Lukas Hörmann, BSc.

Overcoming the Configurational Explosion for Surface Structure Search by Coarse Grained Modeling and Bayesian Learning

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Technische Physik

eingereicht an der

Technischen Universität Graz

Betreuer

Dipl.-Ing. Dr.techn. Oliver T. Hofmann
Univ.-Prof. Dr. Peter Hadley

Institut für Festkörperphysik

Graz, April 2018

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Datum

Unterschrift

Acknowledgements

I would like to thank all those, who helped and supported me while working on this thesis. Above all, I want to express my sincere gratitude to Oliver Hofmann for supervising my work. He gave me freedom to pursue my own ideas, while providing the guidance to the successful conclusion of my master thesis.

Few things can be achieved in solitude. Hence, I thank my colleagues in the endeavor of structure search. Alexander, Andreas, Michael and Veronika helped to develop the methods and algorithms presented in this thesis and aided in refining my personal contribution through countless fruitful discussions.

In this breath, I also wish to thank the entire materials modeling team, who made the time I spent working on this thesis fun and enjoyable. I will miss the serious and sometimes not so serious debates about everything and anything.

On a more technical note, I gratefully acknowledge the Vienna Scientific Cluster 3 and the Argonne national laboratory for providing the computational resources, that made this thesis possible, as well as the FWF for financial support.

Looking at my time at the TU Graz, I am very thankful to my friends and colleagues. To name but a few, Andreas, Bernhard, Marianne, Michael, Sasha and Stefan shared the joys and sorrows of studying with me and I look back fondly at that time.

Finally, I would like to thank my family for their continuous support throughout my studies.

Overcoming the Configurational Explosion for Surface Structure Search by Coarse Grained Modeling and Bayesian Learning

In this thesis I present my contribution to SAMPLE. SAMPLE is a structure search tool, which employs coarse grained modeling and statistical learning to determine the potential energy surface of commensurate organic monolayers on inorganic substrates. Such monolayers often display rich polymorphism and form diverse structures in differently shaped unit cells. Since the properties of a material strongly depend on its structure, predicting these polymorphs from first principles is highly desirable. The main challenge hereby is the exponential increase of the number of possible polymorphs with the system size.

To overcome this configurational explosion, I discretize the potential energy surface. To this end, I first determine adsorption geometries, that isolated molecules adopt on the substrate. Secondly, I generate all possible unique substrate super cells, whose area lies within predefined boundaries. I use the local adsorption geometries as building blocks and combine them with each super cell to generate possible structures.

For a subset of these structures, I determine the adsorption energies using dispersion-corrected density functional theory (DFT). This subset serves as training data for a Bayesian linear regression algorithm, which allows to predict the adsorption energies of all possible polymorphs. I validate the prediction by using DFT to calculate the adsorption energies of a number of structures, that were not contained in the training set. The results of these calculations are then compared to the predicted adsorption energy, to calculate a prediction error.

Finally, I employ ab initio thermodynamics to generate phase diagrams.

I demonstrate the power of SAMPLE on the system of naphthalene on copper(111). For this system I predict the adsorption energies for a large number of structures and determine the phase diagram.

Überwinden der Konfigurationsexplosion bei Oberflächenstruktursuche mit *Coarse Grained Modeling* und Bayesschem Lernen

In dieser Arbeit präsentiere ich meinen Beitrag zu SAMPLE, einer Methode zur Oberflächenstruktursuche. SAMPLE nutzt *coarse-grained modeling* und statistisches Lernen, um die Potentialhyperfläche kommensurabler organischer Monolagen auf anorganischen Substraten vorherzusagen. Solche Monolagen zeigen oftmals einen ausgeprägten Polymorphismus und bilden vielfältige Strukturen in unterschiedlich geformten Einheitszellen. Da Materialeigenschaften stark von der Struktur abhängen, ist es sehr wünschenswert, solche Strukturen auf Basis von Grundprinzipien vorhersagen zu können. Die größte Herausforderung hierbei resultiert aus der Tatsache, dass die Anzahl der möglicher Strukturen exponentiell mit der Systemgröße ansteigt.

Um dieser Konfigurationsexplosion Herr zu werden, diskretisiere ich die Potentialhyperfläche. Hierzu bestimme ich zuerst die Adsorptionsgeometrien, welche isolierte Moleküle auf der Substratoberfläche annehmen. Im zweiten Schritt generiere ich alle möglichen inäquivalenten Superzellen des Substrats, deren Fläche innerhalb zuvor definierter Grenzen liegt. Ich benutze die lokalen Adsorptionsgeometrien als Bausteine und kombiniere sie mit den Superzellen, um mögliche Strukturen zu erzeugen.

Im nächsten Schritt wende ich die dispersionskorrigierte Dichtefunktionaltheorie (DFT) an, um die Adsorptionsenergien einer Teilmenge dieser Strukturen zu berechnen. Diese Teilmenge liefert die Trainingsdaten für eine Bayessche lineare Regression. Diese ermöglicht es, die Adsorptionsenergien aller möglichen Polymorphe vorherzusagen. Anschließend validiere ich die Vorhersage, indem ich die Adsorptionsenergien mehrere Strukturen, welche nicht Teil der Trainingsdaten waren, mit DFT nachrechne. Die Ergebnisse dieser Rechnungen werden dann mit den vorhergesagten Adsorptionsenergien verglichen, um die Unsicherheit der Vorhersage zu bestimmen.

Im letzten Schritt benutze ich die ab initio Thermodynamik, um Phasendiagramme zu erstellen.

Ich demonstriere die Leistungsfähigkeit von SAMPLE anhand des Systems von Naphthalen auf Kupfer(111). Für dieses System bestimme ich die Adsorptionsenergien einer großen Anzahl von Strukturen und berechnen das Phasendiagramm.

Contents

1 Introduction

1.1 Structure Search

1.1.1 Motivation for Structure Search

In 1988 John Maddox famously stated:

One of the continuing scandals in the physical sciences is that it remains in general impossible to predict the structure of even the simplest crystalline solids from a knowledge of their chemical composition.[1]

Many material properties such as solubility, tensile strength or electric conductivity depend on the crystal structure. Hence, the crystal structure is arguably the principal information about a substance. The ability to predict the crystal structure would therefore constitute a powerful tool to gain insight into a materials properties without producing it. This makes the pursuit of structure search a very worthwhile endeavor.

1.1.2 Configurational Explosion

One of the largest challenges for the pursuit of structure search is the immense number of possible structures one has to investigate. This circumstance is referred to as configurational explosion. In the framework of structure search, the term configurational explosion refers to a special case of combinatorialal explosion, that considers the spatial arrangement of atoms or molecules.

We will illustrate the configurational explosion based on the following example: Let us consider a monolayer of rigid molecules adsorbed on a flat substrate. For this system, we intend to find the most energetically favorable structure, that has one molecule in the unit cell. In this case we need to consider three degrees of freedom, translation in x and y direction as well as rotation. For the sake of argument, let us choose a discretization of 10 steps for translation in x and y, as well as 10 steps for rotation. Searching the most energetically favorable structure then requires to investigate $10 \cdot 10 \cdot 10 = 1000$ configurations.

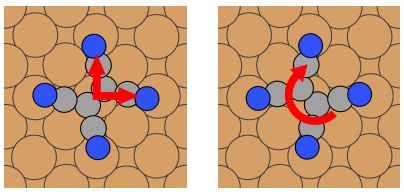


Figure 1: Independent degrees of freedom of a single rigid molecule adsorbed on a flat surface

Let us now expand the previous example and consider unit cells with two molecules. In this case, we need to take six degrees of freedom into account. We start by placing the first molecule into the unit cell - as before we find 1000 possible ways to do so. Now we take

each of these 1000 possible ways to add one molecule and add a second molecule to them. Neglecting collisions allows us to place molecule two independent of molecule one. Hence, for each position and orientation of molecule one, we find 1000 arrangements of molecule two. Therefore an exhaustive search for the *best* two-molecule configuration requires to look at $(10 \cdot 10 \cdot 10)^2 = 10^6$ configurations. Similarly for three molecules per unit-cell and nine degrees of freedom, the configuration space grows to $(10 \cdot 10 \cdot 10)^3 = 10^9$. Hence, for this example the number of configurations is given by n^f, where n is the discretization and f the number of independent degrees of freedom.

In general, constructing configurations involves a number of considerations we have neglected for this simple example (see chapter 2). For instance, we have to take care of molecule collisions, consider symmetries, select different discretizations for the respective degrees of freedom, etc.. However, the fact still remains, that the number of configurations grows exponentially with the number of independent degrees of freedom. Structure search thus poses an NP-hard problem[1], making exhaustive search methods unfeasible for all but the simplest of systems.

1.1.3 Methods used in Structure Search

Structure search poses a global optimization problem - structure search methods aim to find the minimum of the potential energy surface. The following chapter intends to briefly introduce a small and by no means complete list of such methods. The particular examples were chosen according to their relevancy to this thesis. On this note we will particularly focus on Bayesian learning.

The deliberations regarding local optimization, basin hopping and genetic algorithms are mainly based on the book *Modern Methods of Crystal Structure Search*[2]. The chapter on Bayesian learning is based on the book *Gaussian Processes for Machine Learning*[3] by Carl Edward Rasmussen and Christopher K. I. Williams.

1.1.3.1 Local Optimization

Local optimization[4] aims to improve a given starting structure by following the gradient of the potential energy surface. Employing local optimization allows us - as we will explain later - to reduce the number of degrees of freedom we need to consider for the global optimization problem.

Global optimization may for instance only consider the coordinates of a molecules' center of mass and its rotation when proposing a structure. Local optimization can be used to refine these *rough* structures. To do this, we need to consider all degrees of freedom, but since we merely intend to find the closest minimum of the potential energy surface, the search space is much smaller compared to a global search. In the overall picture the addition of local optimization, therefore greatly simplifies the global optimization problem.

[1] An NP-hard problem is a problem that most likely cannot be solved in polynomial time by a deterministic algorithm. This means that the computational effort grows more quickly with problem size than any polynomial function.

Such an approach is justified, because some degrees of freedom are correlated. For instance, if there are two atoms connected by a chemical bond (perturbations from the equilibrium distance entail a large energy penalty) and translate one of the two, then the other molecule has to be moved in a similar way to preserve the bond, i.e. the positions of the atoms are correlated.

The circumstance, that some atomic positions strongly depend on each other, allows us to define a reduced set of degrees of freedom. Such a set might include the coordinates of the molecules' centers of mass, or their rotations.

1.1.3.2 Basin Hopping

Basin hopping structure search[5] tries to find the global minimum of a potential energy surface by consecutively perturbing a structure. Staring from some initial structure, a basin hopping algorithm performs the following steps:

1. A move is proposed, say rotating or translating one or more atoms or molecules in the system. What form the move takes is decided randomly within predefined boundaries. Ideally the perturbation should move the structure into the *drainage basin* of a neighboring local minimum.
2. The energy of the new structure is evaluated by local optimization.
3. The new structure is accepted or rejected, based on the results from step two. One possible method for accepting/rejecting a move is based on the Metropolis-Hastings algorithm.

$$min\left[1, \exp\left(-\frac{E_{new} - E_{old}}{k_B T}\right)\right] \geq q \tag{1}$$

   Here q is a random number between 0 and 1. E_{new} is the energy after and E_{old} the energy before the move. Note that this scheme always accepts a move, if $E_{new} < E_{old}$. If a move is accepted, the new structure serves as starting point for the next move, otherwise the optimization continues from the previous structure.

By repeating these three steps we gradually sample the potential energy surface, identifying more and more local minima. Due to the heuristic nature of the basin hopping approach it is however uncertain to find the global minimum.

1.1.3.3 Genetic Algorithms

Genetic algorithms[4] are inspired by biological evolution, which is based on reproduction, mutation and (natural) selection. In the context of structure search, an algorithm based on these principles employs the following six steps:

1. A suitable representation for the system is defined, whereby each structure is represented by a *gene*. The key requirement is, that a structure must have a one-to-one correspondence to its *gene*.

2. The first generation - a set of possible structures - is initialized.
3. The fitness of each structure in the generation is determined. Different parameters may serve as the fitness function, such as the energy, Gibbs free energy, etc..
4. A crossover operator is employed to mix the genes from the fittest structure in a suitable way, to produce offspring structures. To illustrate the function of the crossover operator let us consider a system with two molecules A and B per unit cell. A crossover operator may use the position of molecule A from the first parent and the position of molecule B from the second parent to generate an offspring structure.
5. Mutations to the *offspring* structure are introduced via a mutation operator. A mutation may for example simply constitute a perturbation of a molecule.
6. The *offspring* structures are evaluated to determine their fitness. The best structures then form the next generation.

Steps three thru six are repeated until some stopping criterion is reached. Examples for such a criterion are energy convergence or a maximum number of iterations steps. Similar to the basin hopping approach, genetic algorithms are heuristic search methods.

1.1.3.4 Bayesian Learning

A common task for structure search is to find low energy structures (or structures with some other property) among a large number of possible structures. However, computational resources usually only allow to calculate the energies (or other properties) of a small subset of these structures. Bayesian learning allows to use the information gained form this subset of structures to predict energies (or other properties) of the remaining structures.

Bayesian learning is based on the Bayes' theorem:

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)} \tag{2}$$

Here H stands for a hypothesis, and E for the evidence. In the framework of structure search the evidence E might constitute a list of known energies (or other properties of interest), that belong to a subset of possible structures. In this context, the hypothesis H may then consist of some model, which supplies the energies for all structures. Finding an appropriate hypothesis would allow to predict the energies of all possible structures. Therefore we would like to gauge how good a given model is. This question can be expressed as the conditional probability $P(H|E)$ for a model H (hypothesis) given a list of known energies E (evidence). The advantage of Bayes' theorem is, that the unknown probability $P(H|E)$ can be expressed in terms of the known quantities $P(E|H)$ and $P(H)$. $P(E|H)$ is called likelihood and denotes the probability to reproduce the known energies E given the model H. The prior $P(H)$ is the probability of our model H without knowing any energies. In simple terms $P(H)$ expresses, what we know about the system without

doing any measurements or calculations. Finally $P(E)$ is the marginal likelihood, which can be expressed as a marginal distribution.

$$P(E) = \sum_i P(E|H_i) \cdot P(H_i) \tag{3}$$

There are a number of different approaches to implement Bayesian learning, such as Bayesian linear regression, kernel ridge regression or Gaussian process regression. In the following discussion we will focus on Gaussian process regression.

We will begin our discussion of Gaussian process regression by considering multivariate Gaussians and show that they can be brought into the form of the Bayes' theorem (equation 2). A multivariate Gaussian gives the probability distribution of a $n \times 1$ vector $\mathbf{x}$ with a $n \times 1$ mean vector $\boldsymbol{\mu}$ and a $n \times n$ covariance matrix C.

$$p(\mathbf{x}|\boldsymbol{\mu}, C) \;=\; ((2\pi)^n |C|)^{-1/2} \;\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T C^{-1}(\mathbf{x}-\boldsymbol{\mu})\right] \tag{4}$$

The covariance matrix is given by the following expression:

$$C_{ij} \;=\; \langle (x_i - \mu_i)(x_j - \mu_j) \rangle \tag{5}$$

Having presented this textbook definition, we shall explain how we can use the multivariate Gaussian for structure search.

When utilizing Bayesian learning for structure search we are posed with the following problem. There is a large number of possible structures, but we only know the energies of a few of these structures. Hence, we would like to predict the energies of the remaining structures. Instead of the energy we could of course consider any other parameter of interest. Since the parameter we wish to learn has no influence on the general idea of Gaussian process regression we will stick with energy.

We use vector $\mathbf{x} = (x_1, x_2, ..., x_n)$ to represent the energies of each structure. We already know a subset of energies $\mathbf{x_2} = (x_{p+1}, x_{p+2}, ..., x_n)$ and wish to predict the remaining energies $\mathbf{x_1} = (x_1, x_2, ..., x_p)$. Additionally, we also have prior knowledge regarding the structures - say an educated guess for the energies and some insight on how the energy of one structure is connected to that of another one. Simply put, slightly different structures should have slightly different energies, etc.. We can represent this prior knowledge by a mean vector $\boldsymbol{\mu}$ and a covariance matrix C. On the assumption that our data vector is normally distributed around the mean vector, we can represent our problem using a multivariate Gaussian.

While the mean vector $\boldsymbol{\mu}$ directly represents our guess-energies, the role of the covariance matrix is not as straightforward. The covariance matrix should represent the correlation between the energies of different structures. It is reasonable to assume, that similar structures have similar energies. Hence, their correlation should be high. Vice versa, dissimilar structures should have barely correlated energies. So the task at hand is to find a function, that fulfills the aforementioned requirement, while also providing a mathematically valid (symmetric and positive semi-definite) covariance matrix. Such functions are provided by

kernel functions. Supposing that every structure x_i possesses a feature vector ξ_i, we can express the elements of the covariance matrix as follows.

$$C_{ij} \;=\; k(\xi_i, \xi_j) \tag{6}$$

There exists a large number of different kernel functions, such as the uniform, the triangular, the quadratic or the squared exponential kernel. The suitability of a specific kernel function depends on the problem. For reasons of brevity, we content ourselves with presenting the squared exponential kernel.

$$k_{SE}(\xi_i, \xi_j) \;=\; \sigma^2 \;\exp\left(-\frac{1}{2\tau^2}||\xi_i - \xi_j||^2\right) \tag{7}$$

The bandwidth parameter τ controls the strength of the correlation between two points ξ_i and ξ_j. A large τ means large correlation length and vice versa. The variance σ^2 determines the average deviation of the energies x_i from their mean.

Let us now return to the multivariate Gaussian. We suppose, that the vector $\mathbf{x}$ consists of two stacked vectors. Vector $\mathbf{x_2}$ refers to energies we have calculated, while $\mathbf{x_1}$ contains the energies, we want to predict. In this case, the mean vector consist of $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$ and the covariance matrix becomes a block-matrix.

$$\mathbf{x} \;=\; \begin{bmatrix}\mathbf{x_1}\\ \mathbf{x_2}\end{bmatrix} \qquad \boldsymbol{\mu} \;=\; \begin{bmatrix}\boldsymbol{\mu_1}\\ \boldsymbol{\mu_2}\end{bmatrix} \qquad C \;=\; \begin{pmatrix}C_{11} & C_{12}\\ C_{21} & C_{22}\end{pmatrix}$$

Matrix C is symmetric, hence $C_{12} = C_{21}^T$. Using these expressions, we can rewrite equation 4. Hereby, we omit the normalization, since it is a constant and does not depend on $\mathbf{x_1}$ or $\mathbf{x_2}$.

$$p(\mathbf{x_1}, \mathbf{x_2}|\boldsymbol{\mu}, C) \;\propto\; \exp\left[-\frac{1}{2}\begin{pmatrix}\mathbf{x_1} - \boldsymbol{\mu_1}\\ \mathbf{x_2} - \boldsymbol{\mu_2}\end{pmatrix}^T \begin{pmatrix}C_{11} & C_{12}\\ C_{21} & C_{22}\end{pmatrix}^{-1} \begin{pmatrix}\mathbf{x_1} - \boldsymbol{\mu_1}\\ \mathbf{x_2} - \boldsymbol{\mu_2}\end{pmatrix}\right] \tag{8}$$

With some effort we can convert expression 8 into expression 9. The math of the issue is presented in appendix A.

$$p(\mathbf{x_1}, \mathbf{x_2}|\boldsymbol{\mu}, C) \;\propto\; \exp\left[-\frac{1}{2}(\mathbf{x_1} - \bar{\boldsymbol{\mu}})^T A(\mathbf{x_1} - \bar{\boldsymbol{\mu}})\right] \cdot \exp\left[-\frac{1}{2}(\mathbf{x_2} - \boldsymbol{\mu_2})^T C_{22}^{-1}(\mathbf{x_2} - \boldsymbol{\mu_2})\right] \tag{9}$$

Here matrix $A \;=\; (C_{11} - C_{12}C_{22}^{-1}C_{21})^{-1}$ stems from inverting the covariance matrix C. We can interpret the first exponential as a conditional probability $p(\mathbf{x_1}|\mathbf{x_2})$ and the second exponential $p(\mathbf{x_2})$.

$$p(\mathbf{x_1}, \mathbf{x_2}|\mu, C) \;=\; p(\mathbf{x_1}|\mathbf{x_2}) \cdot p(\mathbf{x_2})$$

Rewriting this equation gives us the following expression. Note the similarity to the Bayes' theorem (equation 2).

$$p(\mathbf{x_1}|\mathbf{x_2}) \;=\; \frac{p(\mathbf{x_1}, \mathbf{x_2}|\mu, C)}{p(\mathbf{x_2})}$$

Hence, we find that the conditional probability for a vector $\mathbf{x_1}$, given a vector $\mathbf{x_2}$ can be expressed as a multivariate Gaussian with a mean vector $\bar{\boldsymbol{\mu}}$ and a covariance matrix A.

$$p(\mathbf{x_1}|\mathbf{x_2}) \;\propto\; \exp\left[-\frac{1}{2}(\mathbf{x_1}-\bar{\boldsymbol{\mu}})^T A(\mathbf{x_1}-\bar{\boldsymbol{\mu}})\right] \tag{10}$$

This expression allows to compute the probability of a vector of guess-energies $\mathbf{x_1}$. Furthermore we can determine the *best* guess-energy vector by calculating the expectation value. For a Gaussian, the expectation value is given by the mean vector $\bar{\boldsymbol{\mu}}$. The mean vector $\bar{\boldsymbol{\mu}}$ has the following expression:

$$\bar{\boldsymbol{\mu}} \;=\; \boldsymbol{\mu_1} + C_{12}C_{22}^{-1}(\mathbf{x_2}-\boldsymbol{\mu_2}) \tag{11}$$

The covariance matrix A can be expressed in terms of the original covariance matrix C.

$$A \;=\; (C_{11} - C_{12}C_{22}^{-1}C_{21})^{-1} \tag{12}$$

Predicting the unknown energy vector $\mathbf{x_1}$ requires a prior guess for the known and unknown energies $\boldsymbol{\mu_2}$ and $\boldsymbol{\mu_1}$, the known energies $\mathbf{x_2}$ themselves and a covariance matrix C. To determine the covariance matrix, we represent each structure with a feature vector and calculate the covariance matrix elements using kernel functions. Then we simply evaluate equation 11 and we are done - Huzzah!

