 1997, edited by B. J. Berne, G. Ciccotti, and D. F. Coker (World Scientific Publishing Company Incorporated, 1998) pp. 385 – 404.

[21] J. J. Mortensen, L. B. Hansen, and K. W. Jacobsen, Physical Review B **71**, 035109 (2005).

[22] 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, Journal of Physics: Condensed Matter **22**, 253202 (2010).

[23] 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, Journal of Physics: Condensed Matter **29**, 273002 (2017).

[24] Atomic Simulation Environment (ASE), `https://wiki.fysik.dtu.dk/ase/` (2020).

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

[26] J. Li, X. Li, H.-J. Zhai, and L.-S. Wang, Science **299**, 864 (2003).

[27] M. Böyükata and J. C. Belchior, Journal of the Brazilian Chemical Society **19**, 884 (2008).

[28] J. P. Doye and D. J. Wales, New journal of chemistry **22**, 733 (1998).

[29] M. Kabir, A. Mookerjee, and A. Bhattacharya, The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics **31**, 477 (2004).

[30] U. J. Rangel-Pena, R. L. Camacho-Mendoza, S. González-Montiel, L. Feria, and J. Cruz-Borbolla, Journal of Cluster Science **32**, 1155 (2021).

Supplemental Material for:
Machine-learning enabled optimization of atomic structures using atoms with fractional existence

Casper Larsen,[1] Sami Kaappa,[1,2] Andreas Lynge Vishart,[3,4]
Thomas Bligaard,[3,4] and Karsten Wedel Jacobsen[1]

[1] *CAMD, Department of Physics, Technical University of Denmark, Kongens Lyngby, Denmark*

[2] *Computational Physics Laboratory, Tampere University, P.O. Box 692, FI-33014 Tampere, Finland*

[3] *CatTheory, Department of Physics, Technical University of Denmark, Kongens Lyngby, Denmark*

[4] *ASM, Department of Energy Conversion and Storage, Technical University of Denmark, Kongens Lyngby, Denmark*

(Dated: November 24, 2022)

MACHINE LEARNING MODEL

Fingerprint

The fingerprint is based on the one used in BEACON [1] with the inclusion of existence fractions $q_i \in [0, 1]$ for each atom i. It is similar to the one used in ICE-BEACON [2] with the difference that in ICE-BEACON all atoms have both a fraction $q_{i,A}$ of element A, and a fraction for element B that satisfies $q_{i,B} = 1 - q_{i,A}$, while here q_i denotes the total existence of the atom.

The fingerprint is denoted by $\rho(\mathbf{x}, Q)$, where $\mathbf{x}$ is the full set of Cartesian coordinates and Q is the full set of existence fractions. $\rho(\mathbf{x}, Q)$ is divided into a radial part, $\rho^R(r; \mathbf{x}, Q)$ and an angular part, $\rho^\alpha(\theta; \mathbf{x}, Q)$, which for a single-element system is given by:

$$\rho^R(r; \mathbf{x}, Q) = \sum_{\substack{i,j \\ i \neq j}} q_i q_j \frac{1}{r_{ij}^2} f_c(r_{ij}; R_c^R)\, e^{-|r - r_{ij}|^2/2\delta_R^2} \tag{S1}$$

$$\rho^\alpha(\theta; \mathbf{x}, Q) = \sum_{\substack{i,j,k \\ i \neq j \neq k}} \left(q_i q_j q_k f_c(r_{ij}; R_c^\alpha) f_c(r_{jk}; R_c^\alpha) \cdot e^{-|\theta - \theta_{ijk}|^2/2\delta_\alpha^2} \right) \tag{S2}$$

$$f_c(r_{ij}; R_c) = \begin{cases} 1 - (1+\gamma)\left(\frac{r_{ij}}{R_c}\right)^\gamma + \gamma\left(\frac{r_{ij}}{R_c}\right)^{1+\gamma} & \text{if } r_{ij} \leq R_c \\ 0 & \text{if } r_{ij} > R_c \end{cases} \tag{S3}$$

where the indices i, j, and k run over all atoms. Here r_{ij} is the distance between atoms i and j, θ_{ijk} is the angle between atoms i, j and k, and f_c is a smooth cutoff function going to zero at the radial and angular cutoff radii R_c^R and R_c^α, respectively. γ is a parameter set to 2. Hence ρ^R describes a sum over all pairs of atoms whereas ρ^α describes a sum over all triplets. The full fingerprint $\rho(\mathbf{x}, Q)$ is created by concatenating ρ^R and ρ^α.

In Eqs. S1 and S2, the values for R_c^R and R_c^α are fixed for a given system but scaled with the covalent radius r_{cov} of the element as $R_c^R = 5r_{cov}$ and $R_c^\alpha = 3r_{cov}$. The constants $\delta_R = 0.4$ Å and $\delta_\alpha = 0.4$ rad are identical for all systems.

Gaussian process in the Ghost-BEACON framework

Following the notation for the fingerprint, the energies and forces, $\mu = (E, -F)$, are calculated with the standard expression for a Gaussian Process [3, 4]:

$$\mu(\mathbf{x}, Q) = \mu_p(\mathbf{x}, Q) + K(\rho[\mathbf{x}, Q], P)C(P, P)^{-1}(y - \mu_p(X)) \tag{S4}$$

where $\mu_p(\mathbf{x}, Q)$ and $\rho(\mathbf{x}, Q)$ are the prior mean and the fingerprint, respectively, K and C are the covariance matrix without and with regularization, P a matrix containing the training data fingerprints, y the training data targets and $\mu_p(X)$ the prior function applied to all structures in the training data. The uncertainty of the predicted energy is given by:

$$\Sigma(\mathbf{x}, Q) = \left\{\tilde{K}(\rho[\mathbf{x}, Q], \rho[\mathbf{x}, Q]) - K(\rho[\mathbf{x}, Q], P)C(P, P)^{-1}K(P, \rho[\mathbf{x}, Q])\right\}^{1/2}, \tag{S5}$$

where $\tilde{K}(\rho(\mathbf{x}, Q), \rho(\mathbf{x}, Q))$ represents the covariance matrix for the fingerprint.

The applied kernel function for the covariance matrices is a squared exponential kernel function (SE). The SE uses a prefactor (σ^2) and one length-scale (l) hyperparameters (the routine for optimization of the hyperparameters is described below). The covariance matrix between two atomic configurations has three components[1, 5]. The first components are the covariances between energies (k), the second are the covariances between energies and forces ($\nabla_i k$), and the third component are the covariances between forces ($\nabla_i \nabla_j k$). ∇_i is the gradient operator with respect to the Cartesian coordinates $\mathbf{x}_i$. The covariance matrix is written as

$$K(\rho_1, \rho_2) = \begin{bmatrix} k(\rho_1, \rho_2) & (\nabla_2 k(\rho_1, \rho_2))^\top \\ \nabla_1 k(\rho_1, \rho_2) & \nabla_1(\nabla_2 k(\rho_1, \rho_2))^\top \end{bmatrix}. \tag{S6}$$

We observe from Eq. S4 that K and $\mu_p(\mathbf{x}, Q)$ are the only terms including the existence fractions. Details about the construction of K, C, and y are reported in Ref. [1]. Keeping the order of all terms but simplifying the notation, we can rewrite Eq. S4 as

$$\mu = \mu_{p,\mathbf{x}} + KC^{-1}(y - \mu_{p,X}). \tag{S7}$$

If we denote the number of atoms by N, the number of elements per data point will be $F = 1 + 3N$ for one energy and $3N$ force components. If we further denote the number of structures in our training data D, the full training data will include DF features. Keeping the order of terms in Eq. S7, we have the following dimensions:

$$[F] = [F] + [F \times DF][DF \times DF][DF] \tag{S8}$$

which is the standard scenario for a Gaussian process. When predicting features on a structure with N^* atoms (comprising the real and the ghost atoms), the amount of predicted features becomes $G = 1 + 3N^*$, but the number of features on all structures in the training data is still F and hence Eq. S8 becomes

$$[G] = [G] + [G \times DF][DF \times DF][DF] \tag{S9}$$

Prior function

For the simultaneous optimization of positions and existence fractions of Figs. 3 and 4 in the main text, a repulsive prior modified to include the existence fractions is used [1, 6]:

$$\mu_p(\mathbf{x}, Q) = \mu_c + \sum_{\substack{i,j \\ i \neq j \\ r_{ij} < 2R}} q_i q_j \left(\frac{2\sigma_p \tilde{r}_{cov}}{r_{ij}} \right)^{12}, \tag{S10}$$

where σ_p is a repulsive constant set to 0.4 and $\tilde{r}_{cov}$ is an atomic radius set to be $0.8 r_{cov}$ of the element and μ_c is a constant prior. This prior is chosen to disfavor atoms with overlapping atomic radii, but in such a way that low existence atoms do not interfere with the clustering of high existence atoms. For all other simulations the prior is set to a constant value $\mu_p = \mu_c$ which is updated throughout the run.