1.2 Molecular Polymorphism

The following chapter is mostly based on the book *Polymorphism in Molecular Crystals*[6] by Joel Bernstein as well as the paper *Substrate-induced and Thin-Film Phases: Polymorphism of Organic Materials on Surfaces*[7] by Andrew O. F. Jones et al.

The term polymorphism stems from the Greek *poly* = many and *morph* = from and is used to describe the diversity of nature in many disciplines. In solid state physics, polymorphism describes the ability of a solid to crystallize in more than one crystal structure.

When studying the literature, one will encounter a number of definitions related to the topic of polymorphism. Although the use of these terms is somewhat debated, we will briefly mention some of the most prevalent ones:

- **Packing polymorphism**
  polymorphism, that results from differences in crystal packing (type of polymorphism considered in this thesis)
- **Conformational polymorphism**
  polymorphism, that results from different conformers (different molecular conformation due to rotation around a single bond)
- **Pseudopolymorphism**
  most commonly used to describe the structure of solvates or hydrates of a material[2]

A further distinction can be made between monotropic and enantiotropic polymorphic systems (see figure 2). In a monotropic system one finds only a single thermodynamically stable polymorph below the melting point. In an enantiotropic system the stability ranking between different polymorphs depends on the temperature. Hence, in enantiotropic systems, it is possible to induce phase-transitions by changing the temperature.

1.2.1 Reasons for Polymorphism

The formation of molecular crystals is governed by close packing[3] in combination with maximizing favorable non-covalent interactions within and between molecules, such as hydrogen bonds, halogen bonds, π-π interactions and van der Waals interactions. Polymorphs may form, if multiple configurations of the molecules sufficiently fulfill close packing and also result in favorable interactions that are close in energy. The energy difference between such polymorphs is often smaller than $20meV$[8][9].

[2]Pseudopolymorphism as used here, describes a wide range of behaviors found in solvates that exist between two extremes. On the one hand the solvent may be strongly bound and is in many cases an integral part of the original crystal structure, which would collapse if the solvent was removed. On the other hand the solvant may be loosely bound, such that removing it will not lead to a collapse of the original crystal structure[6].

[3]The close packing principle is valid for molecules with non-directional bonds, such as van der Waals interactions. In this case maximizing the packing density maximizes the interaction energy.

1.2.2 The Phase Rule

Gibbs' phase rule describes the number of different coexisting phases P of a material in thermodynamic equilibrium. It is given by the following equation:

$$F = C - P + 2 \tag{13}$$

Here F is the number of degrees of freedom, C is the number of components in the system and P is the number of phases, that are in thermodynamic equilibrium with each other. According to Glasstone 1940 and Findely 1951 a phase is defined as *any homogeneous and physically distinct part of a system which is separate from other parts of the system by definite bounding surfaces.* Any substance therefore has one gaseous and one liquid phase. In the solid form, each crystalline structure constitutes a phase.

The number of components C is the number of different chemical species, such as molecules, atoms, ions, etc. in all phases of the system.

The number of degrees of freedom F is the number of variables, like temperature, pressure or species concentration, that can be varied independently from each other.

1.2.3 Thermodynamics of Polymorphism

The thermodynamic stability of these polymorphs is determined by the Gibbs free energy (see chapter 1.4), whereby the most stable polymorph has the lowest Gibbs free energy. The Gibbs free energy is given by equation 14. Here U is the internal energy, T is the temperature, S is the entropy and pV is the mechanical work.

$$G = U + pV - TS \tag{14}$$

Let us consider a simple system with two polymorphs and take a closer look at the temperature dependence of the Gibbs free energy. We can plot the G versus the temperature for each polymorph. Such a plot is shown in figure 2.

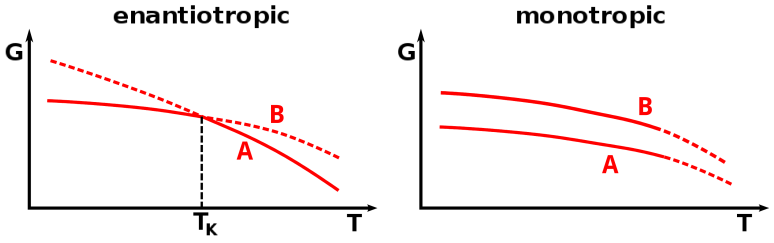


Figure 2: Exemplary temperature dependence of the Gibbs free energy for an enantiotropic and a monotropic system of two polymorphs A and B

As shown in figure 2, in the enantiotropic system polymorph B has a lower Gibbs free energy at lower temperatures. At a critical temperature T_K the Gibbs free energy of polymorph A becomes the lower one and a phase transition occurs. In the monotropic system T_K lies above the melting point. Hence, in the solid phase polymorph A always has the lowest Gibbs free energy and is therefore the most stable polymorph.

1.2.4 Kinetic Factors in Polymorphism

Thermodynamics tells only half of the story. Kinetics can also have an influence on polymorph formation. Figure 4 shows the reaction coordinate of a supersaturated fluid with a free energy G_0, that can crystallize into two polymorphs. For the crystallization the system must overcome a reaction barrier G_i^*. In this case kinetics favors polymorph 2, due to the lower reaction barrier, and thermodynamics favors polymorph 1, which has the lower Gibbs free energy.

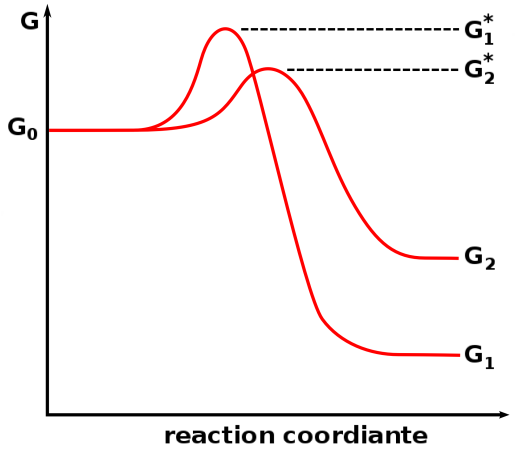


Figure 3: Schematic for crystallization of a dimorphic system; to form polymorph 1 a higher reaction barrier has to be overcome than for polymorph 2

The above described competition between thermodynamic and kinetic processes is formulated in Ostwald's rule of stages: a system seeks to crystallize in the *nearest* metastable structures, rather than the most stable one. It can then transition to more thermodynamically stable polymorphs[10]. This requires overcoming a reaction barrier. Thus it is possible, that metastable phases become kinetically trapped during crystallization.

1.2.5 Substrate-Induced Phases

Molecules which crystallize on solid substrates, often form unique polymorphs near the interface that are not found in their bulk phase[7]. In the scientific literature a number of terms such as thin-film phases, surface-mediated phases and substrate-induced phases are used to describe these polymorphs. Hereafter we will use the term *substrate-induced phases*.

Substrate-induced phases extend over more than one adsorbate layer and can therefore differ from self assembled monolayers. Further substrate-induced phases may or may not show epitaxial growth, since they are defined as a divergence from the bulk structure in the vicinity of an interface.

The formation of a substrate-induced phase depends on thermodynamic and kinetic factors. These factors result from the concurrence of intra/intermolecular and molecule-substrate interaction and in case of solution deposition also from molecule-solvent interactions. To illustrate the complexity of substrate-induced phases, we will look at the examples of pentacene.

The structure and morphology of pentacene thin films strongly depend on the growth conditions and the substrate material. On graphene or metals like copper or gold, the molecules lie flat on the surface[11][12][13], while on polymer surfaces or silica the molecules arrange slightly tilted to the vertical orientation[14][15][16][17]. The latter phase has been identified as a substrate-induced phase[18]. In submonolayer and monolayer coverage pentacene forms a vertically standing structure[19][20], which however does not constitute a substrate-induced phase, since it does not extend over multiple layers. In the bulk two polymorphs, the Campbell phase and the low temperature or LT phase, have been observed.

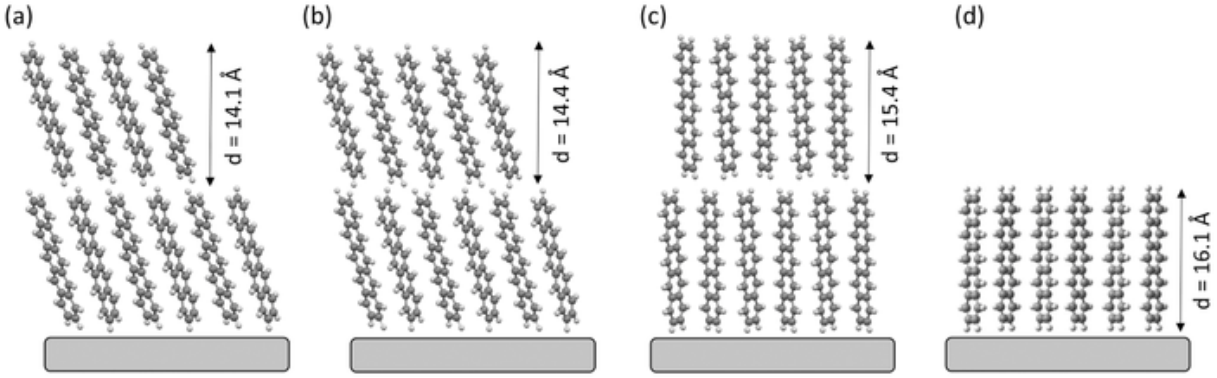


Figure 4: Different packing structures of pentacene: a) bulk (LT) phase, b) bulk (HT) phase, c) substrate-induced phase, d) monolayer, taken from Andrew O. F. Jones et al [7]

1.2.6 Types of Epitaxy

In this overview of polymorphism we have so far considered bulk and thin-film structures. Now we will take a closer look at adsorbate monolayers on substrates and discuss the classification of such monolayers in terms of geometry. The unit cell of any periodic adsorbate layer on a substrate may be described in terms of the substrate-lattice vectors with a linear transformation.

$$\begin{pmatrix} \mathbf{l_1} \\ \mathbf{l_2} \end{pmatrix} = \begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{v_1} \\ \mathbf{v_2} \end{pmatrix} \tag{15}$$

Here $\mathbf{l_1}$ and $\mathbf{l_2}$ are the lattice vectors of the adsorbate layer and $\mathbf{v_1}$ and $\mathbf{v_2}$ are the primitive lattice vectors of the substrate. The transformation matrix is called the epitaxy matrix. By looking a the elements of the epitaxy matrix, it is possible to define four types of epitaxy [21].

- **Commensurability**
  A structure is called commensurate, if all elements of the epitaxy matrix are integers. In this case the adsorbate unit cell is a substrate super cell and every adsorbate lattice point coincides with the substrate lattice.
- **Coincidence I** or **Point-on-Line Coincidence**
  The adsorbate lattice lies on lines corresponding to the primitive substrate lattice. Here one column of the epitaxy matrix contains integer elements.
- **Coincidence II**
  Only a fraction of the adsorbate lattice points coincide with the substrate, i.e. only the points of an adsorbate lattice super cell coincide. In case of coincidence II all elements of the epitaxy matrix are rational numbers.
- **Incommensurability**
  The adsorbate lattice does not coincide with the substrate lattice or lattice lines thereof. The elements of the epitaxy matrix are real numbers.

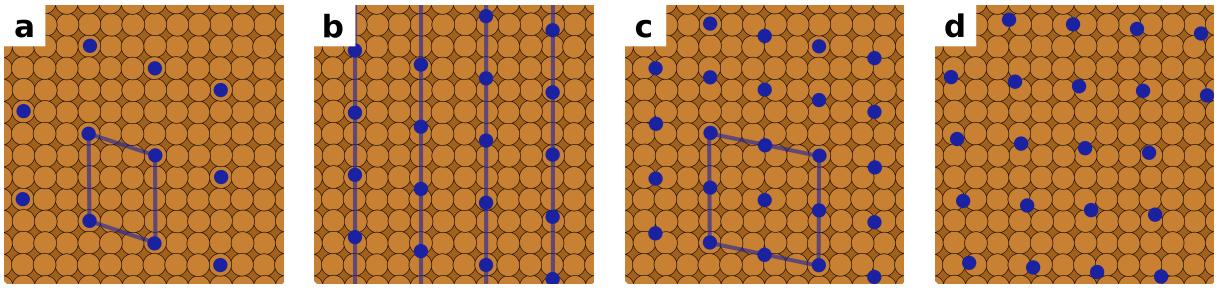


Figure 5: Types of epitaxy: a) commensurability, b) point-on-line coincidence, c) coincidence II, d) incommensurability

1.3 Density Functional Theory

1.3.1 Many-Particle Problem

The term *quantum mechanical many-particle problem* refers to a variety of physical questions, concerning the microscopic properties of systems with a large number of particles. A crystal contains in the order of 10^{23} atoms and thus constitutes a many-particle problem. To describe such a system in a quantum mechanical framework, we use the Schrödinger equation. Equation 16 shows the time-independent, non-relativistic Schrödinger equation for a many-particle problem.

$$\hat{H} \cdot \psi(\mathbf{r}_1, ..., \mathbf{r}_N; \mathbf{R}_1, ..., \mathbf{R}_M) \;=\; E \cdot \psi(\mathbf{r}_1, ..., \mathbf{r}_N, \mathbf{R}_1, ..., \mathbf{R}_M) \tag{16}$$

Here $\hat{H}$ is the Hamiltonian with the eigenvalue E. ϕ is the wave function, which depends on the coordinates of electrons $\mathbf{r}_i$ and nuclei $\mathbf{R}_\alpha$. The Hamiltonian is shown in equation (17).

$$\hat{H} \;=\; \hat{T}_e \;+\; \hat{T}_N \;+\; \hat{V}_{eN} \;+\; \hat{V}_{ee} \;+\; \hat{V}_{NN} \tag{17}$$

T_e and T_N denote the kinetic energy of electrons and nuclei, V_{eN} is the nuclei-electron interaction, V_{ee} is the electron-electron interaction and V_{NN} denotes the nuclei-nuclei interaction. The individual components are written as follows.

$$\begin{aligned}
\hat{T}_e &= -\sum_{i=1}^{N} \frac{\hbar^2}{2m_e} \nabla_i^2 \\
\hat{T}_N &= -\sum_{\alpha=1}^{M} \frac{\hbar^2}{2M_\alpha} \nabla_\alpha^2 \\
\hat{V}_{eN} &= -\sum_{i=1}^{N} \sum_{\alpha=1}^{M} \frac{Z_\alpha e^2}{|\mathbf{r}_i - \mathbf{R}_\alpha|} \\
\hat{V}_{ee} &= \sum_{i=1}^{N} \sum_{j>i}^{N} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} \\
\hat{V}_{NN} &= \sum_{\alpha=1}^{M} \sum_{\beta>\alpha}^{M} \frac{Z_\alpha Z_\beta e^2}{|\mathbf{R}_\alpha - \mathbf{R}_\beta|}
\end{aligned}$$

While the Schrödinger equation can be solved analytically for some extremely simple problems, such as a single hydrogen atom, more complex systems can only be treated numerically. Considering the aforementioned multitude of particles and the resulting large number of degrees of freedom, one can easily appreciate that solving the many-particle problem in full exceeds the capability of existing computer technology. Hence a number of approximations are employed.

Let us first consider the Born-Oppenheimer approximation[22]. It allows to decouple the motion of electrons and nuclei and therefore to separate the Schrödinger equation into an equation for electrons and nuclei respectively. We can intuitively understand why this is justified in most cases by considering the large difference in mass between electrons

and nuclei. A proton weighs about 1836 times more than an electron. The particles mainly interact through Coulomb interactions. Electrons and protons have the same (but opposite) charge and will therefore *feel* Coulomb forces of the same magnitude. Due to the significantly higher inertia nuclei will therefore move much slower than electrons and hence can be considered fixed when looking at the electrons. The wave-function can be expressed as a product of an electron $\phi(\mathbf{r};\mathbf{R})$ and a nuclei wave-function $\eta(\mathbf{R})$.

$$\psi(\mathbf{r};\mathbf{R}) = \phi(\mathbf{r};\mathbf{R}) \cdot \eta(\mathbf{R}) \tag{18}$$

The electron wave-function $\phi(\mathbf{r},\mathbf{R})$ depends on the positions $\mathbf{r}$ of the electrons, while the positions of the nuclei $\mathbf{R}$ are considered as fixed parameters. For the electron wave-function, we find an electronic Hamiltonian, shown in equation (19). The Coulomb potential of the nuclei V_{NN} becomes a constant $C(\mathbf{R})$.

$$\hat{H} \;=\; \hat{T}_e \;+\; \hat{V}_{eN} \;+\; \hat{V}_{ee} \;+\; C \tag{19}$$

Although the Born-Oppenheimer approximation simplifies the problem, it remains difficult to solve the Schrödinger equation. To illustrate the challenge we can consider, that the electron-electron interaction still contributes $\frac{(N-1)(N-2)}{2}$ terms, that lead to a large system of coupled differential equations. To compute the many-particle problem efficiently, these differential equations must be decoupled. There are several methods to achieve this simplification, such as Hartree-Fock theory or density functional theory (DFT).

1.3.2 Density Functional Theory

Density functional theory (DFT) is a method to determine the non-degenerate ground state of a system of non-interacting electrons in an effective potential. It is based on the Hohenberg-Kohn theorem [23]. The theorem states, that there exists only one electron ground state density for a given external potential of a system. We can prove this theorem by reductio ad absurdum.

Let us assume that a system is described by the Hamiltonian $H^I = F + U^I$, where F is the electronic Hamiltonian and U^I is an external potential. We find the ground state ψ_0^1 and the ground state electron density $n_0(\mathbf{r}) = \langle\psi_0^I|n|\psi_0^I\rangle$. Using this expression, we can calculate the ground state energy E_0^I.

$$E_0^I \;=\; \langle\psi_0^I|H^I|\psi_0^I\rangle \;=\; \langle\psi_0^I|F|\psi_0^I\rangle + \int U^I(\mathbf{r})n_0(\mathbf{r})\; dV \tag{20}$$

Now we propose a second Hamiltonian $H^{II} = F + U^{II}$ with $U^I \neq U^{II}$. The corresponding ground state ψ_0^{II} should give the same ground state electron density $n_0(\mathbf{r}) = \langle\psi_0^{II}|n|\psi_0^{II}\rangle$ as before. To prove the theorem, we must disprove this proposition.

We employ the Rayleigh-Ritz principle, which states, that the ground state energy $E_0 \leq \langle\psi|H|\psi\rangle$ is smaller or equal to the expectation value of the state $|\psi\rangle$. This allows us to formulate two inequalities.

$$E_0^I \;<\; \langle\psi_0^{II}|H^I|\psi_0^{II}\rangle \;=\; E_0^{II} + \int (\,U^I(\mathbf{r}) - U^{II}(\mathbf{r}))n_0(\mathbf{r})\; dV \tag{21}$$

$$E_0^{II} \;<\; E_0^I + \int (\,U^{II}(\mathbf{r}) - U^I(\mathbf{r}))n_0(\mathbf{r})\; dV \tag{22}$$

We add the two inequalities 21 and 22 together and find a contradiction.

$$E_0^I + E_0^{II} \;<\; E_0^I + E_0^{II} \tag{23}$$

Hence, the ground state electron density uniquely determines the external potential up to a constant. Safe in the knowledge, that the Hohenberg-Kohn theorem is correct, we will now briefly discuss DFT.

DFT serves to find the electron density $n(\mathbf{r})$, which in turn allows to determine all other ground state properties of a system. By using an effective potential V_{eff}, we can express the many-electron problem, given by equation 18, as a one-electron problem. This one-electron problem consists of N Kohn-Sham equations[24], which are one-electron Schrödinger-like equations with an effective potential V_{eff}. Their solutions are N one-electron wave-functions φ_j, referred to as the Kohn-Sham orbitals. The Kohn-Sham orbitals are decoupled. Hence, this approach is more efficient than solving the Schrödinger equation for N electrons simultaneously.

$$\left(-\frac{\hbar^2}{2m_e}\nabla^2 + V_{\mathit{eff}}(\mathbf{r}) - E_j\right)\varphi_j(\mathbf{r}) \;=\; 0 \tag{24}$$

Obtaining a solution for the Kohn-Sham functions φ_j, allow us to determine the electron density $n(\mathbf{r})$ as follows.

$$n(\mathbf{r}) \;=\; \sum_j^N |\varphi_j(\mathbf{r})|^2 \,. \tag{25}$$

In turn the electron density allows to calculate the effective potential.

$$V_{\mathit{eff}}[n(\mathbf{r})] \;=\; V_{\mathit{ext}}(\mathbf{r}) + e^2 \int \frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\; dV' + V_{XC}[n(\mathbf{r})] \tag{26}$$

Here $V_{\mathit{ext}}(\mathbf{r})$ is the external potential, which results from the Coulomb interaction between electrons and nuclei. The second term is called the Hartree term and accounts for the electrostatic interaction of electrons with each other. $V_{XC}(\mathbf{r})$ is the exchange correlation potential, which contains the quantum mechanical interactions of the particles in the system. The external potential and the Hartree term are of a classical nature and can be calculated exactly. However the exact expression of the exchange-correlation potential is not known. Therefore a number of approximations have been developed - the arguably most prominent of these approximations are discussed in the following.

- **Local-density approximation (LDA)**
  The local-density approximation (LDA) is based on the assumption that $V_{XC}(\mathbf{r})$ depends on $n(\mathbf{r})$. $\epsilon_{XC}(n)$ is the exchange correlation energy density of the homogeneous electron gas.

$$E_{XC}^{LDA}[n] \;=\; \int \epsilon_{XC}[n(\mathbf{r})]\; n(\mathbf{r})\; dV \tag{27}$$

- **Local spin-density approximation (LSDA)**
  A generalization of the LDA approximation is the local spin-density approximation (LSDA), where the spin is included.

$$E_{XC}^{LSDA}[n_\uparrow(\mathbf{r}), n_\downarrow(\mathbf{r})] \;=\; \int \epsilon_{XC}[n_\uparrow, n_\downarrow]\; n(\mathbf{r})\; dV \tag{28}$$

- **Generalized gradient approximation (GGA)**
  The generalized gradient approximation (GGA) also takes the gradient of the density into account.

$$E_{XC}^{GGA}[n_\uparrow, n_\downarrow] \;=\; \int \epsilon_{XC}[n_\uparrow(\mathbf{r}), n_\downarrow(\mathbf{r}), \nabla n_\uparrow(\mathbf{r}), \nabla n_\downarrow(\mathbf{r})]\; n(\mathbf{r})\; dV \tag{29}$$

- **Hybrid functionals**
  Hybrid functionals mix the Hartree-Fock exact exchange (equation 30) with any number of different exchange-correlation functionals.

$$E_X^{HF}[n_\uparrow, n_\downarrow] \;=\; -\frac{1}{2}\sum_{i,j}^{N}\int \varphi_i^*(\mathbf{r}_1)\varphi_j^*(\mathbf{r}_1)\frac{1}{r_{12}}\varphi_i(\mathbf{r}_2)\varphi_j(\mathbf{r}_2)\,dV_1\,dV_2 \tag{30}$$

  For example, the B3LYP functional[25][26] is constructed from the exchange functionals E_X^{LDA}, E_X^{GGA} and E_X^{HF}, as well as the correlation functionals E_C^{LDA} and E_C^{GGA}. The terms a_i are constant parameters.

$$\begin{aligned} E_{XC}^{B3LYP} \;=\; & E_X^{LDA} \;+\; a_0(E_X^{HF} - E_X^{LDA}) \;+\; a_X(E_X^{GGA} - E_X^{LDA}) \\ & +\; E_C^{LDA} \;+\; a_C(E_C^{GGA} - E_C^{LDA}) \end{aligned}$$

As discussed above, the effective potential V_{eff} depends on the electron density $n(\mathbf{r})$. The electron density, however, is calculated from the Kohn-Sham functions which must first be determined by the Kohn-Sham equations. Therefore the Kohn-Sham equations are solved iteratively, starting with an initial guess for the electron density. The subsequent solutions for the electron density are plugged back into the Kohn-Sham equations to find a new electron density. The iteration is continued until the difference between two consecutive electron densities meets a predefined convergence criterion. Further, it is also possible to use other properties of the system, such as the total energy, to define a convergence criterion.

We should also note, that in DFT the wave functions φ_j and the eigenenergy E_j are merely approximate values to determine the electron density. Strictly these values have no physical significance, but can be used as an approximation to calculate the ground state properties.

1.4 Ab Initio Thermodynamics

The following chapter is mainly based on the book-chapter *Ab Initio Atomistic Thermodynamics for Surfaces: A Primer*[27] by Jutta Rogal and Karsten Reuter from the book *Experiment, Modeling and Simulation of Gas-Surface Interactions for Reactive Flows in Hypersonic Flights*.

The physical quantities we determine with electronic structure calculations, such as DFT, are in a strict sense only valid for a temperature of $T = 0\ K$ and a pressure of $p = 0\ Pa$. For structure search however, the influence of temperature and pressure is highly important, since these two variables strongly affect the stability of polymorphs. Hence it would be worthwhile to find a way to account for finite temperatures and pressures - ab initio thermodynamics provides such an approach. The key quantity to consider is the Gibbs free energy.

$$G(T,p) \;=\; E_{tot} + F_{vib} - TS_{conf} + pV \tag{31}$$

The first term E_{tot} is the total energy of the system as determined by electronic structure calculations. F_{vib} denotes the vibration enthalpy, T the temperature and S_{conf} the configuration entropy. Finally, p is the pressure and V is the volume of the system.

The above equation describes a general case for a (T,p)-ensemble. Since this thesis mainly discusses surface structure prediction, we will focus on surfaces for the remaining discussion.

1.4.1 Assumptions

We consider a (T,p)-ensemble that consists of a layer of adsorbate molecules, which forms on a perfect substrate. All adsorbate layers are assumed to have full monolayer coverage. Further we suppose that the monolayer has been grown in thermodynamic equilibrium. This can be experimentally realized by techniques such as hot wall epitaxy.

1.4.2 Gibbs Free Energy of Adsorption

The aim of this chapter is to find an expression for the Gibbs free energy of adsorption, since this is the key quantity one needs to consider when searching surface structures at finite temperature and pressure. In thermodynamic equilibrium, the internal energy of a single component bulk system is given by:

$$E_{bulk} \;=\; TS - pV + N\mu \tag{32}$$

If we create a surface, for instance by cutting the bulk in half, we have to consider the additional energy contribution from the surface. This contribution is proportional to the surface area A, whereby the proportionality constant is given by the surface energy γ. Using the surface energy, the internal energy can be written as follows:

$$E_{surf} \;=\; TS - pV + N\mu + \gamma A \tag{33}$$

By introducing the Gibbs free energy $G = E - TS + pV$, we can rewrite equation 33. We also introduce an index, indicating that γ_M is the surface energy of the clean metal substrate.

$$\gamma_M = -\frac{1}{A}(G_M - N_M\mu_M) \tag{34}$$

The above equation can be easily extended to a multi-component system, that is in thermodynamic equilibrium with a surrounding gas or liquid phase.

$$\gamma_{M,Ads} = -\frac{1}{A}(G_{M,Ads} - \sum_i N_i\mu_i) \tag{35}$$

In case of a two component system, such as metal substrate thermodynamic equilibrium with a surrounding gas of adsorbate molecules, we can simply include the chemical potential of the adsorbate. Hereby N_M and N_{Ads} are the respective numbers of the metal atoms and adsorbate molecules in direct vicinity of the interface.

$$\gamma_{M,Ads} = -\frac{1}{A}(G_{M,Ads} - N_M\mu_M - N_{Ads}\mu_{Ads}) \tag{36}$$

Ultimately we are interested in comparing the stability of different adsorption phases. To this end we need to consider the Gibbs free energy of adsorption. For an adsorbate layer in thermodynamic equilibrium with the adsorbate in gas phase we can write:

$$\gamma = \gamma_M - \gamma_{M,Ads} = -\frac{1}{A}(\Delta G_{M,Ads} - N_{Ads}\mu_{Ads}) \tag{37}$$

If we write $\Delta G_{M,Ads}$ out in full we get the final expression for the Gibbs free energy of adsorption.

$$\gamma = \frac{1}{A}(\Delta E + pV - TS_{conf}N_{ads} + F_{vib} - \mu_{ads}N_{ads}) \tag{38}$$

In the above equation ΔE is the adsorption energy, that we directly determine with electronic structure calculations. The term pV stands for the mechanical work. We assume that this term is the same for all configurations and can therefore be left out, when comparing Gibbs free energies. In the third term T is the temperature and S_{conf} is the configuration entropy, which we will discuss later in more detail. F_{vib} denotes the vibration enthalpy, which consists of the zero point vibration energy and the vibration entropy. Finally μ_{ads} is the chemical potential of the adsorbate in gas phase and N_{ads} is the number of adsorbates in the unit cell. A derivation of the chemical potential of an ideal gas can be found in appendix C.

In thermodynamic equilibrium the systems we consider, seek to minimize the Gibbs free energy per area. Hence, the polymorph that forms at a certain temperature and pressure is the one with the lowest γ.

1.4.2.1 Configuration Entropy

In discussing the configuration entropy, we intend to gauge its contribution to the Gibbs free energy of adsorption. As we will show, it can be neglected in many cases.

The configuration entropy S_{conf} mainly results from the concentration and distribution of defects in the monolayer. In total we have N adsorption sites on the surface, whereby n of these sites are defect sites. Therefore the configuration entropy is:

$$S_{conf}^{tot} \;=\; k_B \;\ln\left(\frac{N!}{(N-n)!n!}\right) \tag{39}$$

We calculate the configuration entropy per site S_{conf}/N. Using the Stirling formula $\ln(N!) = N\ln(N) - N$ we get:

$$S_{conf} \;=\; \frac{S_{conf}^{tot}}{N} \;=\; k_B\left[\frac{n}{N}\;\ln\left(\frac{N}{n}-1\right) - \ln\left(1-\frac{n}{N}\right)\right] \tag{40}$$

In this work we consider highly ordered systems within a temperatures range of 0 K and 400 K. Therefore we can estimate the contribution of the configuration entropy when comparing configurations. Assuming a defect density of 10 % i.e. $\frac{n}{N} = 0.1$ and a temperature of 400 K, equation (40) yields a configuration entropy contribution of about 12 meV per adsorption site.

However, equation (40) depends only on the defect density. Therefore TS_{conf} will be the same for all configurations, but scales with system size and number of particles. Therefore, if we compare two configurations, the contribution of the configuration entropy to the difference in the Gibbs free energy is given by:

$$T\;S_{conf}\;\left(\frac{N_{ads,1}}{A_1} - \frac{N_{ads,2}}{A_2}\right) \tag{41}$$

Continuing with the assumptions from above, we consider two configurations with the respective coverages of $\frac{N_{ads,1}}{A_1} = \frac{1}{10}$ and $\frac{N_{ads,2}}{A_2} = \frac{1}{15}$. For this representative example the difference in configuration entropy is $TS_{conf} = 0.4\ meV$. Further it is easy to see, that two configurations with similar coverage do not differ in configuration entropy.

Considering a reasonable convergence criterion of 10 meV for the parameters (k-points, basis functions etc.) that feed into an electronic structure calculation, we can neglect the contribution of the configuration entropy when comparing Gibbs free energies.

2 Methods

2.1 Introduction to SAMPLE

In this chapter we introduce the SAMPLE approach[28][29] to structure search. SAMPLE stands for *Surface Adsorbate Polymorph Prediction with Little Effort* and allows to efficiently predict surface structures from first principle.

Figure 6: The SAMPLE logo

Unlike other structure search methods, such as basin hopping or genetic algorithms, SAMPLE is capable to consider all possible structures in a given configuration space. We accomplish this by coarse-graining the potential energy surface and using Bayesian learning to predict adsorption energies. To this aim, we first determine the adsorption energies of a selected set of structures with dispersion-corrected density functional theory. These structures serve as training set for a machine learning algorithm, which allows to predict the energies of the remaining structures. SAMPLE is based on two key assumptions:

1. All structures are commensurate.
2. All structures are in thermodynamic equilibrium.

The first assumption results from using periodic boundary condition DFT (density functional theory). With the assumption of commensurability, we also implicitly assume that the interactions of the adsorbate molecules with the substrate are dominant compared to the interactions between the molecules.

The workflow of SAMPLE comprises five steps, which will be discussed in more detail in the following chapters.

1. **Finding local adsorption geometries**
   We determine the local adsorption geometries of an isolated adsorbate molecule on the substrate.
2. **Generating all unit cells**
   We generate all unit cells, which are super cells of the substrate.
3. **Combining geometries and unit cells**
   We combine all local adsorption geometries with every unit cell to generate structures.
4. **Bayesian learning**
   We employ Bayesian learning to predict the energies of all structures.
5. **Predicting phase diagrams**
   We use ab initio thermodynamics to calculate the Gibbs free energy of every structure and to predict the concurrent phase diagram.

2.2 Finding Local Adsorption Geometries

We use the term "local adsorption geometry" to describe the way an isolated molecule adsorbs on the substrate. Hence, a local adsorption geometry represents a local minimum of the potential energy surface of a single molecule on the substrate.

We note that the local adsorption geometries are invariant under any symmetry operation of the substrate, like translation by a primitive lattice vector or reflection by mirror axis. This fact will be of great utility for building possible structures (see chapter 2.4).

To find these local adsorption geometries, we define a number of starting geometries that serve as starting points for local geometry optimizations. These optimization calculations then converge to a small number of distinct geometries, i.e. different starting geometries converge to the same final geometry.

To perform this task efficiently, we utilize the substrate symmetries to choose the set of starting geometries. First, we determine the high symmetry points on the surface: These points then serve as focal points for the molecules' center of mass. In figure 7, the top, bridge and the hollow positions of a fcc(100) surface are displayed.

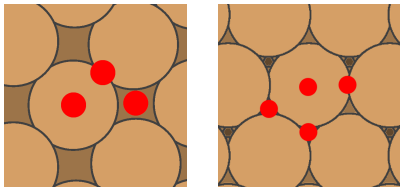


Figure 7: High symmetry points on different substrate surfaces
Left: top, bridge, hollow positions of a fcc(100) surface
Right: top, bridge, hcp/fcc hollow positions of a fcc(111) surface

Secondly, we consider the rotation of the molecule around the surface normal (z-axis). Naively we could simply choose a discritization of the rotation, say 15° and rotate the molecule by 0°, 15°, ..., 330°, 345°. We can, however, reduce the number of orientations by considering the rotation and mirror symmetries of the respective high symmetry points. The fcc(111) surface, for instance, has a three-fold rotation symmetry, as well as three mirror symmetries (2D point group D_3) for the top and the two hollow positions. Hence we can limit the orientations for the molecule to a $[0°, 120°]$ angle range. For the bridge position, however, only one mirror symmetry (and no rotation symmetries) exists, which necessitates to consider all orientations.

If we also take the symmetries of the molecule into account, we can further limit the angle range. Let us, for example, look at a molecule with point group D_{2h} (in two dimensions, two-fold rotation symmetries and two mirror axis). The two-fold rotation symmetries allow to halve all the angle ranges, ie. $[0°, 60°]$ for top, hcp/fcc hollow and $[0°, 180°]$ for the bridge position. Additionally the mirror symmetries enable halving the angle ranges a second time.

Thirdly, we determine a suitable discritization of rotation and generate a set of starting geometries for the local geometry optimization. Finally, the FHI-aims quantum chemistry code[30] serves as tool of choice to perform the optimizations.

2.3 Generating all Unit Cells

The SAMPLE approach aims to consider all possible structures. In order to acquire such a complete set, we must first know all possible unit cells.

2.3.1 Constraints

In principle an infinite number of different unit cells exist on a given substrate. However, considering an infinite number of unit cells and therefore also structures would not only be impossible but unnecessary. Energetically favorable highly ordered, closed packed structures exist in a comparatively small configuration space. In fact, physical insight allows to define constraints that on the one hand limit the number of unit cells (and thereby structures) to a finite amount, and on the other hand mostly retain the prerequisite of a first principles search. These constraints are:

1. Commensurability
2. Coverage
3. Number of molecules per unit cell
4. Minimum width of the unit cell

Commensurability requires that the unit cell is a super cell of the substrate. This constraint results from using periodic boundary condition DFT to calculate adsorption energies. Constraints two and three determine the area of the unit cell. If we define coverage as number of molecules per area $\Theta = N/A$, then the area is given by $A = N/\Theta$. The fourth constraint avoids extremely elongated unit cells. It results from the simple fact, that at least one molecule needs to fit in the unit cell. The minimum width is defined as the height of the parallelogram, formed by the lattice vectors.

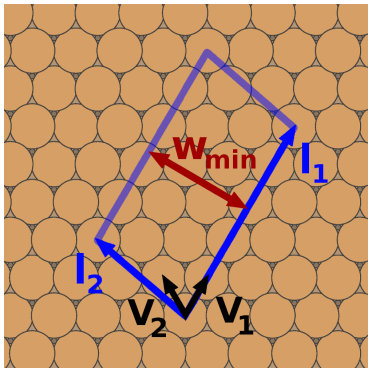


Figure 8: Parameters of a unit cell of a commensurate structure on a (111) surface: $\mathbf{l_1}$, $\mathbf{l_2}$ are the lattice vectors of the unit cell and $\mathbf{v_1}$, $\mathbf{v_2}$ are the primitive lattice vectors of the substrate. w_{min} is the minimum width of the unit cell.

With these four constraints we can limit the number of possible unit cells to a finite number.

2.3.2 Generating possible Unit Cells

SAMPLE only considers commensurate structures. Hence, the relevant unit cells are super cells of the substrate. Further, the coverage constraint, combined with the desired number of molecules, results in a predefined area for these unit cells. Fixing the area is also useful for symmetry considerations, since only unit cells of the same area can be equivalent. In case we desire unit cells for a range of coverages, we simply repeat the generation for different areas.

To start discussing the unit cell search, let us first take a look at the epitaxy matrix C. In general, we can represent any two-dimensional unit cell in fractional coordinates of the primitive substrate lattice vectors. Expressing the fractional coordinates in terms of the epitaxy matrix allows to calculate the lattice vectors of the unit cell (super-lattice vectors) as follows.

$$\begin{pmatrix} \mathbf{l_1} \\ \mathbf{l_2} \end{pmatrix} = C \cdot \begin{pmatrix} \mathbf{v_1} \\ \mathbf{v_2} \end{pmatrix} = \begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{v_1} \\ \mathbf{v_2} \end{pmatrix} \tag{42}$$

Here $\mathbf{l_1}$ and $\mathbf{l_2}$ are the two dimensional super-lattice vectors of the unit cell and $\mathbf{v_1}$ and $\mathbf{v_2}$ are the primitive lattice vectors of the substrate. C is the epitaxy matrix. For commensurate structures the elements of the epitaxy matrix are integers $n_i \in \mathbb{Z}$.

To generate unit cells, we iteratively propose different epitaxy matrices by varying the epitaxy matrix elements n_i. The following example illustrates this process. We set the second and third element of the epitaxy matrix to $n_2 = n_3 = 0$ and vary the first and fourth element within limits of $n_1 = \{1, 2, 3\}$ and $n_4 = \{1, 2, 3\}$ respectively. We find the following epitaxy matrices:

$$\begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix} = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}, \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, \dots \right\}$$

Usually, we are interested in unit cells of a certain area. The area directly relates to the epitaxy matrix via the following equation:

$$A = |n_1 n_4 - n_2 n_3| \tag{43}$$

Therefore, we can calculate the area of each proposed epitaxy matrix and only accept epitaxy matrices, that give a unit cell with the desired area.

As mentioned before, we propose epitaxy matrices by iterating over different values of the epitaxy matrix elements n_i. Sufficiently choosing the limits for this iteration, will enable us to generate all possible unit cells. Finding the limits requires some geometric considerations. The range within which we need to vary the matrix elements n_i depends on the largest super-lattice vector of a unit cell with a given area and a given minimum width. Using the area A and the minimum width w_{min} allows to calculate the maximum length ℓ_{max} as follows:

$$\ell_{max} = \frac{A}{w_{min}} \tag{44}$$

Finding all unit cells requires to determine the largest n_i, that results in ℓ_{max}. Using the primitive lattice vectors $\mathbf{v_1}$ and $\mathbf{v_2}$, as well as the epitaxy matrix elements n_1 and n_2, we can calculate the maximum super-lattice as follows:

$$\ell_{max}^2 \;=\; (\mathbf{v_1} n_1 + \mathbf{v_2} n_2)^2 \tag{45}$$

We want to find the maximum values for n_1 and n_2. To make the notation easier we define:

$$\ell_{\mathbf{1}} \;=\; \mathbf{v_1} n_1 \qquad \ell_{\mathbf{2}} \;=\; \mathbf{v_2} n_2 \qquad \cos\alpha \;=\; \tfrac{\mathbf{v_1 v_2}}{|\mathbf{v_1}||\mathbf{v_2}|}$$

The law of cosines allows to rewrite equation 45.

$$\ell_{max}^2 \;=\; \ell_1^2 + \ell_2^2 - 2\ell_1\ell_2 \;\cos\alpha \tag{46}$$

Solving this for ℓ_1 gives:

$$\ell_1(\ell_2) \;=\; \ell_2 \;\cos\alpha \pm \sqrt{\ell_2^2 \;\cos^2\alpha - \ell_2^2 + \ell_{max}^2} \tag{47}$$

To find the maximum of ℓ_1, we need to take the derivative with respect to ℓ_2 and determine the roots.

$$\frac{d\ell_1}{d\ell_2} \;=\; \cos\alpha \pm \frac{1}{\sqrt{\cdots}}\;\ell_2\;(\cos^2\alpha - 1) \;=\; 0 \tag{48}$$

We solve the above relation for ℓ_2.

$$\ell_2 \;=\; \pm\left[\frac{\ell_{max}^2\;\cos^2\alpha}{1-\cos^2\alpha}\right]^{1/2} \tag{49}$$

Substituting ℓ_2 in equation 47 results in the expression for the maxima of $\ell_1(\ell_2)$. There are two solutions for ℓ_2, a positive and a negative one. Hence, equation 47 also has two solutions. We are only interested in the maximum absolute value $\ell_{1,max}$ of $\ell_1(\ell_2)$, which leads to the following expression. Due to symmetry, the solution for $\ell_2(\ell_1)$ is the same.

$$\ell_{1,max} \;=\; |\ell_2|\;|\cos\alpha| + \sqrt{\ell_2^2\;\cos^2\alpha - \ell_2^2 + \ell_{max}^2} \tag{50}$$

Now we can calculate the maximum value that n_i can take.

$$n_1 \;=\; \frac{\ell_{1,max}}{|\mathbf{v_1}|} \tag{51}$$

$$n_2 \;=\; \frac{\ell_{1,max}}{|\mathbf{v_2}|} \tag{52}$$

$$n_{max} \;=\; max(n_1, n_2) \tag{53}$$

In order to find all unit cells, we have to vary the elements n_i of the epitaxy matrix within a range given by n_{max}:

$$n_i \in [-n_{max}, n_{max}] \tag{54}$$

Having found the required range, we can now iterate the elements of the epitaxy matrix to find new unit cells. The computational effort of this task scales with $\mathcal{O}^4$ $((2 * n_{max})^4)$, since we need to iterate over all four elements of the epitaxy matrix. However, using the area constraint reduces the number of independent epitaxy matrices elements n_i to three. The fourth n_i results automatically from equation 43. By considering three cases, it is possible to reduce the computational effort to $\mathcal{O}^3$ $((2 * n_{max})^3)$.