Acquisition function

We use the acquisition function f for a structure $\mathbf{x}$ given by $f(\mathbf{x}) = \mu(\mathbf{x}) - \kappa \Sigma(\mathbf{x})$, where $\kappa = 2$ is a constant while $\mu(\mathbf{x})$ and $\Sigma(\mathbf{x})$ are the predicted energy and uncertainty of Eq. S4 and Eq. S5 [1, 6]. The dependency on Q is omitted as the acquisition function is always evaluated on structures without ghost atoms.

The acquisition function is used to select which of the relaxed structures to include in the DFT database. However, sometimes the relaxations mostly reproduce an already investigated structure. It is therefore an advantage to discard structures that are closer than a certain distance, d_{fp}, in fingerprint space from already known structures. For the optimization of both atomic coordinates and existence values we set $d_{\mathrm{fp}} = 5$. For the optimization on a grid d_{fp} was set to a small value to exclude already visited structures without disqualifying any other structures.

Robust determination of hyperparameters

During the BEACON and Ghost-BEACON runs, the hyperparameters are updated by using the maximum a posteriori probability (MAP) for the hyperparameters given the training data, $p(l, \sigma_r, \sigma | y)$. A uniform prior is considered for the prefactor, σ, whereas a log-normal prior distribution is used for the length-scale hyperparameter, l, as explained in the section below. The noise, σ_n, is set by a relative noise, $\sigma_r^2 = \frac{\sigma_n^2}{\sigma^2}$.

The MAP is calculated by using the analytical solution of the prefactor, σ^2_{MLE}, from maximizing the posterior distribution:

$$\sigma^2_{\mathrm{MLE}} = \frac{1}{DF}(y - \mu_p)^\top C_0^{-1}(y - \mu_p) \tag{S11}$$

where $C_0(P, P) = K_0(P, P) + \sigma_r^2 I$ is the covariance matrix of the training data without the prefactor and a relative-noise. $K_0(P, P)$ denotes the covariance matrix of the training data without the prefactor and noise. The same relative-noise is used for energy and force contributions. The log-posterior distribution, LP, is:

$$LP(l, \sigma_r, y) \propto MLL(l, \sigma_r, y) + \ln\left(p(l)\right) \tag{S12}$$

where the MLL is the maximum log-likelihood with respect to the prefactor hyperparameter. The MLL is expressed as:

$$\begin{aligned} MLL =& \frac{-1}{2}\left(DF + \ln\left(|C_0|\right) + DF\ln\left(\frac{1}{DF}(y - \mu_p)^\top C_0^{-1}(y - \mu_p)\right) + DF\ln\left(2\pi\right)\right) \\ =& \frac{-1}{2}\left(DF + \sum_{i=1}^{DF}\ln\left([\Lambda]_{ii} + \sigma_r^2\right) + DF\ln\left(\frac{1}{DF}\sum_{i=1}^{DF}\frac{[E^\top(y - \mu_p)]_i^2}{[\Lambda]_{ii} + \sigma_r^2}\right) + DF\ln\left(2\pi\right)\right) \end{aligned} \tag{S13}$$

where E is the eigenvectors and Λ is the diagonal matrix with the eigenvalues of the covariance matrix without prefactor and relative-noise hyperparameters, $K_0(P, P) = E\Lambda E^\top$. All relative-noise hyperparameter values can be searched from a single eigendecomposition. A small noise is added to the covariance matrix to ensure it is invertible. However, a fixed relative-noise of 0.001 is used to avoid the maximum likelihood values that corresponds to the overfitting models.

A uniform grid with a spacing of 0.1 in the log-space of the length-scale hyperparameter is constructed from the mean nearest neighbour to 100 times the maximum Euclidean distance in the fingerprint space. All intervals surrounding a maxima of the log-posterior can be identified by using finite difference on the grid. Afterwards, a golden-section search is performed for all intervals containing a maxima.