- $n_1 \neq 0$
  Equation 43 allows to calculate the epitaxy matrix element n_4 directly.

$$n_4 = \frac{A + n_2 n_3}{n_1}$$

- $n_1 = 0$ and $n_2 \neq 0$
  In this case, the area is $A = n_2 n_3$ and n_4 becomes a free parameter. Hence, the value of n_3 is assigned to n_4. Now equation 43 allows to determine n_3 so that the area constraint is fulfilled.

$$\begin{aligned} n_4 &= n_3 \\ n_3 &= \frac{A}{n_2} \end{aligned}$$

- $n_1 = 0$ and $n_2 = 0$
  The area is always 0.

Implementation of the above considerations allows us to generate a set of candidate unit cells. Some of these unit cells will, however, be equivalent. To find the unique unit cells among all candidates, we need to perform symmetry reduction.

2.3.3 Symmetry Reduction

In the previous chapter we discussed the methods to generate a non-unique set of unit cells. Now we will introduce different ways for unit cells to be equivalent and discuss the respective concepts used in symmetry reduction. These concepts will allow us to find a set of unique or non-equivalent unit cells.

2.3.3.1 Linear Combinations of Lattice Vectors

A two dimensional unit cell spans a grid on the surface. By connecting grid points and conserving the area, it is possible to construct equivalent unit cells.

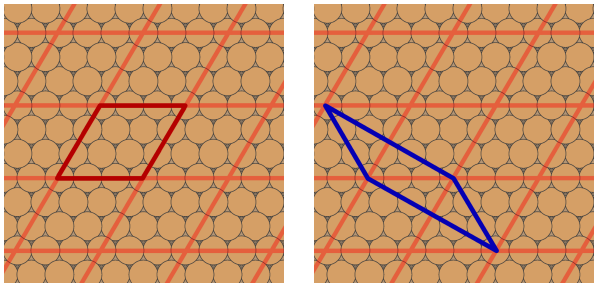


Figure 9: Example of two unit cells that are equivalent by a combination of lattice vectors

We want to compare two unit cells with the lattice vector tuples $(\mathbf{l'_1}, \mathbf{l'_2})^T$ and $(\mathbf{l_1}, \mathbf{l_2})^T$ respectively. Using the grid of the original unit cell, we can find new unit cells by connecting grid points. This is equivalent to a linear combination of lattice vectors:

$$\mathbf{l'_1} = a\mathbf{l_1} + b\mathbf{l_2} \tag{55}$$
$$\mathbf{l'_2} = c\mathbf{l_1} + d\mathbf{l_2} \tag{56}$$

If $a, b, c, d \in \mathbb{Z}$ and if the area is conserved, the first unit cell (with $(\mathbf{l_1}, \mathbf{l_2})^T$) is equivalent to the second unit cell (with $(\mathbf{l'_1}, \mathbf{l'_2})^T$). We rewrite the linear combination from above in matrix form.

$$\begin{pmatrix} \mathbf{l'_1} \\ \mathbf{l'_2} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} \mathbf{l_1} \\ \mathbf{l_2} \end{pmatrix} = M \cdot \begin{pmatrix} \mathbf{l_1} \\ \mathbf{l_2} \end{pmatrix} \tag{57}$$

For more efficient implementation, the symmetry reduction transforms the epitaxy matrix directly. Using the relation $\begin{pmatrix} \mathbf{l_1} \\ \mathbf{l_2} \end{pmatrix} = \begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{v_1} \\ \mathbf{v_2} \end{pmatrix}$ to substitute the lattice vector tuple $(\mathbf{l_1}, \mathbf{l_2})^T$ we get:

$$\begin{pmatrix} n'_1 & n'_2 \\ n'_3 & n'_4 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{v_1} \\ \mathbf{v_2} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{v_1} \\ \mathbf{v_2} \end{pmatrix} = M \cdot \begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{v_1} \\ \mathbf{v_2} \end{pmatrix} \tag{58}$$

Now we solve the above equation for the transformation matrix M.

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} n'_1 & n'_2 \\ n'_3 & n'_4 \end{pmatrix} \cdot \begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix}^{-1} \tag{59}$$

Inspecting the transformation matrix M allows us to efficiently evaluate whether two unit cells are equivalent. If all elements of M and M^{-1} are integers, the linear combination parameters are integers as well ($a, b, c, d \in \mathbb{Z}$). Additionally, if the first unit cell (with $(\mathbf{l_1}, \mathbf{l_2})^T$) and the second unit cell (with $(\mathbf{l'_1}, \mathbf{l'_2})^T$) have the same area, they are equivalent.

2.3.3.2 Substrate Symmetries

In addition to combinations of lattice vectors, unit cells can also be equivalent by substrate symmetries, i.e. if a combination of substrate symmetry transformations maps one unit cell onto another one. Figure 10 illustrates this concept.

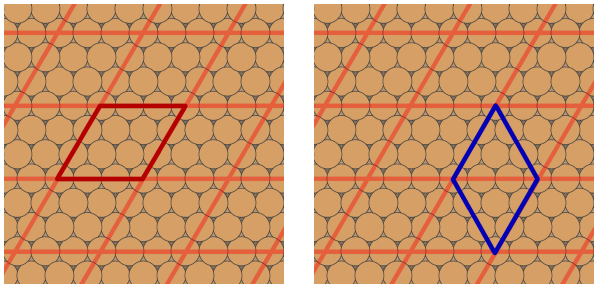


Figure 10: Example of two unit cells, that are equivalent by the 120° rotation symmetry of the (111) substrate

We can realize such a transformation by multiplying the lattice vectors $\mathbf{l_1}$ and $\mathbf{l_2}$ with a symmetry matrix R.

$$\begin{aligned} \mathbf{l_1'} &= R \cdot \mathbf{l_1} \\ \mathbf{l_2'} &= R \cdot \mathbf{l_2} \end{aligned}$$

For computational purposes, it is more convenient to use the integer representation provided by fractional coordinates. Employing the symmetry matrix in fractional coordinates R_{frac}, we can directly transform the epitaxy matrix:

$$\begin{pmatrix} n_1' & n_2' \\ n_3' & n_4' \end{pmatrix} = \begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix} \cdot R_{frac}^T \tag{60}$$

Symmetry reduction can simply be done by exhaustively applying transformations and comparing epitaxy matrices.

2.3.3.3 Sign Transformations

A further possibility for unit cells to be equivalent pertains to the signs of the epitaxy matrix elements. Flipping the sign of the following components of the epitaxy matrix results in a geometrically similar unit cell.

$$\begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix} \Leftrightarrow \begin{pmatrix} -n_1 & -n_2 \\ n_3 & n_4 \end{pmatrix} \Leftrightarrow \begin{pmatrix} n_1 & n_2 \\ -n_3 & -n_4 \end{pmatrix} \Leftrightarrow \begin{pmatrix} -n_1 & -n_2 \\ -n_3 & -n_4 \end{pmatrix} \tag{61}$$

2.3.3.4 Chirality

Two equivalent unit cells may also differ by the order of their epitaxy matrix elements. Switching the lines in the epitaxy matrix flips the sign of the area. We call this property chirality.

$$A = n_1 n_4 - n_2 n_3 \tag{62}$$

The area is nothing more than the z component of a vector product. If the epitaxy matrix has right-hand chirality, the area will be positive. A unit cell with left-hand chirality has a negative area. The transformation to right-hand chirality is done by simply flipping the epitaxy matrix.

$$\begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix} \rightarrow \begin{pmatrix} n_3 & n_4 \\ n_1 & n_2 \end{pmatrix} \tag{63}$$

2.3.3.5 Putting the Symmetry Reduction Together

Having considered the concepts used for symmetry reduction, we will now briefly discuss the workflow of unit cell symmetry reduction. As already mentioned, only unit cells with an equal area can be equivalent, which is enforced by the unit cell generator (see chapter 2.3.2). Symmetry reduction therefore requires three consecutive steps.

1. **Chirality**
   First, we enforce right hand chirality for all unit cells.
2. **Symmetries + sign transformations**
   The second step makes use of the substrate symmetries, to first iteratively transform the unit cell and secondly apply all sign transformations to each symmetry-transformed unit cell. If a resulting epitaxy matrix matches a previously found sign-unique unit cell, the iteration stops and the algorithm rejects the new unit cell. If the iteration ends without a match, the new unit cell is appended to the list of sign-unique unit cells.
3. **Symmetries + lattice vector combination**
   Similarly to the second step, the third step combines iterative symmetry transformations with lattice vector combinations. We reject a unit cell, if it fulfills the condition described in chapter 2.3.3.1 for any unique unit cell. Otherwise the unit cell is added to the list of unique unit cells.

Repeating the unit cell generation discussed in chapter 2.3.2 for a range of areas followed by symmetry reduction, allows to generate a set of all unique unit cells.

2.3.4 Unit Cell Hash

In section 2.3.3.1 we discussed that combinations of lattice vectors result in equivalent unit cells, which raises the question: What is the standard unit cell? To answer this question, we have to define a set of criteria for unit cells, whereby the unit cell that fulfills the largest number of features is considered as the standard unit cell. The epitaxy matrix of the standard unit cell is then called the unit cell hash.

2.3.5 Criteria for the Standard Unit Cell

In principle, one can find an infinite number of equivalent cells for every unit cell. By defining a set of conclusive criteria, we can however select one of these equivalent unit cells as the *standard* cell. Aside from conclusiveness, there are no other limits for these criteria. The criteria developed for the SAMPLE approach mainly enforce compact unit cells. They are defined in hierarchical order, with criteria higher up in the hierarchy trumping the ones with lower priority. This avoids conflicts between criteria.

First, we minimize the larger diagonal of the unit cell. This criterion enforces compact unit cells.

1. $min(d_{max}^2)$

Secondly, we compare elements of the epitaxy matrix. Criteria 2 thru 5 enforce that the epitaxy matrix is as close to a diagonal matrix as possible.

2. $|n_0| \geqslant |n_1|$
3. $|n_3| \geqslant |n_2|$
4. $|n_0| \geqslant |n_3|$
5. $|n_2| \geqslant |n_1|$

Thirdly, elements of the epitaxy matrix should be positive.

6. $n_0 \geqslant 0$
7. $n_1 \geqslant 0$
8. $n_2 \geqslant 0$
9. $n_3 \geqslant 0$

If two unit cells fulfill criteria 1 thru 9, we pick the unit cell with the larger first element in the epitaxy matrix.

10. $max(n_1)$

2.3.5.1 Finding the Standard Unit Cell

The algorithm to find the *standard* unit cell uses lattice vector combinations and transformations by substrate symmetries. For better handling we employ fractional coordinates and can therefore directly combine and transform epitaxy matrices.

Let us first discuss how combinations of epitaxy matrices allow us to find the *standard* unit cell. As discussed in chapter 2.3.3.1, linear combinations of epitaxy matrices produce new equivalent unit cells, if two conditions are fulfilled: First, the area must be conserved and secondly, the parameters of this linear combination must be integers. Finding the combination of epitaxy matrices that best fulfills the criteria, discussed in chapter 2.3.5, allows us to determine the *standard* unit cell. The following equation shows such a combination of epitaxy matrices.

$$\begin{pmatrix} n_1' & n_2' \\ n_3' & n_4' \end{pmatrix} = \begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix} + a \begin{pmatrix} n_1 & n_2 \\ 0 & 0 \end{pmatrix} + b \begin{pmatrix} n_3 & n_4 \\ 0 & 0 \end{pmatrix} + c \begin{pmatrix} 0 & 0 \\ n_1 & n_2 \end{pmatrix} + d \begin{pmatrix} 0 & 0 \\ n_3 & n_4 \end{pmatrix} \quad (64)$$

The parameters $a, b, c, d \in \mathbb{Z}$ denote how often a specific lattice vector is added. The parameters a and d combine a lattice vector with itself and allow the cell to flip, while b and c control the shear of the unit cell. Figure 11 illustrates the process of combining epitaxy matrices.

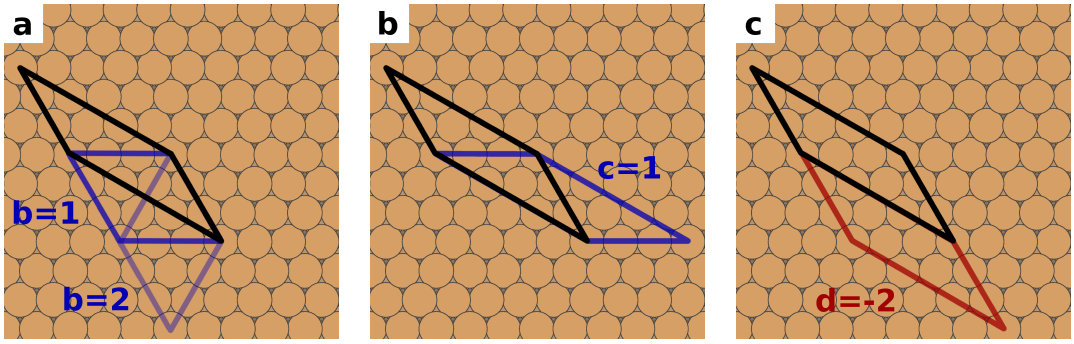


Figure 11: Combining epitaxy matrices to generate the *standard* unit cell
a) introducing shear through variation of $b \in \{0, 1, 2\}$
b) introducing shear through variation of $c \in \{0, 1\}$
c) *flipping* the unit cell with $d = -2$
(we note that if $d = -1$ and $a = b = c = 0$ results in a unit cell with area 0)

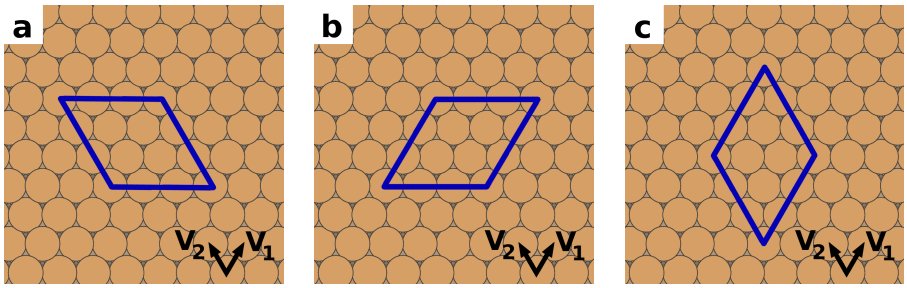
To systematically find the *standard* unit cell, we iterate over the parameters a, b, c, d and evaluate each epitaxy matrix on the basis of the criteria presented in chapter 2.3.5. This requires finding the range in which the parameters a, b, c, d need to be iterated. To determine this range, we can make use of the fact that the area of the unit cells must be conserved. Let us first consider the parameters a and d, which combine a lattice vector with itself. Doubling the length of a lattice vector ($a, d = 1$), while retaining its direction, will double the area. Apart form that, negative values of a and d are useful, whereby $a, d = -2$ is the limit (flipping the lattice vector). Hence we choose the following limits:

$$a, d \in \{-2, -1, 0\}$$

The parameters b and c control the shear of the unit cell. Here we alter the direction of the lattice vector, so the argument for a and d no longer holds true. Therefore, the latter parameters are iterated within a shell size s, which we will discuss later in more detail.

$$b, c \in [-s, s]$$

Additionally, we transform each candidate epitaxy matrix, generated through combinations of epitaxy matrices, by each symmetry transformation of the substrate. While such transformations do not change the shape of the unit cell in real space, they alter the compactness in the oblique-angled fractional coordinate space of the epitaxy matrix. Figure 12 illustrates this process.



$$\begin{pmatrix} 0 & 4 \\ -4 & 4 \end{pmatrix} \quad \begin{pmatrix} 4 & 0 \\ -4 & 4 \end{pmatrix} \quad \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$$

Figure 12: Symmetry transformations alter the compactness of the epitaxy matrix
a) epitaxy matrix with large non-diagonal elements
b) epitaxy matrix fulfills criteria better than (a)
c) diagonal epitaxy matrix - *standard* unit cell for this cell-shape

The above described process of iterating the parameters a, b, c, d and applying symmetry transformations, is wrapped in a shell-like algorithm. The algorithm varies the parameters b and c within consecutively larger shells. The shell size s serves as the limit for the parameters, so that $b, c \in [-s, s]$. In turn, the shell size takes the value $s \in [1, s_{max}]$, whereby s_{max} is the maximum shell size. The maximum shell size can be determined by calculating the shear of the current best unit cell.

$$\begin{aligned} h_1 &= \frac{|\mathbf{l_1 l_2}|}{|\mathbf{l_2}|} \quad \rightarrow \quad s_1 = ceil\left(\frac{h_1}{|\mathbf{l_2}|}\right) \\ h_2 &= \frac{|\mathbf{l_1 l_2}|}{|\mathbf{l_1}|} \quad \rightarrow \quad s_2 = ceil\left(\frac{h_2}{|\mathbf{l_1}|}\right) \\ s_{max} &= max(s_1, s_2) \end{aligned}$$

To better understand the principle of a shell-like algorithm, let us consider two iteration steps of this algorithm. In the first shell $s = 1$, and therefore we iterate the parameters b and c within the limits $b, c \in [-1, 0, 1]$. In the second shell $s = 2$, which means that $b, c \in [-2, -1, 0, 1, 2]$. Values of b and c, that have already been explored in previous shells, are skipped.

Whenever the algorithm finds an epitaxy matrix that fulfills the criteria better, the iteration stops and the process of combining lattice vectors starts anew. The advantage of the shell-like algorithm is that a better epitaxy matrix is often found in the first or second shell, which means that we don't have to look at all possible values of b and c. Figure 13 presents a flow diagram, that illustrates the algorithm to determine the *standard* unit cell.

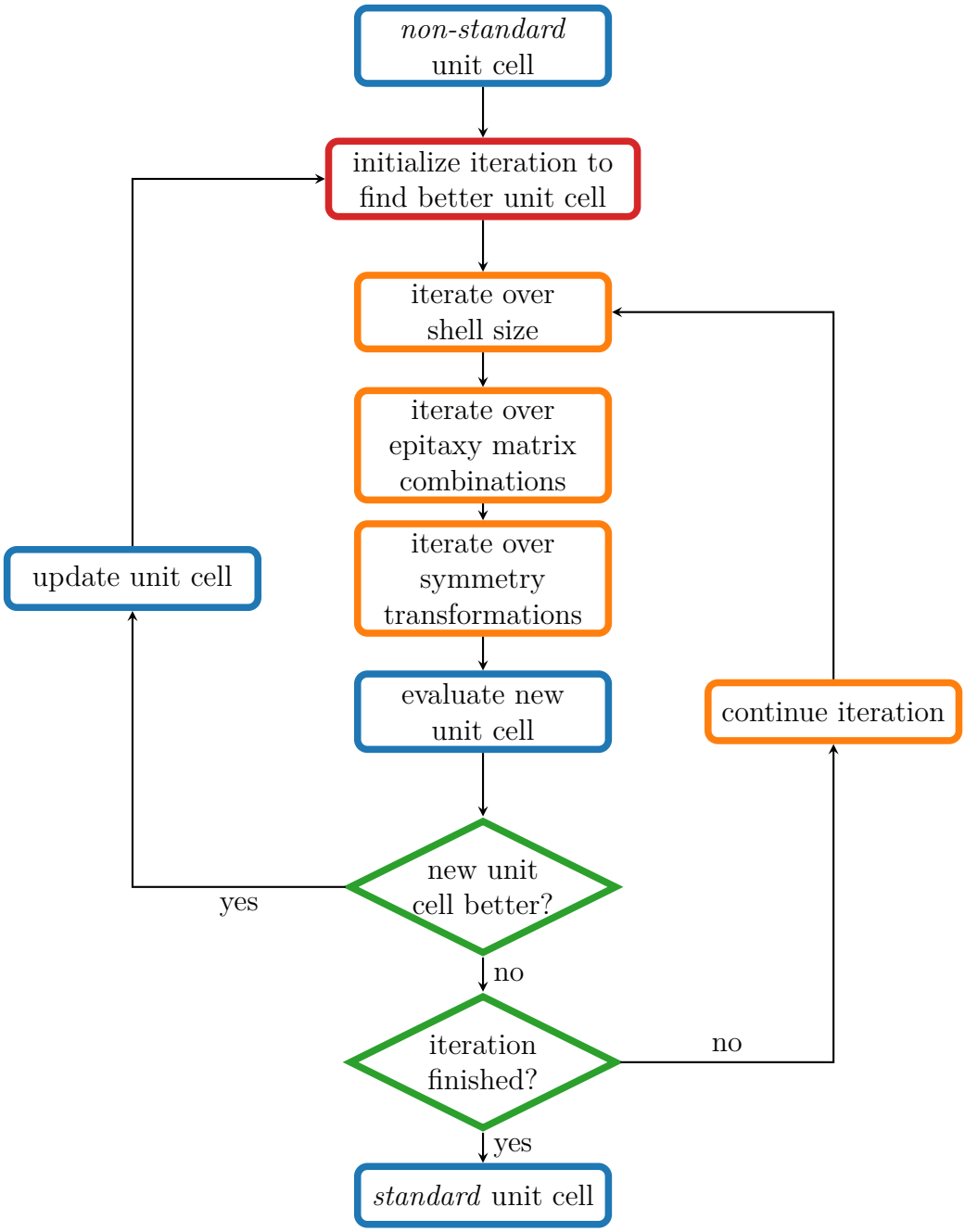


Figure 13: Flow diagram of the algorithm to determine the *standard* unit cell

2.4 Combining Geometries and Unit Cells

In the previous two chapters, we discussed our methods to find local adsorption geometries and unit cells. Here, we will elaborate on combining these components to generate all possible surface structures in a given configuration space.

2.4.1 Discretizing the Search Space

Before discussing the methods used to generate all possible structures, it is unavoidable to chew through some concepts and abstractions. Therefore, we will explain our method to course-grain the configuration space and discuss two ways to represent the discretized coordinates.

We discussed in chapter 2.2, that local adsorption geometries are local minima of the potential energy surface of an isolated molecule on the substrate. Simply put, they represent the way a molecule latches onto the surface. It is important to note that local adsorption geometries are invariant under the symmetry operation of the substrate.

Hence, translating a local adsorption geometry by a primitive substrate lattice vector will result in the same local adsorption geometry. In other words, the substrate conveniently provides a natural discretization in form of the primitive substrate unit cells, where every grid point is a symmetry equivalent position for a geometry. In addition we can apply every symmetry operation in the substrates point-group to a local adsorption geometry and get an equivalent geometry. This allows us to build a set of equivalent geometries for every local adsorption geometry, as shown in figure 14.

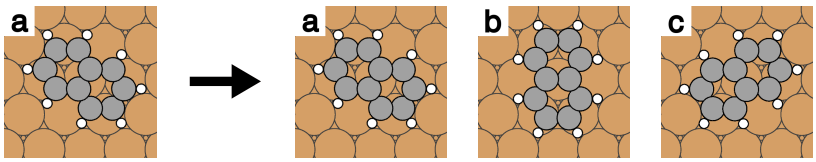


Figure 14: Left: Original local adsorption geometry of naphthalene on copper (111)
Right: A set of symmetry equivalent geometries
of one local adsorption geometry
a) original local adsorption geometry
b) local geometry a rotated by 120°
c) local geometry a reflected by the vertical axis

Hereby some symmetry operations produce the same local adoption geometry. For instance, reflection by the vertical mirror axis as well as rotation by $-120°$ will map geometry a in figure 14 onto geometry c. To detect similar symmetry equivalents, we calculate Cartesian distances between respective atoms of two geometries. If these distances are smaller than a symmetry threshold, we consider the geometries identical and only keep a list of dissimilar symmetry equivalent geometries for further use.

Further, the unit cells generated according to chapter 2.3 also exist on the primitive lattice grid. The grid points act as equivalent sites to place local adsorption geometries. Given these two observations, it makes sense to use this grid as our coordinate system. The transformation of a point in Cartesian coordinates to fractional coordinates is performed in the following way:

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \mathbf{v_1} & \mathbf{v_2} \end{pmatrix}^{-1} \cdot \begin{pmatrix} x \\ y \end{pmatrix} \tag{65}$$

Here, a and b are fractional coordinates, x and y Cartesian coordinates and $\mathbf{v_1}$ and $\mathbf{v_2}$ are the primitive substrate lattice vectors. Furthermore, a and b are integers, which avoids edge cases in numerical treatment. This representation will hereafter be referred to as 2D representation.

A second possibility to represent our problem is the 1D representation. We will see in chapter 2.4.4 that it allows us to uniquely identify structures. To construct the 1D representation, we systematically assign a number to every site in the unit cell, with the order depending on the 2D representations. To explain the procedure, we need to consider the fractional coordinates of grid-points. They can be expressed as tuples (a, b). We simply order these tuples according to the size of their elements to find the 1D representation.

To illustrate the two different representations, we consider the following example. We wish to find the 1D and 2D representation for a unit cell, with the epitaxy matrix $\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$ on a Cu(111) substrate. Figure 15 shows the two representations.

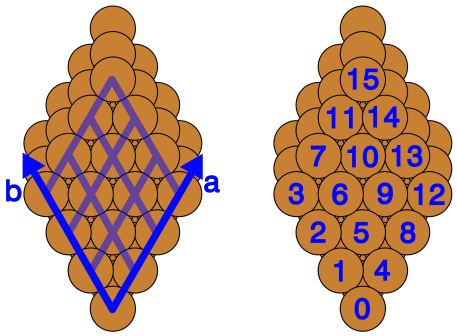


Figure 15: Example of the 1D and 2D representation for a Cu(111) unit cell
Left: 2D representation of the grid positions in a unit cell
Right: 1D representation of the grid positions in a unit cell

We perform the transformation from the 2D to the 1D representation in the following way: First we determine the fractional coordinates of each surface atom in the unit cell, which yields the 2D representation. Hereby, the position of each atom is given by a tuple (a, b). We now order these tuples according to the size of their elements. For example, the tuple $(0, 1)$ is smaller than $(1, 0)$. To attain the 1D representation, we simply number the ordered tuple, starting with 0. The following example illustrates this process:

2D representation	$\rightarrow$	1D representation
$(0, 0)$	$\rightarrow$	0
$(0, 1)$	$\rightarrow$	1
$(0, 2)$	$\rightarrow$	2
$(0, 3)$	$\rightarrow$	3
$(1, 0)$	$\rightarrow$	4
...		...
(a, b)	$\rightarrow$	α

To summarize, we have found a suitable discretization of the configuration space in form of the primitive substrate lattice as well as the local adsorption geometries and their symmetrical equivalents. Additionally, the points on the grid can be represented in 2D fractional coordinates and in a 1D representation. The latter will become important in the next chapter.

2.4.2 Assembling Structures

In the previous chapter we explained our method to discretize and represent the configuration space. Additionally, we composed a set of building blocks, namely the unit cell and the local adsorption geometries together with their symmetry equivalents. Armed with this toolkit, we can start assembling structures.

The concept behind building a structure is remarkably simple. We place a number of local adsorption geometries onto the discretization-grid, that spans the unit cell and prevent molecule collisions, since these would result in unphysical structures. Therefore, we enforce that the Cartesian distances between the atoms of respective molecules are larger than a certain threshold. To determine said threshold, we consider atomic parameters, such as the van der Waals radius or the covalent radius. Obviously, these parameters differ from species to species. Therefore, we use a separate threshold for each different pair of atoms.[31]

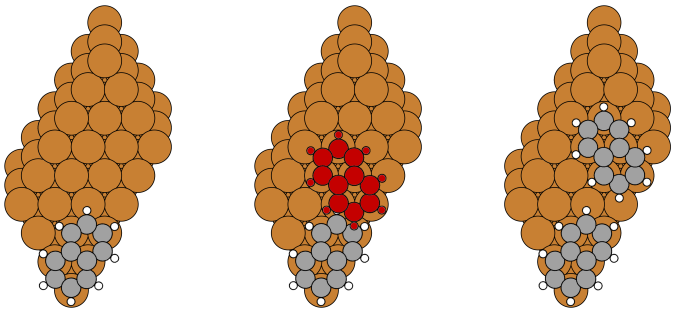


Figure 16: Process of assembling a structure
Left: We place the first local adsorption geometry.
Middle: A second geometry is added, but it collies with the first one.
Right: We move the second geometry to another point on the grid and find an appropriate position.

2.4.3 The Structure Hash

As previously stressed, the main challenge for structure search results from the large number of possible structures. Therefore, it would be desirable to represent structures in an efficient and easily comparable way. Since these requirements are reminiscent of the purpose hash-functions have in informatics, it seems fitting to call such a representation *structure hash*.

To illustrate the necessity for such a structure hash, let us consider the important task of identifying symmetrically unique structures. The straight-forward approach would be to compare the Cartesian coordinates of structures, but this is inefficient. Repeatedly computing distances between individual atoms for millions of structures is time-consuming, requires a high degree of accuracy to account for edge cases and necessitates to keep large coordinate arrays in memory.

A structure hash would significantly improve the efficiency of such operations and hence constitutes an integral component of SAMPLE. In simple terms, the structure hash is a one-to-one representation of a symmetrically unique structure that incorporates the epitaxy matrix, the 1D representation and the local adsorption geometries. Grouped together in an integer-tuple, these three pieces of information constitute the structure hash:

$$(n_1, n_2, n_3, n_4, g_1, g_2, ..., g_n, p_1, p_2, ..., p_n)$$

Here, n_1 thru n_2 are the elements of the epitaxy matrix, g_1 thru g_n are integers, that represent the local adsorption geometries and p_1 thru p_n are the positions of the respective geometries in 1D representation.

To illustrate how the structure hash corresponds to the actual structure, we consider the following example. Let $(6, 0, 0, 4, 1, 5, 0, 15)$ be the structure hash. Figure 17 shows that the first four elements $(6, 0, 0, 4)$ determine the unit cell. The next two elements $(1, 5)$ constitute the local adsorption geometries, and the last two integers $(0, 15)$ define the positions of the geometries.

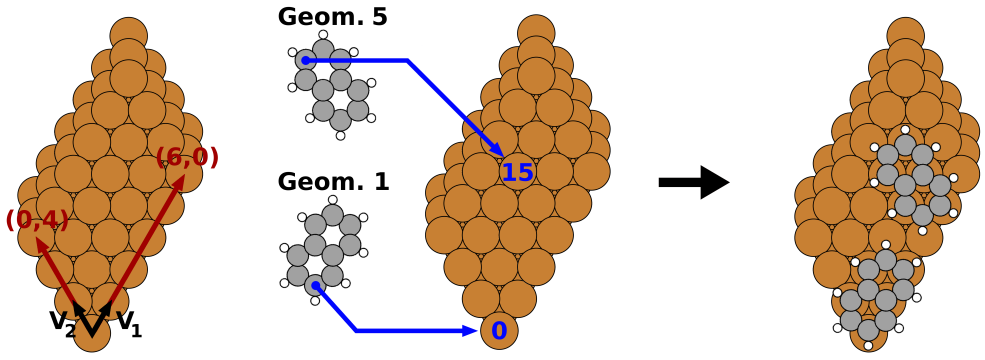


Figure 17: Schematic of the relation between the structure hash, in this case $(6, 0, 0, 4, 1, 5, 0, 15)$, and the geometry of the actual structure

By design, the structure hash does not contain information about the substrate and the geometries. This data is the same for all structures and can therefore be saved globally. Conversely, reconstructing a structure from its hash requires to know the primitive substrate unit cell, as well as the local adsorption geometries and their symmetry equivalents. Therefore, we globally save this system information to fulfill the stipulated one-to-one correspondence between structures and hashes.

Apart from the reduction in disc and memory space, the most important function of the structure hash is to identify symmetry unique structures. To this end, the structure hash function assigns the same hash to all structures, which are symmetry equivalent to each other. Structures may be equivalent by translation symmetry or by any point-group symmetry of the unit cell. The symmetries of the unit cell are a subset of the substrate symmetries, which map the unit cell onto the untransformed grid (see figure 18).

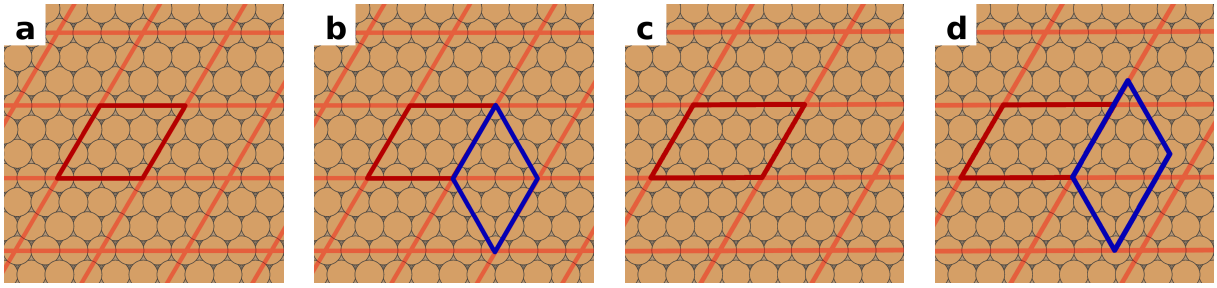


Figure 18: Example for symmetries of the unit cell:
The Cu(111) surface has three mirror axis as well as a 120° rotation symmetry
a, b) unit cell with a 120° rotation symmetry
c, d) unit cell, that does not have a 120° rotation symmetry

The hash function uses these symmetries to first generate all symmetry equivalents of a given structure. Secondly, it determines the hash-tuple for all these structures. Finally, element-wise comparison allows to determine the smallest of hash-tuple, which we consider the structure hash.

2.4.4 Generating all Structures

The integral goal of SAMPLE is to consider all possible structures in a given configuration space. This differentiates it from other structure search methods, such as genetic algorithms. To achieve this goal, we systematically construct structures in a five-step process. The input for the structure generator consists of a unit cell, a set of local adsorption geometries, the number of molecules in the unit cell (and thereby the coverage) as well as the distance thresholds for different pairs of atoms (see chapter 2.4.2).

1. **Remove local adsorption geometries that are too large**
   First, we determine all geometries and their symmetry equivalents that fit in the unit cell. We place a single geometry or symmetry equivalent in the unit cell and test for collisions with its periodic replicas according to the approach discussed in chapter 2.4.2.

2. **Calculate all pair collisions**
   In this step, we generate all possible pairs of geometries in the given unit cell. We test the pairs for collisions within the cell and with periodic replicas and save the results in a collision table. A pair-hash $(g1, g2, \alpha, \beta)$ serves as an identifier for each pair. The parameters $g1$ and $g2$ are the geometries and the parameters α and β are the elements of the vector from geometry $g1$ to $g2$ in fractional coordinates. This representation of pairs allows for easy indexing of the collision check results.

3. **Initialize first layer of the tree**
   The approach to systematically generate all structures, takes inspiration from tree diagrams. The first layer of the tree contains all structures with one molecule per unit cell, the second layer all structures with two molecules and so on. In step three, we initialize the first layer by placing each molecule at the origin of the unit cell.

4. **Add consecutive layers to the tree**
   In step four, we build the consecutive layers of the tree until we reach the desired number of molecules per unit cell. A layer contains all combinations of its respective number of geometries. Each combination of geometries comprises all arrangements of these geometries. Figure 19 illustrates the tree.

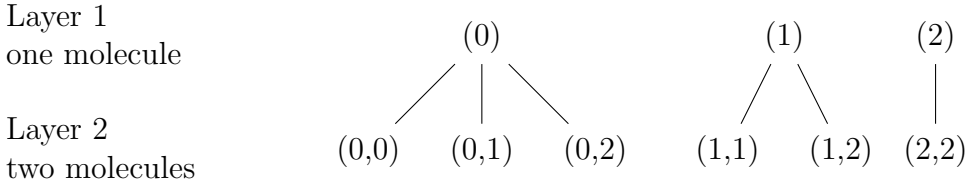


Figure 19: Schematic for building structures:
Layer 1: all structures with geometries 0, 1, 2 are generated
Layer 2: all structures with combinations of two geometries are generated

   A new layer consists of all combinations of each new geometry with the structures from the previous layer. For example, layer one contains all structures with one molecule per unit cell. For layer two, we take each structure from layer one and add a second geometry. Hereby we cycle through all geometries and successively try to place each geometry onto every position of the unit cell. The preparations done in step 2 now allow to check collision for each new geometry and position by simply looking up the respective pair in the collision table. If no collisions are found, we save the new two-molecule structure. Adding further layers follows the same principle.

5. **Remove symmetry equivalent structures**
   Finally we reduce the structures generated in step 4 to get a unique set. We calculate the structure-hash as discussed in chapter 2.4.4 for every structure. If two or more structures have the same structure hash, we only keep one.

Following this procedure allows SAMPLE to generate all structures in a given unit cell. To generate all possible structures in a given configuration space, we simply repeat the structure generation for all possible unit cells.

2.5 Bayesian Learning

In the previous chapter we discussed the ability of the SAMPLE approach to generate all possible structures. Consequently, it would be desirable to rank all these structures with regard to their adsorption energy. However, calculating the energies of all possible structures is unfeasible due to the large number of structures (see chapter 3.2.2).

To tackle this challenge, we employ Bayesian linear regression, which is based on a simple energy model and a number of physically motivated prior assumptions. Training this model with a small number of structures, whose energies were calculated with DFT, allows to predict the energies of all structures. The flowchart below outlines the basic idea.

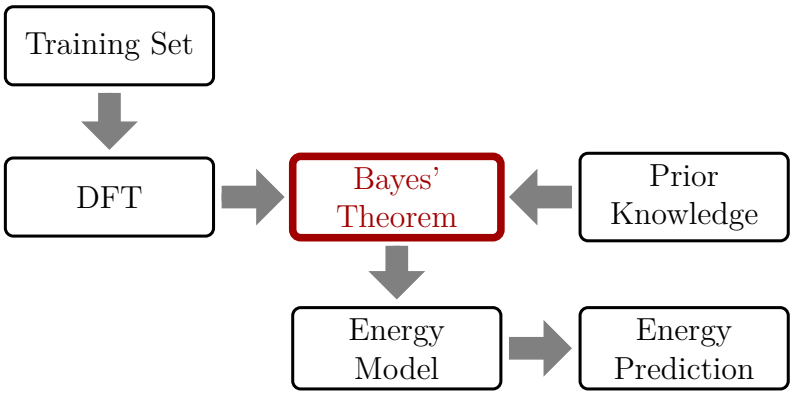


Figure 20: Implementation of Bayesian linear regression in SAMPLE

2.5.1 Energy Model

The energy model proposes that the energy of a structure can be approximated by considering one- and two-body interactions. Figure 21 illustrates these interactions.

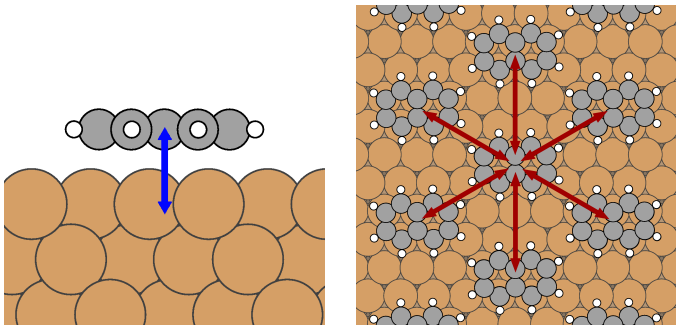


Figure 21: Schematic representation of the energy model
Left: one-body interaction
Right: two-body interaction

One-body interactions are the interactions between the molecule, at a designated adsorption site, and the substrate. In other words, we assign each local adsorption geometry a scalar energy value.

Two-body interactions take account of the energy contributions of individual pairs of local adsorption geometries. Put another way, we assign each distinct pair of geometries a scalar energy value. The discretization employed in SAMPLE guarantees that the number of different geometry pairs within a proposed distance cutoff is finite. The motivation for such a distance cutoff is the assumption, that interaction energies between molecules decay to zero at large distances ($15 - 20$Å, see chapter 3.6.2). Therefore, it is possible to describe all structures by considering two-body contributions of these relevant pairs.

Using this energy model, we can express the energy of a structure as a sum of one-body interactions U_i and two-body interactions V_p, whereby n_i and n_p describe how often these interactions occur.

$$E = \sum_i n_i U_i + \sum_p n_p V_p \tag{66}$$

Using equation 66 allows to predict the energies of all structures, provided we know all one- and two-body interactions. However, it is difficult to calculate these model parameters directly. We could, for instance, use a large unit cell with periodic boundary conditions, to find the one-body energy of an isolated local adsorption geometry. This approach would, however, neglect collective effects, such as depolarization, that result from a closed packed layer. Therefore, it would be desirable to calculate energies of closed packed structures and infer the one- and two-body interactions. This way, the aforementioned effects would be implicitly included in the energy model.