The grid search method finds the global maxima of the log-posterior distribution under the constraints if the grid spacing is finer than the length of the basin of attraction.

The prior mean constant is optimized from the maximum likelihood [1] under the constraint that it must be greater than or equal to the average between the smallest and the mean energies of the training data as:

$$\mu_p = \begin{cases} \frac{(\min(Energy)+\text{mean}(Energy))}{2} & \text{if } \mu_p < \min(Energy) \\ \frac{\mathbf{U}^\top C(P,P)^{-1} y}{\mathbf{U}^\top C(P,P)^{-1}\mathbf{U}} & \text{otherwise} \end{cases} \tag{S14}$$

where $\mathbf{U}$ is a vector with the length of DF and has $U_i = 1$ for energy components and $U_i = 0$ for force components.

The prior distribution and constrained interval of the length-scale hyperparameter improves the model quality at small data sets at the beginning of a run, where the model could be likely to either overfit (short length-scales with low noise) or underfit (very large length-scales with high noise). At the beginning of a run, the length scale is set to 2.5 times the maximal distance in fingerprint space. The prefactor, noise and prior are updated at every BEACON cycle, whereas the length scale is updated every fifth cycle.

Prior distribution of the length scale

A prior distribution of the length scale is introduced to hinder the algorithm in over-fitting for small data sets and because it is observed that a longer length scale improves the interpolation in existence space. The length scale prior is defined as a log-normal distribution, i.e. a normal distribution in the logarithmic space:

$$P(l) = \frac{1}{l\sigma_{LN}\sqrt{2\pi}} \exp\Big(- \frac{(\ln(l) - \mu_{LN})^2}{2\sigma_{LN}^2} \Big), \tag{S15}$$

where μ_{LN} and σ_{LN}^2 are the mean and variance in the logarithmic space.

A simple estimate of the length is $l_0 = 0.5(\text{mean}(\Delta_{FP}) + \max(\Delta_{FP}))$, where Δ_{FP} are all the Euclidian distances between any two fingerprints. We set the parameters μ_{LN} and σ_{LN} using $\text{mode}(l) = \exp(\mu_{LN} - \sigma_{LN}^2) = l_0$ and take $\sigma_{LN} = 0.75$.

For the Gaussian processes for EMT-evaluated Cu structures, which are fitted to only few data points, we simply keep the initial estimate of the length as 2.5 times the maximal distance in the fingerprint space.

ALGORITHMIC DETAILS AND COMPUTATIONAL PARAMETERS

Random structure generator

In this study, all random configurations not placed on a grid are set up using a cubic box with a volume which is five times the sum of the volumes of atomic spheres with radii equal to the covalent atomic radii of the elements. The atoms initially placed randomly in the box are then repelled until all atom centers are at least $1.6 r_{cov}$ away from each other. 7.5 Å of vacuum is then added around the structure to complete the unit cell. This procedure ensures a similar initial atomic packing fraction independent on the number of atoms in the BEACON/Ghost-BEACON runs.

Random fraction generator

The random sampling of the initial existence values is done using the Dirichlet-Rescale algorithm [7, 8]. This allows for a uniform distribution of the existence values satisfying the constraints $q_i \in [q_{\min}, 1]$ and $\sum_i q_i = N + (N^* - N)q_{\min}$, where $0 \leq q_{\min} < 1$ is the lower existence bound.

Surrogate surface relaxations

The relaxations on the surrogate potential energy surface are performed using sequential least squares programming [9] as implemented in the SCIPY package [10]. This allows for efficient gradient-driven optimization under the inequality constraint that all atoms have an existence value between $q_{\min}$ and 1 as well as the equality constraint for the total amount of existence.

While optimizing the coordinates and the existence fractions simultaneously, the existence of an atom might fall to zero, effectively removing its interactions with the rest of the system. To counteract this unwanted effect in the algorithm and to proceed with the most efficient optimization, a lower limit to the existence is introduced, and the following procedure is adopted:

1) Initialize a system of random atomic positions and existence fractions between $q_{\min}^{\text{init}}$ (> 0) and 1 with a total existence of $N + (N^* - N)q_{\min}^{\text{init}}$.
2) Relax the system on the surrogate PES for n_{relax} steps.
3) Decrease the lower limit in n_D steps of $q_{\min}^{\text{init}}/n_D$ and, at each level, perform a relaxation with n_d steps.
4) Relax the system for n_p steps with all existence variables fixed to 0 or 1 to effectively remove the ghost atoms.