We can use equation 66 to formulate a linear regression, which will bring us closer to the aforementioned goal. Therefore, we first rewrite equation 66 as a vector multiplication.

$$E = \mathbf{n} \cdot \boldsymbol{\omega}^T \tag{67}$$

Vector $\mathbf{n}$ indicates which interactions occur. Hence, its elements are zero for interactions that are not present in the structure. For interactions that occur, the respective elements in $\mathbf{n}$ consist of the number of occurrences divided by the number of molecules in the unit cell N_{ads}.

$$\mathbf{n} = \left(0, \dots, \frac{n_i}{N_{ads}}, 0, \dots, \frac{n_p}{N_{ads}}, 0, \dots\right) \tag{68}$$

The vector ω contains the one- and two-body interactions.

$$\boldsymbol{\omega} = (U_1, U_2, \dots, V_1, V_2, \dots) \tag{69}$$

For a set of structures, equation 66 becomes a matrix multiplication of a model matrix $\mathbf{N}$, whose lines are the vectors $\mathbf{n}$, and the vector $\boldsymbol{\omega}$.

$$\mathbf{E} = \mathbf{N} \cdot \boldsymbol{\omega}^T \tag{70}$$

The predicted energies $\mathbf{E}$ for a set of structures, deviate from the true energies $\mathbf{E}_{DFT}$ (in our case the DFT energy) by a normally distributed error ϵ.

$$\mathbf{E}_{DFT} = \mathbf{E} + \boldsymbol{\epsilon} = \mathbf{N} \cdot \boldsymbol{\omega}^T + \boldsymbol{\epsilon} \quad \textit{with} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}) \tag{71}$$

The expression above has the form of a standard linear regression. However, calculating the energies of a number of structures $\mathbf{E}_{DFT}$ and trying to find the least squares solution will fail, if the system is under-determined, which is often the case for the SAMPLE approach. A solution for this problem is to employ the framework of Bayesian linear regression.

2.5.2 Feature Vector

Before we discuss Bayesian linear regression, let us introduce the feature vector. This property of a geometry pair allows to calculate the similarity between different pairs of geometries, whereby the similarity measure is the distance in feature space.

To determine the feature vector, we first calculate distances between the respective atoms of a geometry pair, whereby we order these distances according to the specific pair of atom species. For instance, we calculate the distance between C-atoms on geometry A and H-atoms on geometry B. The feature vector itself consists of sections belonging to specific atom pairs. Within such a section the feature vector elements contain the fraction between a distance threshold d_0 and the atom distance d_α, raised to a decay power n.

$$f_\alpha \;=\; \left(\frac{d_0}{d_\alpha}\right)^n \tag{72}$$

Note, that the atom distance d_α serves as denominator, hence close pairs of molecules lead to large elements in the feature vector, whereas distant pairs lead to small elements. Therefore, small geometric differences of close molecule pairs entail large differences of the feature vectors. At the same time, feature of distant molecule pairs show small variations. This construction of the feature vector is physically motivated. Since interactions decay with distance, close pairs contribute much stronger to the total energy of a structure than distant pairs. Additionally, the energy contributions of close pairs vary more strongly with geometric differences than the small energy contributions of distant pairs.

We further order the elements in each section according to their magnitude to obtain the final expression of the feature vector.

$$\mathbf{f} = \left(\underbrace{d_{0,SP1}, d_{1,SP1}, \ldots}_{\text{species pair 1}}, \underbrace{d_{0,SP2}, d_{1,SP2}, \ldots}_{\text{species pair 2}}, \underbrace{d_{0,SP3}, d_{1,SP3}, \ldots}_{\text{species pair 3}}, \ldots \right) \tag{73}$$

Finally, a feature dimension determines the number of elements of a species pair in the feature vector. Hereby, we retain first respective elements of each section that are also the largest and most important ones.

2.5.3 Bayesian Linear Regression

In the previous chapter, we described an energy model consisting of one- and two-body interactions, which allows to predict the adsorption energies of all structures. We also succeeded in formulating a linear regression for these model parameters $\boldsymbol{\omega} = (\mathbf{U}, \mathbf{V})$, which we now aim to solve with the aide of Bayes' theorem. This approach enables us to express the unknown probability for the one- and two-body interactions $p(\boldsymbol{\omega}|\mathbf{E}_{DFT})$ given a set of calculated energies $\mathbf{E}_{DFT}$ in terms of two known probability distributions, namely a likelihood and a prior.

$$p(\boldsymbol{\omega}|\mathbf{E}_{DFT}) = \frac{p(\mathbf{E}_{DFT}|\boldsymbol{\omega})\ p(\boldsymbol{\omega})}{p(\mathbf{E}_{DFT})} \tag{74}$$

The likelihood $p(\mathbf{E}_{DFT}|\omega)$ directly follows from equation 71, whereby γ is the uncertainty between the energy model and DFT.

$$p(\mathbf{E}_{DFT}|\boldsymbol{\omega}) \approx \exp\left(-\frac{1}{2\gamma^2}||\mathbf{E}_{DFT} - \hat{N}\boldsymbol{\omega}||^2\right) \tag{75}$$

The prior probability allows to include physical knowledge about the system and thereby enables us to find solutions for the one- and two-body interactions, even if the problem posed by equation 71 is under-determined. We can write the prior as a normal distribution with a prior mean $\boldsymbol{\omega}_0$ and a covariance matrix C:

$$p(\boldsymbol{\omega}) \approx \exp\left(-\frac{1}{2}(\boldsymbol{\omega} - \boldsymbol{\omega}_0)^T C^{-1}(\boldsymbol{\omega} - \boldsymbol{\omega}_0)\right) \tag{76}$$

The prior mean takes account of our initial knowledge regarding the one- and two-body interactions. For the one-body interactions, the adsorption energy of a local adsorption geometry, i.e. an isolated molecule, provides a convenient estimator. Regarding the two-body interactions, we lack such an estimator. Since their energy contribution can either be attractive or repulsive, the obvious choice is zero (this is equivalent to non-interacting).

The prior covariance enables us to include correlations between different two-body interactions and rests upon the assumption, that similar pairs of geometries have similar two-body interactions. Further, the covariance also accounts for the supposition that two-body interactions decay to zero at large distances. To this end, we use a multiplicative exponential kernel, which consists of two contributions:

- First, the distance in feature space $|\mathbf{f}_i - \mathbf{f}_j|$ provides a measure for the similarity between pairs, whereby $\mathbf{f}$ is the feature vector. Here, α is a decay length in feature space.

$$C_{ij} = \sigma_i \sigma_j \exp\left(-\frac{|\mathbf{f}_i - \mathbf{f}_j|}{\alpha}\right) \tag{77}$$

- Secondly, the minimum distance $d_{min,i}$ between the atoms of two geometries accounts for the decay of the two-body interactions. Here d_0 is a distance threshold, τ is a decay length in real space and σ_0 is the uncertainty of the prior.

$$\sigma_i = \sigma_0 \exp\left(-\frac{d_{min,i} - d_0}{\tau}\right) \tag{78}$$

More precisely, this term controls the width of the Gauss peak in the prior distribution. For small distances, σ_i becomes large and therefore allows for a larger variance between the two-body interactions and their prior (initialized with 0 meV, i.e. non-interacting). For large distances, σ_i becomes small and therefore only allows for a small variance.

Having discussed likelihood and prior, we can write the posterior probability. Hereby, we neglect the marginal probability $p(\mathbf{E}_{DFT})$. The marginal probability is constant and therefore not relevant for our purposes. Some mathematical effort enables us to express the posterior probability as a Gaussian (the derivation is similar to appendix A).

$$p(\omega|\mathbf{E}_{DFT}) \approx \exp\left(-\frac{1}{2}(\boldsymbol{\omega} - \bar{\boldsymbol{\omega}})^T A^{-1}(\boldsymbol{\omega} - \bar{\boldsymbol{\omega}})\right) \tag{79}$$

Here, the new covariance matrix has the following expression:

$$A^{-1} = \frac{\hat{N}^T \hat{N}}{\gamma^2} + C^{-1} \tag{80}$$

The posterior mean $\bar{\boldsymbol{\omega}}$ is also the expectation value for the vector ω, which contains the one- and two-body interactions.

$$\bar{\boldsymbol{\omega}} = A\left(\frac{\hat{N}^T E_{DFT}}{\gamma^2} + C^{-1}\boldsymbol{\omega}_0\right) \tag{81}$$

Equation 81 is the best estimator for vector ω. To fit the energy model, we can supply a set of structures with energies from DFT and calculate the one- and two-body interactions. This in turn allows to predict the energies of all structures.

2.5.4 D-optimal Selection

Finally, let us briefly discuss how to select the training set for the Bayesian linear regression. The aim is to choose a set of structures, which maximizes the information gain per computational effort.

In equation 79, the covariance matrix A determines the uncertainty of the mean vector $\bar{\boldsymbol{\omega}}$. Hence, minimizing A (equivalent to maximizing A^{-1}) will improve the accuracy of the fit coefficients. Since the *size* of a matrix in not defined, this cannot be done directly.

However, experimental design theory provides a number of criteria for gauging the information contained in a matrix A. Here, we present a selection, albeit incomplete, of the most common optimality criteria.

- **A-optimality** seeks to minimize the trace of A.
- **D-optimality** seeks to minimize the determinant of A.
- **E-optimality** seeks to minimize the largest eigenvalue of A.

The criterion chosen for SAMPLE is D-optimality. Hence, we seek to minimize the determinant of the covariance matrix:

$$\min\{\det(A)\}$$

Equation 80 shows that matrix A consists of the model matrix N and the prior covariance matrix C. Both matrices depend on the choice of the training set. Therefore, we systematically select a training set that minimizes the determinant of A and thereby maximizes the efficiency of SAMPLE.

2.6 Predicting Phase Diagrams

Density functional theory, our method of choice for calculating energies, describes a material at a temperature of 0 K. For experiments and technical applications, however, the interesting temperature range lies significantly above absolute zero. In addition, we know from experience that materials can form different polymorphs at different temperatures and pressures. These polymorphs can have vastly different physical properties and only specific structures of a material may be suitable for a given application. Having the ability to predict such polymorphs would therefore be highly desirable.

So far we discussed the ability of SAMPLE to provide predictions of adsorption energies at 0 K. To include the influence of temperature and pressure, we employ ab initio thermodynamics. The relevant quantity for this effort is the Gibbs free energy of adsorption, which we discussed in chapter 1.4.

$$\gamma \;=\; \frac{1}{A}(\Delta E - \mu_{ads} N_{ads} + pV - TS_{conf} N_{ads} + F_{vib})$$

The current implementation of ab initio thermodynamics in SAMPLE neglects the last three terms of the above equation. As discussed in chapter 1.4, disregarding the mechanical work pV and the contribution from configuration entropy $TS_{conf}N_{ads}$ does not significantly alter the prediction accuracy. However, for the contribution from vibrational enthalpy F_{vib}, such arguments cannot be offered. In fact, we expect this contribution to be important. However, the discrete energy model used in SAMPLE is currently not capable to predict vibration modes of structures. Therefore we approximate the Gibbs free energy of adsorption in the following way:

$$\gamma(p,\, T) \;=\; \frac{1}{A}(\Delta E - \mu_{ads} N_{ads})$$

Here, A is the area of the unit cell, ΔE is the adsorption energy, μ_{ads} is the chemical potential of the adsorbates in gas phase and N_{ads} is the number of molecules in the unit cell. To find the phase that forms at a certain temperature and pressure, i.e. at a point in the phase diagram, we calculate $\gamma(p,\, T)$ for all structures and determine the minimum. Repeating this calculation for a range of temperatures and pressures allows to generate the full phase diagram.

3 Results

We demonstrate the capability of SAMPLE for the system of naphthalene on Cu(111). Naphthalene is a small, organic molecule, that forms commensurate structures in the Cu(111) substrate[32][33]. Therefore, it is an ideal test system. In this chapter we will discuss details, settings and results of the computational experiments performed within this work.

3.1 Experimental Structures

Experimentalists have reported three commensurate structures for the system of naphthalene on Cu(111)[32][33]. Additionally, they found a number of incommensurate structures[33]. Two of the commensurate polymorphs contain one naphthalene molecule in the unit cell, while the third polymorph contains six molecules in the unit cell. Further, the reported structures contain only flat lying molecules.

3.1.1 Experimental Polymorph One

Experimental polymorph one, reported by Yamada et. al.[32].

area	12 Cu-surface-atoms per unit cell
molecules in unit cell	1
coverage	12 Cu/naphthalene
exposition temperature	T = 120 K
epitaxy matrix	$A = \begin{pmatrix} 4 & 2 \\ 0 & 3 \end{pmatrix}$

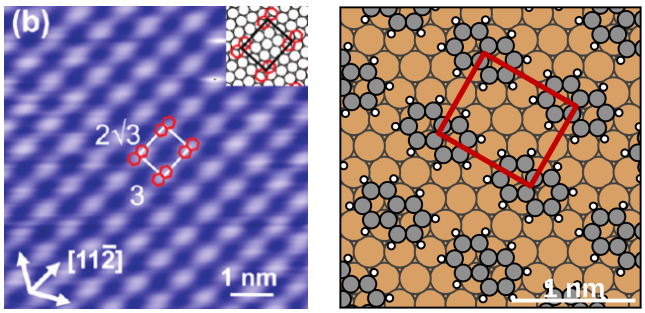


Figure 22: Left: STM (scanning tunneling microscope) image of polymorph one[32]
Right: SAMPLE rendering of polymorph one

3.1.2 Experimental Polymorph Two

Experimental polymorph two, reported by Forker et. al.[33].

area	15 Cu-surface-atoms per unit cell
molecules in unit cell	1
coverage	15 Cu/naphthalene
exposition temperature	T = 140 K
epitaxy matrix	$A = \begin{pmatrix} 5 & 1 \\ 0 & 3 \end{pmatrix}$

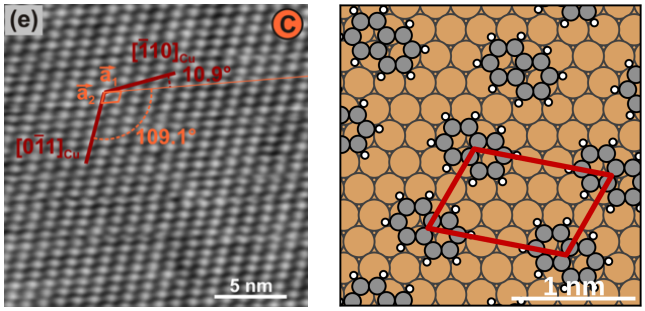


Figure 23: Left: STM image of polymorph two[33]
Right: SAMPLE rendering of polymorph two

3.1.3 Experimental Polymorph Three

Experimental polymorph three, reported by Yamada et. al.[32].

area	75 Cu-surface-atoms per unit cell
molecules in unit cell	6
coverage	12.5 Cu/naphthalene
exposition temperature	T = 120 K
epitaxy matrix	$A = \begin{pmatrix} 10 & 5 \\ -5 & 5 \end{pmatrix}$

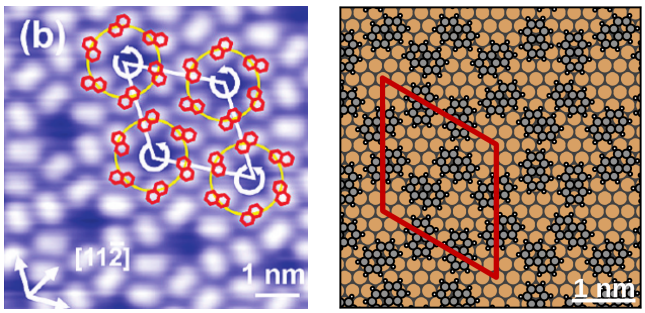


Figure 24: Left: STM image of polymorph three[32]
Right: SAMPLE rendering of polymorph three

3.2 Possible Structures

In this chapter we will discuss details and settings regarding the process of generating structures with SAMPLE. Based on the assumptions of SAMPLE (see chapter 2) we generate commensurate, closed packed structures that are in thermodynamic equilibrium.

3.2.1 Parameters for Generating Structures

The experimental polymorphs we discussed in chapter 3.1 exhibit coverages of 12 to 15 Cu-atoms per naphthalene molecule and have one to six molecules per unit cell. Unbiased structure search from first principles therefore requires including a range of coverages larger than or equal to the experimental range. Hence, we select a coverage range of 10 to 15 Cu-atoms per naphthalene molecule. Additionally, the experimental structures contain up to six flat lying molecules in the unit cell. Therefore, we consider structures with flat lying molecules. Further, we focus on structures with up to four molecules per unit cell in this work.

To generate the required unit cells and structures, we have to determine three parameters. First, the minimum cell width results roughly from the smallest extent of the flat lying naphthalene molecule (≈ 5Å). Secondly, the symmetry tolerance must allow to detect the correct number of symmetry equivalent local adsorption geometries (see chapter 2.4.1). Thirdly, geometric considerations necessitate only a distance threshold for the H-atoms (see chapter 2.4.2). This is justified, since all outside atoms of a flat lying naphthalene molecule are H-atoms. We determine the distance threshold in such a way that structures remain geometrically stable, i.e. the structures do not rearrange during geometry optimization.

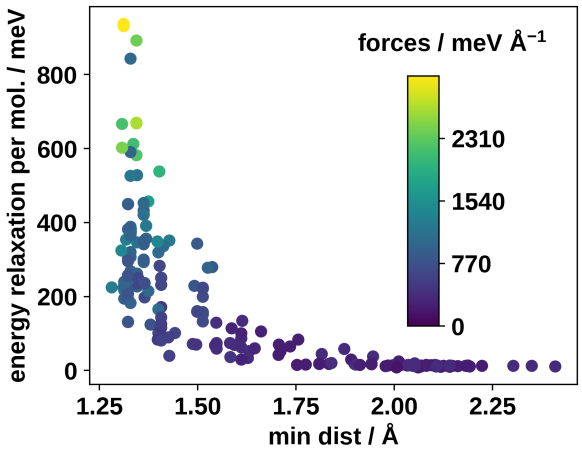


Figure 25: Energy gained through geometry relaxation; Energy relaxation is given by energy before minus energy after relaxation; As expected, we observe larger forces for more unstable structures.

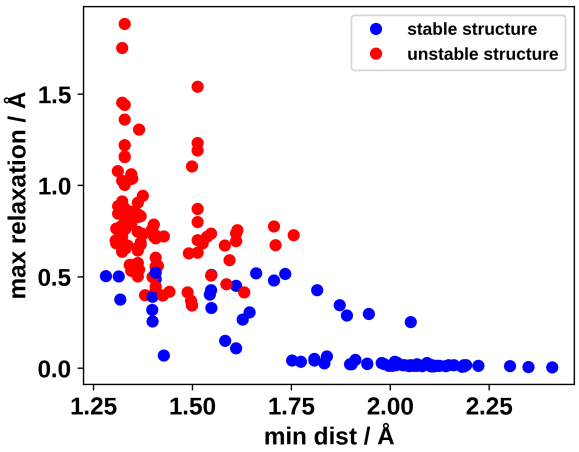


Figure 26: Stability of structures during geometry optimization

In order to test the stability of structures during optimization as well as the relaxation of their total energy, we perform local geometry optimizations on structures with one and two molecules per unit cell and a minimum atom distance of 1.3Å. Figures 25 and 26 show the results of this investigation. The stability rating (stable/unstable) results from a comparison of Cartesian coordinates. First, we identify local adsorption geometries with a symmetry threshold of 0.3Å and secondly, we compare their optimized positions to the original positions with a symmetry threshold of 0.3Å as well.

Based on these results, we expect that structures with a minimal atom distance of more than 2Å are stable. Further, structures whose minimal atom distance is lager than 1.5Å may still be stable, whereby some of these structure also show noticeable energy relaxations (about 200 meV). Therefore, we select a distance threshold of 1.5Å. In summary we generate structures using the following settings:

Table 1: Settings for structure generation

parameter	**setting**
minimum cell width	4.0 Å
symmetry tolerance	0.3 Å
distance threshold for H-atoms	1.5 Å

3.2.2 Generated Structures

In chapter 3.2.1 we formulated requirements for an unbiased SAMPLE run. Additionally, we discussed the necessary parameters for the structure generator (see table 1). Considering these parameters, we generate structures of flat lying molecules with the following unit cell sizes:

Table 2: Unit cell size for different number of molecules

Nr. molecules	**area / Cu**
1 molecule	10 - 15
2 molecules	20 - 30
3 molecules	30 - 45
4 molecules	40 - 60

In order to generate all structures, the unit cells within these area ranges are populated with local adsorption geometries according to the discussion in chapter 2.4. Figure 27 shows the number of structures, we generate with the unit cells from table 2. Further, the figure demonstrates the exponential dependence of the number of possible structures on the number of molecules per unit cell.

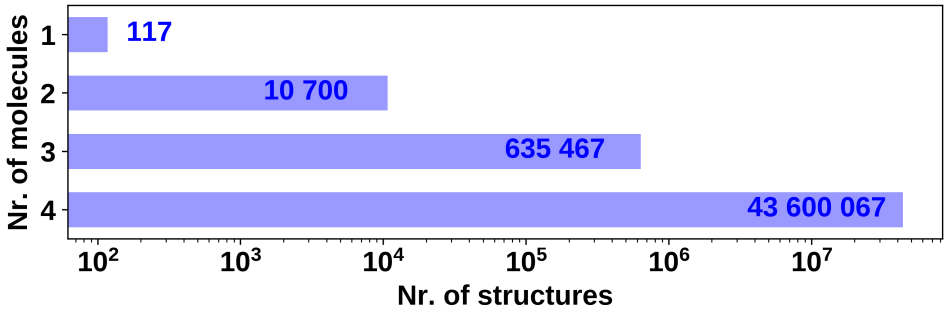


Figure 27: Number of structures with coverage 10 - 15 Cu/naphthalene for different numbers of molecules in the unit cell

3.3 Calculation Settings

SAMPLE employs dispersion-corrected density functional theory, to obtain adsorption energies with high accuracy. In this chapter we will discuss the important settings for these DFT calculations.

First, we use the FHI-aims[30] quantum chemistry code with PBE[34] exchange correlation functional and the TS^{surf}[35] dispersion correction. Further, we use periodic boundary conditions and the repeated slab approach to calculate surfaces.

Secondly, we converge the numerical settings for said code and functional to an accuracy of 20 meV for the total energy per molecule. Hence, the DFT uncertainty is smaller, than the thermal energy-scale $1k_BT$ at room temperature, which is approximately $26meV$. This allows to distinguish different structures according to their energetic favorability. For reasons of brevity, this chapter presents the final settings and refers to appendix D for an exhaustive list of convergence tests.

Finally, to reduce the computation expense, we calculate single points. This is justified by the knowledge gained in chapter 3.2.

3.3.1 Substrate Settings

We already discussed that we use periodic boundary condition based DFT to calculate the adsorption energies of structures. In order to calculate surface structures in this way we use the repeated slab approach. Hereby, the unit cell contains a substrate slab with a thickness of several atom layers. A thick vacuum separates the slabs in vertical direction. We chose a vacuum thickness of about 50 Å.

We optimize the lattice constant of the Cu bulk to a value $a = 3.602$ Å (see figure 28), which corresponds to a layer distance of 2.0796 Å in the (111) direction (see appendix D). Performing layer convergence tests allows us to determine the required number of layers to be five. Additionally, we optimize the layer distance of the two uppermost layers using local geometry optimization.

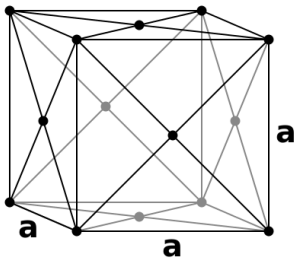


Figure 28: Bulk unit cell of a fcc (face centered cubic) crystal

The layer convergence results in an uncertainty of about 20 meV per 16 surface atoms, compared with a higher number of layers. However, since we only compare calculations with the same number of layers, we assume that this is a systematic error. Further, we are mainly concerned with adsorption energies, where the energy contribution of the slab is subtracted. Hence, we expect that the uncertainties mostly cancel.

3.3.2 k-Grid Settings

Exhaustive structure search, as performed by SAMPLE, requires considering and comparing unit cells of different shapes. Since the results from periodic DFT calculations depend on the k-grid, we cannot compare them readily. Further, it is generally not possible to find commensurate k-grids for different unit cells.

To alleviate the problem, we first over-converge the k-grid to an uncertainty of 0.1 meV per single surface atom. This accuracy requires 80×80 k-point per surface atom, i.e. per primitive substrate unit cell.

Using these k-grid settings, we gauge the impact of the different k-grids on the total energy. For a meaningful comparison of DFT energies, the error due to different k-grids must be lower than the accuracy of the DFT calculation. To test this *k-grid-error*, we calculate the slab energies of a number of compact, as well as two elongated unit cells and compare the results. Figure 29 shows the results.

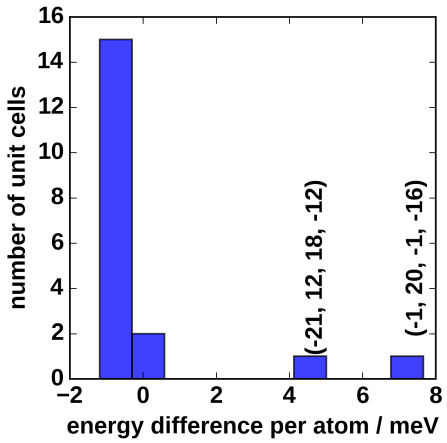


Figure 29: Difference of energy per surface atom across various unit cells; difference due to k-vectors; the two outliers have very narrow shapes

For compact unit cells the difference in slab energy is about 1 meV per surface atom or about 10 meV per molecule. The k-grid-error is therefore smaller than the DFT-uncertainty. However, for the elongated unit cells with epitaxy matrices of $(-21, 12, 18, -12)$ and $(-1, 20, -1, 16)$ the discrepancy is 5 meV and 7 meV per surface atom respectively or about 50 meV and 70 meV per molecule.

Considering these observations, it is possible to compare structures with different unit cells, as long as these unit cells are compact. Since SAMPLE enforces compact unit cells, this criterion is fulfilled. Further, we are interested in adsorption energies, where we expect that the uncertainties mostly cancel.

3.3.3 Additional Calculation Settings

Additionally, we optimize a number of calculation settings:

Table 3: Additional calculation settings

parameter	**setting**	**uncertainty**
cutoff potential	4.6	5.0 meV
radial multiplier	1	1.5 meV
basis function	tier 1 (tight) for H, C and Cu without the Cu-5g basis function	15.3 meV

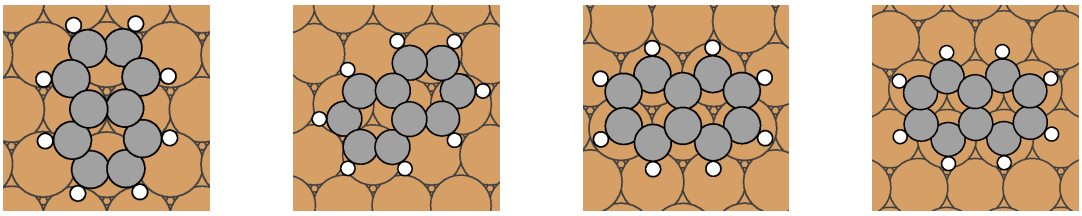
These settings result in a cumulative uncertainty of about 22 meV for the adsorption energy per molecule (coverage 16 Cu-atoms/naphthalene). However, we are mainly interested in comparing adsorption energies of relatively similar systems. Hence, we expect a significantly smaller effective uncertainty.

3.4 Local Adsorption Geometries

Using local geometry optimizations, we identify four distinct, flat lying, local adsorption geometries, that naphthalene adopts on the Cu(111) substrate. The local geometry optimizations use 16 starting geometries as starting points:

- three for the the top position
- seven for the bridge position
- three for each of the two hollow positions

To assure that the geometry optimizations converge to actual local minima of the potential energy surface, we manually perturb each converged geometry and restart the geometry optimization. Figure 30 shows the results of this procedure.



$(-1370 \pm 10)\ meV$ $(-1420 \pm 10)\ meV$ $(-1420 \pm 10)\ meV$ $(-1440 \pm 10)\ meV$

Figure 30: Local adsorption geometries of naphthalene on Cu(111)

3.5 Settings for Bayesian Linear Regression

SAMPLEs implementation of Bayesian linear regression (BLR) requires a number of hyper-parameters and the definition of a feature vector. These parameters are specific to each system. In the following we present the settings for naphthalene on Cu(111).

3.5.1 Test System

To tune the different parameters of the Bayesian linear regression model, we use a test system, consisting of 4222 structures with one to six molecules per unit cell. These structures result from an earlier test-run of the structure generator, using the parameters from chapter 3.2 with the exception that the distance threshold for H-atoms is set to 2.0 Å. To reduce the computational effort, we strip the structures of their metal substrates, i.e. we only considered naphthalene monolayers in vacuum. This is justified, since the feature vector and the majority of parameters solely depend on the geometry of the system.

3.5.2 Feature Vector

As stated in chapter 3.5.2, the feature vector serves as a descriptor for pairs of molecules. Therefore, a suitable feature vector needs to capture the different pair distances and respective orientations of the molecules. For the naphthalene molecule, we determine two obvious variants - a feature vector based on C-atoms and one based on H-atoms. Figure 31 shows both variants.

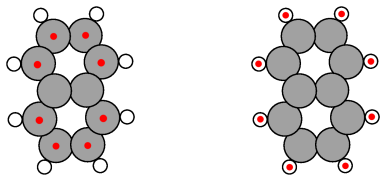


Figure 31: Left: feature vector based on C-atoms (rejected) Right: feature vector based on H-atoms (accepted)

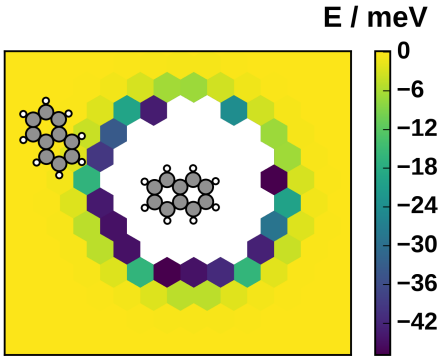


Figure 32: Interaction energy map of a naphthalene dimer in vacuum; we find attractive interactions outside the Pauli push-back region

Based on geometric considerations, we propose that the H-atoms based feature vector is better suited for the system at hand. The feature vector needs to have a high resolution for small distances and a low resolution for large distances. The shape of the naphthalene molecule is such that even though two C-atoms might be separated by a large distance (several Å), two H-atoms might be close. Therefore, a C-atom based feature vector poorly describes the system. Using distances between H-atoms to construct the feature vector, alleviates this problem.

In order to illustrate the difference between the two feature vectors, we calculate energies of various different naphthalene dimers in vacuum. The resulting interaction map (see figure 32) shows relatively isotropic interactions between naphthalene molecules (about $20 - 40$ meV for the closest positions). Further, we find slightly attractive interactions for naphthalene dimers in vacuum, outside the Pauli push-back region.

Additionally, we calculate the C-atom and H-atom based feature vectors for all these dimers respectively. In general, a good feature vector must assign dimers to positions in feature space in such a way, that they form clusters of similar energy. To visualize this property for both feature vectors, we plot the first two dimensions of the feature space. These are the dimensions with the largest elements of the feature vector. Therefore, the distance of two dimers in feature space mainly depends on the respective first and second element of the feature vector. To reiterate, if two dimers have similar energy, they should be close in feature space. If their energies are different, they should be far away from each other. Figure 33 shows this plot for the dimer from figure 32.

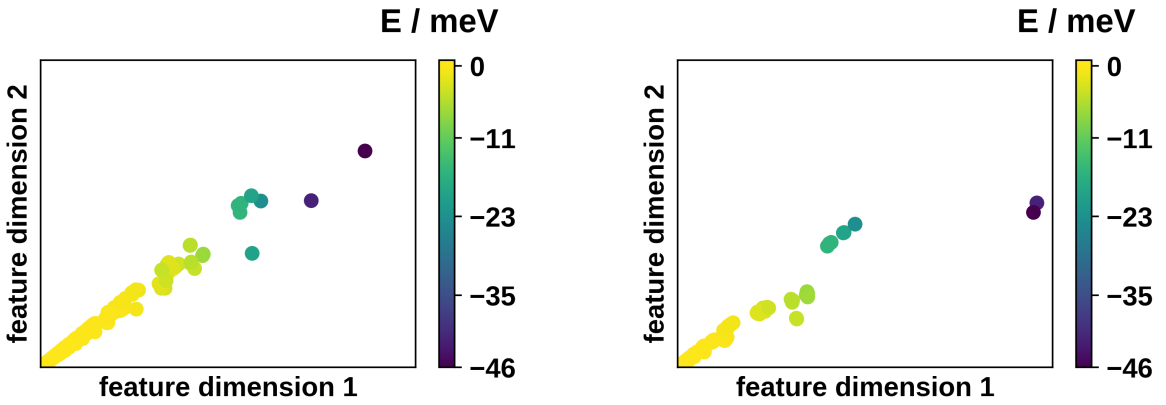


Figure 33: Comparison of the two feature vectors:
Left: C-atom based; sufficient clustering of similar energies
Right: H-atom based; clustering of similar energies is significantly better than with the C-atom based feature vector

Figure 33 shows, that both feature vectors sufficiently cluster the dimers. As expected, the H-atom based feature vector does a better job.

3.5.3 Parameters for the Feature Vector

Having determined that the H-atom based feature vector is suitable to describe the geometry of naphthalene molecule pairs, we investigate the number of pair interactions and therefore features that need to be considered. In general, three settings are important. First, d_0 is the minimal distance between the respective atoms of two molecules. Secondly, d_{max} is the distance cutoff for widely separated pairs. Specifically, we only consider molecule pairs, whose minimal atom distance lies between d_0 and d_{max}. Thirdly, Δf is the feature threshold. We consider two feature vectors as identical, if all element-wise differences are smaller than Δf. Finally, we find that a feature dimension of 16 and a

decay power of 2 produce good predictions (see chapter 3.5.5). For the naphthalene on Cu(111), we choose the following physically motivated settings.

Table 4: Parameter settings for generating the feature vector

parameter	**symbol**	**setting**
minimal distance	d_0	1.5 Å
distance cutoff	d_{max}	16.0 Å
feature threshold	Δf	0.01
feature dimension		16
decay power		2

Using these settings, we generate all features. As discussed in chapter 2.5.1, we wish to determine the two-body interaction of each pair of molecules. In other words, we want to find an energy value for each feature. Therefore a training set (see chapter 2.5.4) should contain as many different features as possible. Additionally, the unit cells in the training set should be as small as possible to minimize the computational effort.

Therefore, we construct separate sets of all structures with one, two and three molecules per unit cell (more molecules means larger unit cells) and investigate how many features they contain. Figure 34 shows the results.

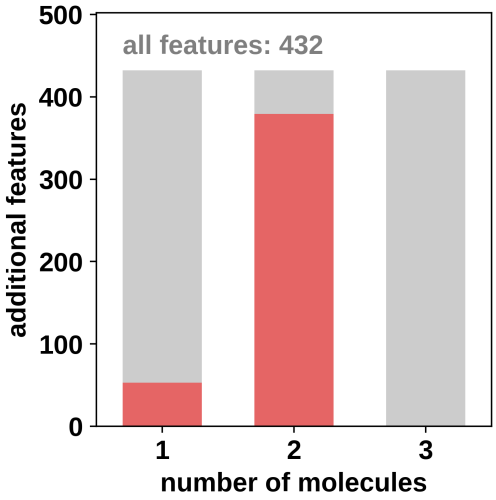


Figure 34: Number of additional features (i.e. features that are not found in structures with fewer molecules) contained in structures with different numbers of molecules; All possible features are found in structures with up to two molecules.

Figure 34 shows that structures with two molecules per unit cell already contain all possible features. This is in fact a general observation. By design, the two-molecule structures contain all combinations of two local adsorption geometries in all possible relative positions that fit the unit cells. Hence, if d_{max} is sufficiently small, or the unit cells with two molecules are sufficiently large, the two-molecule structures contain all possible molecule pairs and therefore all possible features.

3.5.4 Additional Parameters for Bayesian Linear Regression

Bayesian linear regression requires a number of additional parameters, listed in table 5. With exception of the feature decay length, we endow these parameters with physically motivated settings. For the feature decay length, we perform convergence tests (see appendix E).

Table 5: Parameter settings for Bayesian linear regression

parameter	symbol	setting
interaction uncertainty	σ_0	100 meV
DFT-model uncertainty	γ	5 meV
feature decay length	α	10
real space decay length	τ	5 Å

3.5.5 Validation with Test System

Before predicting energies of the real system, we validate the Bayesian linear regression model, its parameters and the feature vector. To this end, we use the calculations from the test system, discussed in chapter 3.5.1. We D-optimally select a number of structures as training data, predict the adsorption energies for the remaining structures and compare the predictions to the energies from DFT. Figure 35 shows the results.

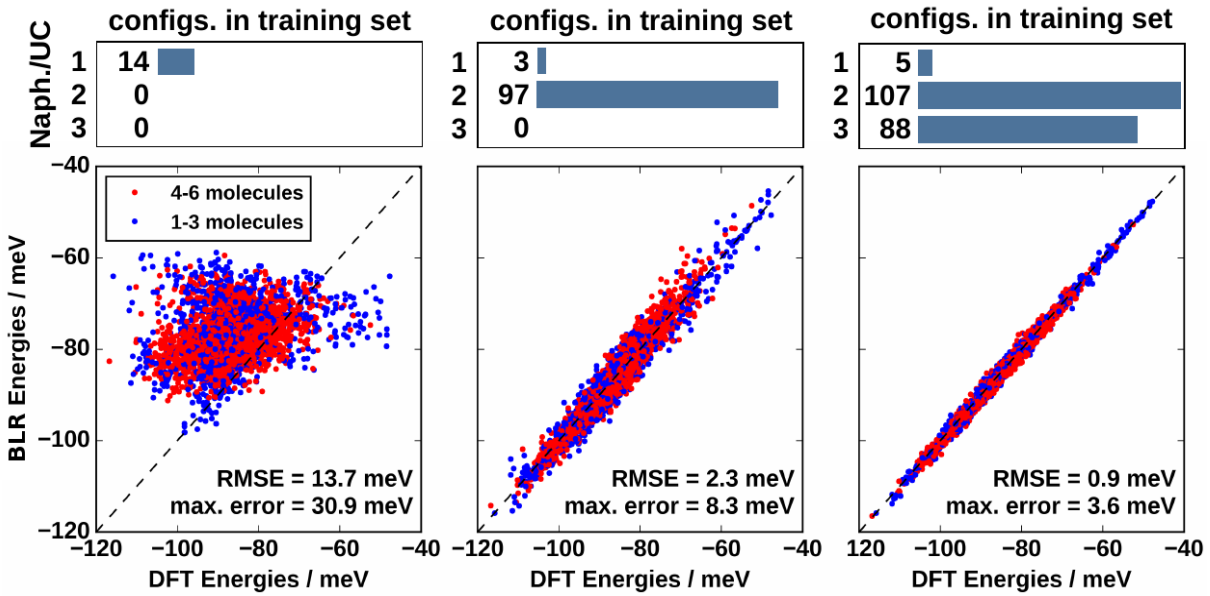


Figure 35: BLR energies (Bayesian linear regression) versus DFT energies for different numbers of training points, whereby the energies are the adsorption energies per adsorbate molecule

Figure 35 shows, that 100 - 200 training points are sufficient to predict the adsorption energies of the entire test system with a high degree of accuracy. We note that the DFT uncertainty of about 20 meV per molecule is much larger than the maximal error of 3.6 meV, which we are able to attain after training on 200 data points.

3.6 Predicted Structures

3.6.1 Validation

Before predicting a phase diagram of naphthalene on Cu(111), we validate our prediction. To this end, we D-optimally select 261 structures from a set of 646284 structures with one to three molecules per unit cell. We calculate the adsorption energies of the training set and use them to train the energy model (see chapter 2.5.1). This allows us to predict the adsorption energies of all 646284 structures, shown in figure 36. To validate the prediction, we select 44 low-energy structures (including the energetically most favorable structure), that are not part of the training set. After calculating the adsorption energies with DFT, we compare them to the predictions. Figure 36 shows the results.

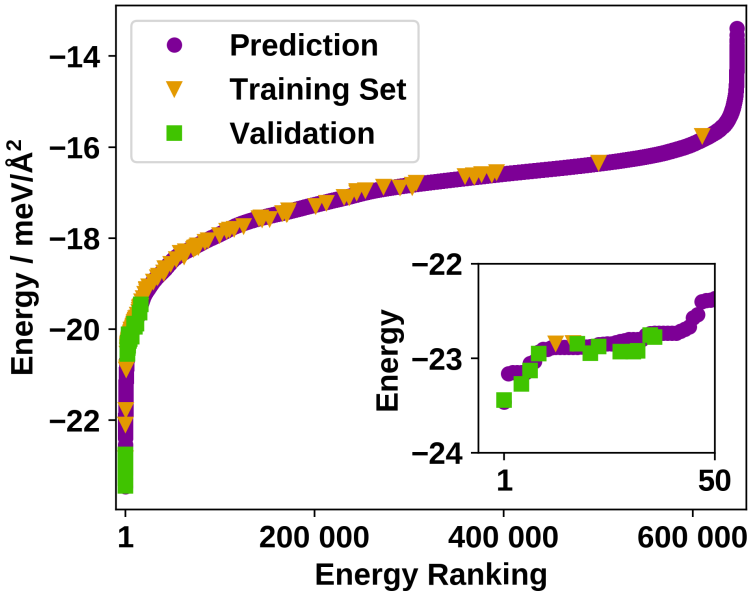


Figure 36: Adsorption energies per area for naphthalene structures on Cu(111): We predict the energies of 646284 structures with 1 thru 4 molecules per unit cell and validate the prediction with 44 validation calculations. The training set comprises 261 structures.

We find a root-mean-square error of 0.07 $meV/Å^2$, which is smaller than the DFT uncertainty of about 0.22 $meV/Å^2$. Further, this error is smaller than the thermal energy scale 1 $k_B T$, which corresponds to about 0.28 $meV/Å^2$ for a coverage of 16 Cu-atoms per naphthalene at room temperature.

After validation, the ranking of the energetically most favorable structures remains largely the same, whereby the best structure stays in first place. Therefore, we are confident in the veracity of our prediction.