The relaxations are terminated if all predicted forces are below 0.001 eV/Å. The low value is picked to counteract underestimation of forces in regions of large uncertainty on the potential energy surrogate surface. In this paper, the simultaneous relaxations of existence and positions are done with $n_{relax} = 200$, $q_{\min}^{\text{init}} = 0.05$, $n_D = 5$, $n_d = 20$, and $n_p = 100$.

The calculations on a grid, where only the existence variables are optimized, do not require a lower boundary. All non-ghost BEACON runs are performed with $n_{relax} = 400$ with $N^* = N$ and all fractions fixed to 1.

Declaration of success

Except for Fig. 4 in the main paper, a success is registered once a structure satisfies the correct nearest neighbor distribution for all atoms in the cluster as compared to the global minimum. This procedure is chosen to identify structures belonging to the correct basin.

Calculation of success curve uncertainty

To calculate the uncertainty of the success curves of the paper, a Bayesian approach was followed. A success curve composed of W independent runs can for a given number of DFT calculations be seen as a binary outcome of n successes and m failures such that the total number of attempts is always $W = n + m$. Using Bayes theorem with a uniform prior, the posterior probability of the chance of success $p_{success}$ becomes a Beta distribution $B(p|\alpha = n+1, \beta = m+1)$. We use the mode of this distribution mode$(p_{success}) = n/(n+m)$

as the value of the success curve. For the uncertainty, we use the square root of the variance

$$\sqrt{\mathrm{var}(p_{success})} = \sqrt{\frac{(n+1)(m+1)}{(n+m+2)^2(n+m+3)}}. \tag{S16}$$

SUPPLEMENTARY DATA

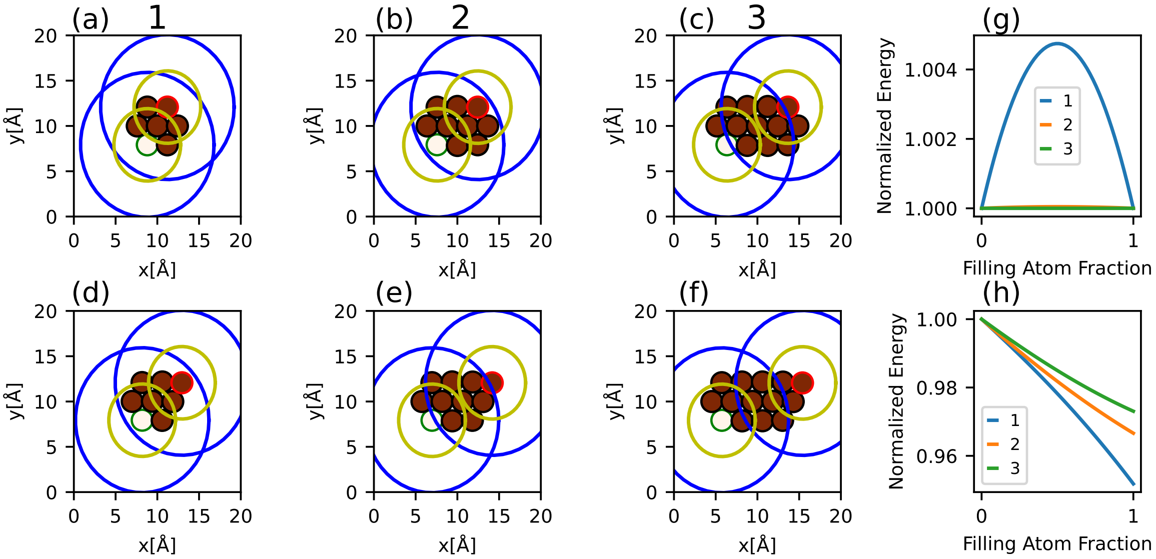


FIG. S1. Illustration of the point that existence transfer between atoms far away from each other does not involve energy barriers. The clusters are similar to the one in Fig. 1 of the main paper. The clusters have different distances between the two atoms indicated with red and green edges. In the upper row (a-c), the two atoms occupy identical sites while in the lower row (d-f) they are different. Blue and yellow circles indicate the radial and angular cutoff radii, respectively, of the red and green edge atoms. Figures (g) and (h) show the energy change during existence transfer from the red edge atom to the green edge atom with all other atoms being constrained at existence 1. The curves are normalized with respect to the energy of the initial configuration shown in figures (a-f). The figure shows that as the two atoms are separated from each other, the potential barrier is completely removed in the case of identical sites. When the sites are different, the energy decays monotonically towards the most stable site.

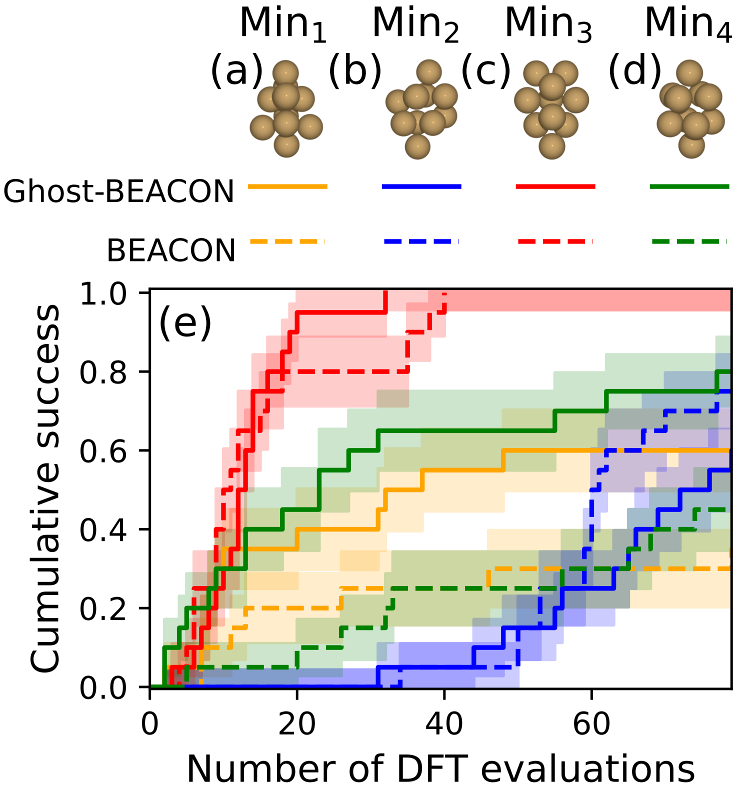


FIG. S2. (a-d) The 4 lowest discovered minimum energy structures for Cu_{10}. (e): Success curves of 20 independent runs of each 80 DFT-calculations. Each iteration of the BEACON cycle is based on 40 surrogate relaxations. Each run has an initial training set of 2 random structures

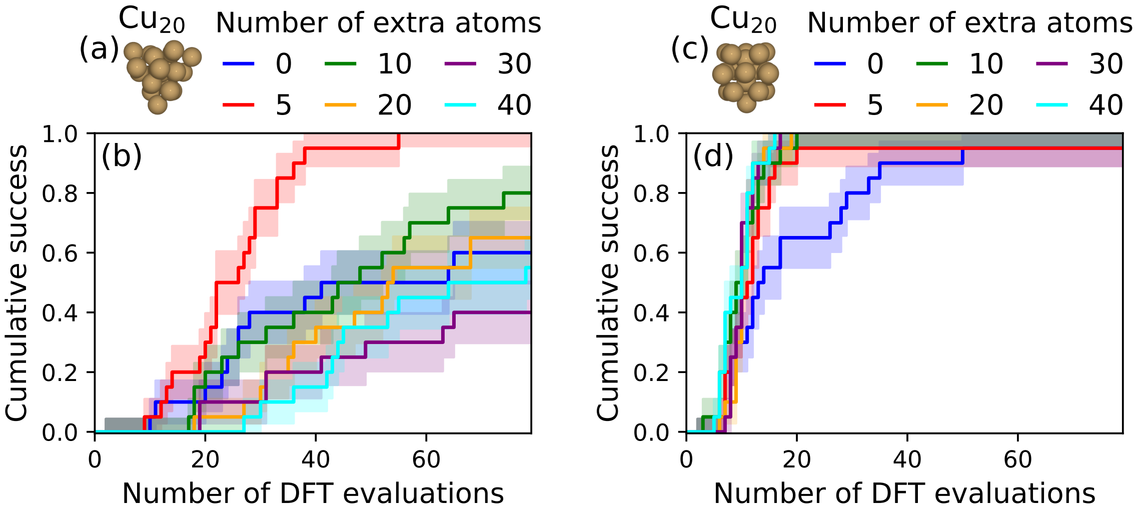


FIG. S3. (a) Global minimum and (c) second lowest energy structure of Cu$_{20}$. (b) and (d) Success-curves of 20 independent runs of each 80 DFT-calculations without ghost atoms (blue) and with 5 different numbers of ghost atoms for finding the structure shown in (a) and (c) respectively. Each iteration of the BEACON cycle is based on 40 surrogate relaxations. Each run has an initial training set of 2 random structures.

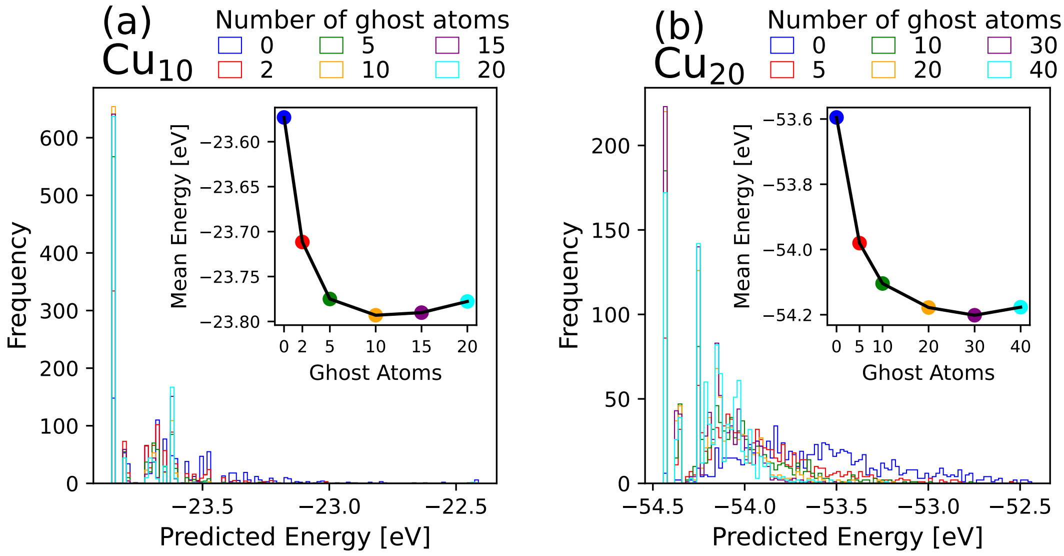


FIG. S4. Histograms of the predicted energies for 1000 surrogate relaxations of (a) Cu$_{10}$ and (b) Cu$_{20}$ for six different numbers of ghost atoms. The figures are similar to Fig. 5 in the main paper for Cu$_{30}$.

[1] S. Kaappa, E. G. del Río, and K. W. Jacobsen, Phys. Rev. B **103**, 174114 (2021).

[2] S. Kaappa, C. Larsen, and K. W. Jacobsen, Physical Review Letters **127**, 166001 (2021).

[3] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, 2006).

[4] J. Wu, M. Poloczek, A. G. Wilson, and P. Frazier, in *Advances in Neural Information Processing Systems* (2017) pp. 5267–5278.

[5] O.-P. Koistinen, F. B. Dagbjartsdóttir, V. Ásgeirsson, A. Vehtari, and H. Jónsson, The Journal of Chemical Physics **147**, 152720 (2017).

[6] M. K. Bisbo and B. Hammer, Physical Review Letters **124**, 086102 (2020).

[7] D. Griffin, I. Bate, and R. I. Davis, in *IEEE Real-Time Systems Symposium, RTSS 2020, Houston, Texas, USA, December 1-4, 2020* (IEEE, 2020).

[8] D. Griffin, I. Bate, and R. I. Davis, dgdguk/drs (2020).

[9] D. Kraft, ACM Trans. Math. Softw. **20**, 262–281 (1994).

[10] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, Nature Methods **17**, 261 (2020).

Technical
University of
Denmark

Anker Engelunds Vej , Building 301
2800 Kgs. Lyngby
Tlf. 4525 1700

www.energy.dtu.dk/english