3.6.2 Two-Body Interactions

The energy model, discussed in chapter 2.5.1 allows us to glean some physical insight into the adsorption of naphthalene on Cu(111). Especially the one- and two-body interactions contain important information about the system and can help to explain the reasons for the occurrence of certain structures. In chapter 3.5.2 we already briefly considered the two-body or pair interactions. We note, that the interaction map in chapter 3.5.2 stems directly from DFT calculations of a dimer in vacuum. Here we present the two-body interactions, which are fit parameters of the Bayesian linear regression model:

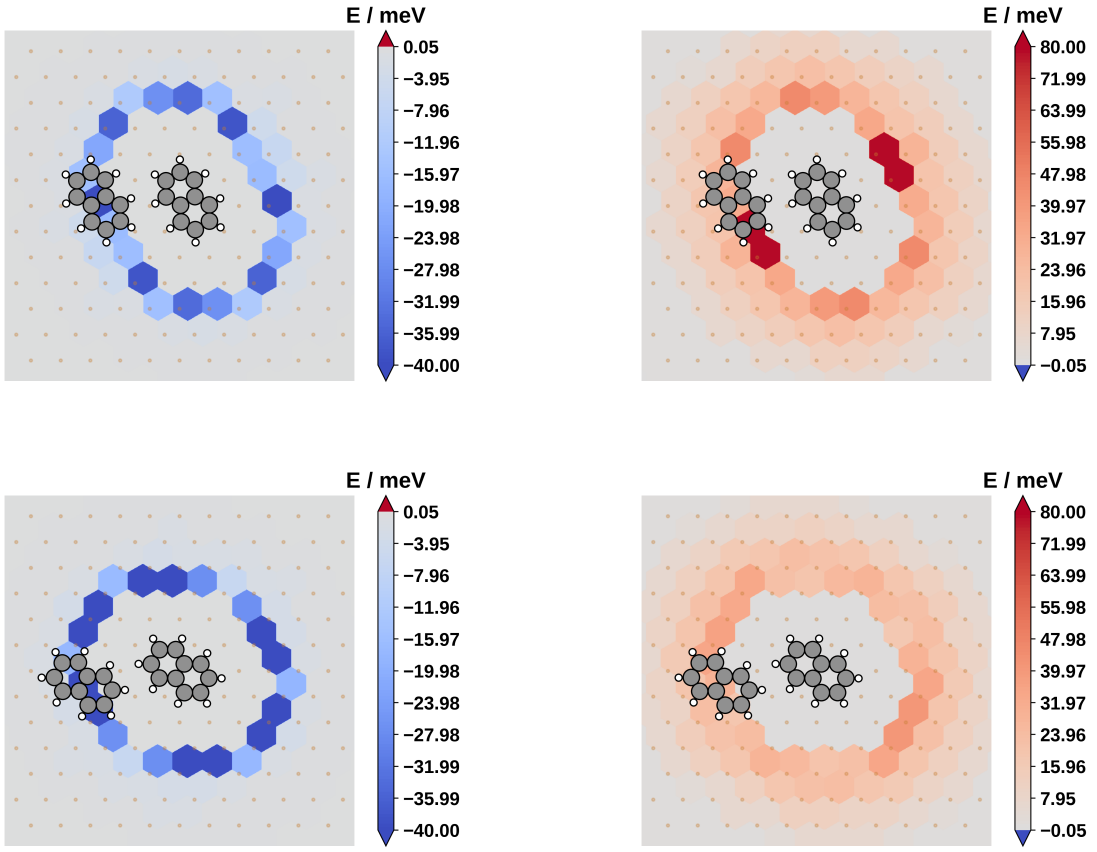


Figure 37: Example of two-body interactions:
Left: naphthalene layer in vacuum
Right: naphthalene on Cu(111)

Figure 37 shows the two-body interactions of two different pairs of local adsorption geometries. The left-hand interaction maps stem from a naphthalene layer in vacuum and the right-hand maps stem form naphthalene on the substrate. Similar to the dimer interaction from chapter 3.5.2 we observe that in vacuum the interactions are attractive. On the substrate however, the interactions of geometrically identical pairs are repulsive. We attribute the attractive interactions in vacuum to van der Waals interactions. Further, we believe that the repulsive interactions on the substrate mainly originate from electrostatic repulsion of the positively charged naphthalene molecules.

However, we do not expect such an inversion of the interaction energies in general. Strongly interacting systems, such as benzoquinone[31] exhibit hydrogen bonds, whose interactions retain their sign and magnitude when adsorbed on a substrate.

3.6.3 Predicted Phase Diagram

As stated in chapter 2.6, phase diagrams allow to consider the system in experimentally accessible conditions. For adsorbate layers on substrates the measure of interest is the Gibbs free energy of adsorption. Using the predicted adsorption energies, we calculate the Gibbs free energy of adsorption according to chapter 2.6. Figure 38 shows the phase diagram of structures with 1 to 4 molecules per unit cell. We note, that a number of assumptions (see chapter 2.1) enter into the prediction of phase diagrams, which are therefore of a more qualitative than quantitative nature.

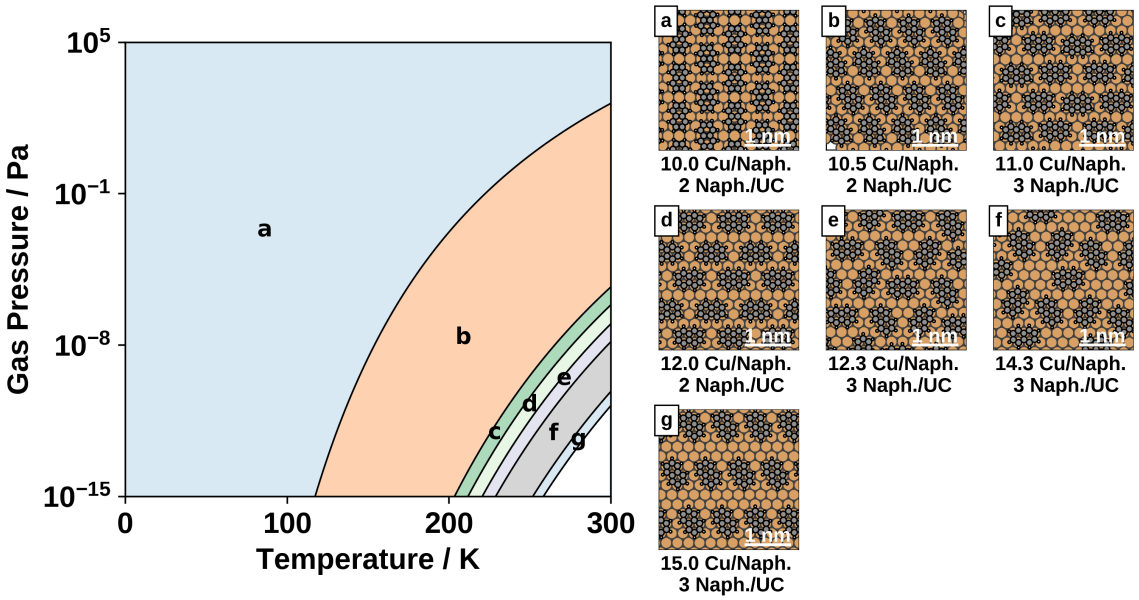


Figure 38: Phase diagram for structure with one to four molecules per unit cell

We observe seven different phases with a coverage range of 10 - 15 Cu-atoms per naphthalene molecule and 2 - 3 molecules per unit cell. The energetically most favorable structure has two molecules per unit cell and a coverage of 10 Cu-atoms per naphthalene molecule. This structure also forms the phase with the highest packing density. Further, we find that the adsorption of naphthalene molecules is no longer energetically favorable, if the temperature rises roughly above 300 K and the pressure drops below 10^{-15} Pa.

A comparison of the experimental structures (with one molecule per unit cell) and the energetically most favorable structures of similar coverage (see table 6) reveals that the experimental structures are predicted slightly higher in energy.

Table 6: Comparison between the predicted adsorption energies of the experimental and the best predicted structures of the same coverage

coverage	**best structure**	**experimental structure**
12	$(-20.5 \pm 0.3)\ meVAA^2$	$(-19.2 \pm 0.3)\ meV/\text{Å}^2$
15	$(-16.8 \pm 0.3)\ meVAA^2$	$(-16.1 \pm 0.3)\ meV/\text{Å}^2$

Further, the prediction tilts towards more densely packed structures. For instance, at 120 K the experimental coverage is 12 Cu-atoms per naphthalene, while the predicted coverage is about 10 Cu-atoms per naphthalene. We note, that the pressure indicated in figure 38 is the partial pressure of the molecules in gas phase, which is difficult to attain experimentally. We can identify three possible reasons for this discrepancy:

- On the one hand, the discrepancy may result from the simplifications we employ in calculating the Gibbs free energy of adsorption. Particularly our neglection of the vibrational enthalpy may account for the tendency towards more densely packed structures in the prediction.
- On the other hand, uncertainties of the DFT calculations, we use as training data for our energy model, may cause such a discrepancy. However, such a discussion is beyond the scope of this thesis.
- A further possibility is that the experimental structures are kinetically trapped, which is possible since the experiments were not done in thermodynamic equilibrium.

4 Conclusion and Outlook

In this thesis I demonstrated my contribution to the development of SAMPLE, a powerful tool for exhaustive surface structure search.

First, I was able the overcome the configurational explosion by coarse-graining the potential energy surface. I employed a discretization consisting of local adsorption geometries and unit cells, the combination of which allowing me to generate all possible structures within the configuration space of the system at hand.

Secondly, I used optimal design theory (D-optimality criterion) to select a subset of these structures as a training set and calculated their adsorption energies using dispersion corrected density functional theory. This procedure allowed me to maximize the information gain, while keeping the computational cost at a minimum.

Thirdly, I fitted an energy model, using a Bayesian linear regression algorithm to extract the one- and two-body interactions from the training set. This model allowed me to predict the energies of all structures. To validate the prediction, I calculated the adsorption energies of a number of structures, which were not included in the training set, with DFT.

Finally, I used ab initio thermodynamics to predict phase diagrams.

I successfully applied SAMPLE to the system of naphthalene on Cu(111) and was able to predict the most energetically favorable closed packed structures. The comparison between prediction and validation revealed a prediction error smaller than the DFT uncertainty. Comparison with experimental phases of this system showed a discrepancy between prediction and experiment. Reasons for this discrepancy may be simplifications in the calculation of the phase diagram, uncertainties inherent in the DFT calculations or kinetic trapping of the experimental structures.

Faced with these challenges, a mayor goal is to include forces and vibration frequencies in SAMPLEs energy model. The capability to learn and predict these properties of a structure would allow SAMPLE to calculate the vibrational enthalpy and make it possible to estimate kinetic barriers.

List of Figures

List of Tables

References

[1] John Maddox. "Crystals from first principles". In: *Nature* 335 (Sept. 1988), p. 201.

[2] "Front Matter". In: *Modern Methods of Crystal Structure Prediction*. Wiley-VCH Verlag GmbH & Co. KGaA, 2010, pp. I–X. ISBN: 9783527632831.

[3] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. ISBN: 0-262-18253-X.

[4] Andriy O. Lyakhov, Artem R. Oganov, and Mario Valle. "Crystal Structure Prediction Using Evolutionary Approach". In: *Modern Methods of Crystal Structure Prediction*. Wiley-VCH Verlag GmbH & Co. KGaA, 2010, pp. 147–180. ISBN: 9783527632831.

[5] David J. Wales. "Energy Landscapes and Structure Prediction Using Basin-Hopping". In: *Modern Methods of Crystal Structure Prediction*. Wiley-VCH Verlag GmbH & Co. KGaA, 2010, pp. 29–54. ISBN: 9783527632831.

[6] *Polymorphism in Molecular Crystals*. Oxford University Press, 2002.

[7] Andrew O. F. Jones et al. "Substrate-Induced and Thin-Film Phases: Polymorphism of Organic Materials on Surfaces". In: *Advanced Functional Materials* 26.14 (2016), pp. 2233–2255. ISSN: 1616-3028.

[8] Joel Bernstein. "Polymorphism - a perspective". In: *Crystal Growth & Design* 11.3 (2011), pp. 632–650.

[9] Jonas Nyman and Graeme M Day. "Static and lattice vibrational energy differences between polymorphs". In: *CrystEngComm* 17.28 (2015), pp. 5154–5165.

[10] Wilhelm Ostwald. "Studien über die Bildung und Umwandlung fester Körper". In: *Zeitschrift für Physikalische Chemie* 22 (1987), 289–330.

[11] Daniel Käfer, Lars Ruppel, and Gregor Witte. "Growth of pentacene on clean and modified gold surfaces". In: *Phys. Rev. B* 75 (8 2007), p. 085309.

[12] Norbert Koch et al. "Adsorption-Induced Intramolecular Dipole: Correlating Molecular Conformation and Interface Electronic Structure". In: *Journal of the American Chemical Society* 130.23 (2008). PMID: 18479108, pp. 7300–7304.

[13] Wi Hyoung Lee et al. "Surface-Directed Molecular Assembly of Pentacene on Monolayer Graphene for High-Performance Organic Transistors". In: *Journal of the American Chemical Society* 133.12 (2011). PMID: 21381751, pp. 4447–4454.

[14] Christine C. Mattheus et al. "Identification of polymorphs of pentacene". In: *Synthetic Metals* 138.3 (2003), pp. 475 –481. ISSN: 0379-6779.

[15] B. Stadlober et al. "Growth model of pentacene on inorganic and organic dielectrics based on scaling and rate-equation theory". In: *Phys. Rev. B* 74 (16 2006), p. 165302.

[16] S. Nishikata et al. "Polycrystalline domain structure of pentacene thin films epitaxially grown on a hydrogen-terminated Si(111) surface". In: *Phys. Rev. B* 76 (16 2007), p. 165424.

[17] Hwa Sung Lee et al. “Effect of the Phase States of Self-Assembled Monolayers on Pentacene Growth and Thin-Film Transistor Characteristics”. In: *Journal of the American Chemical Society* 130.32 (2008). PMID: 18630908, pp. 10556–10564.

[18] C. D. Dimitrakopoulos, A. R. Brown, and A. Pomp. “Molecular beam deposited thin films of pentacene for organic field effect transistor applications”. In: *Journal of Applied Physics* 80.4 (1996), pp. 2501–2508.

[19] Sandra E. Fritz et al. “Structural Characterization of a Pentacene Monolayer on an Amorphous SiO2 Substrate with Grazing Incidence X-ray Diffraction”. In: *Journal of the American Chemical Society* 126.13 (2004). PMID: 15053579, pp. 4084–4085.

[20] Stefan C. B. Mannsfeld et al. “Precise Structure of Pentacene Monolayers on Amorphous Silicon Oxide and Relation to Charge Transport”. In: *Advanced Materials* 21.22 (2009), pp. 2294–2298. ISSN: 1521-4095.

[21] D. E. Hooks, T. Fritz, and M. D. Ward. “Epitaxy and Molecular Organization on Solid Substrates”. In: *Advanced Materials* 13.4 (), pp. 227–241.

[22] M. Born and R. Oppenheimer. “Zur Quantentheorie der Molekeln”. In: *Annalen der Physik* 389.20 (1927), pp. 457–484. ISSN: 1521-3889.

[23] P. Hohenberg and W. Kohn. “Inhomogeneous Electron Gas”. In: *Phys. Rev.* 136 (3B 1964), B864–B871.

[24] W. Kohn and L. J. Sham. “Self-Consistent Equations Including Exchange and Correlation Effects”. In: *Phys. Rev.* 140 (4A 1965), A1133–A1138.

[25] K. Kim and K. D. Jordan. “Comparison of Density Functional and MP2 Calculations on the Water Monomer and Dimer”. In: *The Journal of Physical Chemistry* 98.40 (1994), pp. 10089–10094.

[26] P. J. Stephens et al. “Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields”. In: *J. Phys. Chem.* 98.45 (1994), pp. 11623–11627.

[27] Jutta Rogal and Karsten Reuter. “Ab Initio Atomistic Thermodynamics for Surfaces: A Primer”. In: *Experiment, Modeling and Simulation of Gas-Surface Interactions for Reactive Flows in Hypersonic Flights*. RTO/NATO, 2007, pp. 2–1 –2–18. ISBN: 978-92-837-0057-9.

[28] Veronika Obersteiner et al. “Structure Prediction for Surface-Induced Phases of Organic Monolayers: Overcoming the Combinatorial Bottleneck”. In: *Nano Letters* 17.7 (2017). PMID: 28640634, pp. 4453–4460.

[29] Michael Scherbela et al. “Charting the energy landscape of metal/organic interfaces via machine learning”. In: *Phys. Rev. Materials* 2 (4 2018), p. 043803.

[30] Volker Blum et al. “Ab initio molecular simulations with numeric atom-centered orbitals”. In: *Computer Physics Communications* 180.11 (2009), pp. 2175 –2196. ISSN: 0010-4655.

[31] Andreas Jeindl. *Influence of Molecule Size on Surface Polymorph Formation: An ab-initio Study with Machine Learning*. Graz University of Technology, 2018.

[32] Takashi Yamada et al. “Novel Growth of Naphthalene Overlayer on Cu(111) Studied by STM, LEED, and 2PPE”. In: *The Journal of Physical Chemistry C* 114.31 (2010), pp. 13334–13339.

[33] Roman Forker et al. “The Complex Polymorphism and Thermodynamic Behavior of a Seemingly Simple System: Naphthalene on Cu(111)”. In: *Langmuir* 30.47 (2014). PMID: 25361739, pp. 14163–14170.

[34] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. “Generalized Gradient Approximation Made Simple”. In: *Phys. Rev. Lett.* 77 (18 1996), pp. 3865–3868.

[35] Victor G. Ruiz et al. “Density-Functional Theory with Screened van der Waals Interactions for the Modeling of Hybrid Inorganic-Organic Systems”. In: *Phys. Rev. Lett.* 108 (14 2012), p. 146103.

Appendix A Conditioning the Multivariate Gaussian

We start with equation 8.

$$p(\mathbf{x_1}, \mathbf{x_2}|\boldsymbol{\mu}, C) \;\propto\; \exp\left[-\frac{1}{2}\begin{pmatrix}\mathbf{x_1}-\boldsymbol{\mu_1}\\ \mathbf{x_2}-\boldsymbol{\mu_2}\end{pmatrix}^T \begin{pmatrix}C_{11} & C_{12}\\ C_{21} & C_{22}\end{pmatrix}^{-1} \begin{pmatrix}\mathbf{x_1}-\boldsymbol{\mu_1}\\ \mathbf{x_2}-\boldsymbol{\mu_2}\end{pmatrix}\right] \tag{82}$$

Let us first deal with inverting the covariance matrix. Since the covariance matrix is a symmetric block matrix, its inverse also has symmetric block from. For the different inverse blocks we will use the shorthand notation given below.

$$\begin{pmatrix}C_{11} & C_{12}\\ C_{21} & C_{22}\end{pmatrix}^{-1} \;=\; \begin{pmatrix}A & B\\ B^T & D\end{pmatrix} \tag{83}$$

The expanded expression for the inverse block is given below. Note, that $A = A^T$ and $D = D^T$ due to the symmetry of the covariance matrix.

$$\begin{aligned} A &= (C_{11} - C_{12}C_{22}^{-1}C_{21})^{-1} \\ B &= -(C_{11} - C_{12}C_{22}^{-1}C_{21})^{-1}C_{12}C_{22}^{-1} \;=\; -A\widetilde{B} \\ D &= C_{22}^{-1} + C_{22}^{-1}C_{21}(C_{11} - C_{12}C_{22}^{-1}C_{21})^{-1}C_{12}C_{22}^{-1} \;=\; C_{22}^{-1} + \widetilde{B}^T A\widetilde{B} \end{aligned} \tag{84}$$

We substitute the inverse of the covariance matrix by the simplified expression given in equation 83 and use it to solve the matrix multiplication in equation 8.

$$\begin{aligned} p(\mathbf{x_1}, \mathbf{x_2}|\boldsymbol{\mu}, C) \;\propto\; \exp\Big[-\frac{1}{2}\Big[&(\mathbf{x_1}-\boldsymbol{\mu_1})^T A(\mathbf{x_1}-\boldsymbol{\mu_1}) \\ &+(\mathbf{x_1}-\boldsymbol{\mu_1})^T B(\mathbf{x_2}-\boldsymbol{\mu_2}) \\ &+(\mathbf{x_2}-\boldsymbol{\mu_2})^T B^T(\mathbf{x_1}-\boldsymbol{\mu_1}) \\ &+(\mathbf{x_2}-\boldsymbol{\mu_2})^T D(\mathbf{x_2}-\boldsymbol{\mu_2})\Big]\Big] \end{aligned}$$

Now we substitute the B and C with their expression in equation 84.

$$\begin{aligned} p(\mathbf{x_1}, \mathbf{x_2}|\boldsymbol{\mu}, C) \;\propto\; \exp\Big[-\frac{1}{2}\Big[&(\mathbf{x_1}-\boldsymbol{\mu_1})^T A(\mathbf{x_1}-\boldsymbol{\mu_1}) \\ &+(\mathbf{x_1}-\boldsymbol{\mu_1})^T(-A\widetilde{B})(\mathbf{x_2}-\boldsymbol{\mu_2}) \\ &+(\mathbf{x_2}-\boldsymbol{\mu_2})^T(-A\widetilde{B})^T(\mathbf{x_1}-\boldsymbol{\mu_1}) \\ &+(\mathbf{x_2}-\boldsymbol{\mu_2})^T(C_{22}^{-1} + \widetilde{B}^T A\widetilde{B})(\mathbf{x_2}-\boldsymbol{\mu_2})\Big]\Big] \end{aligned}$$

Looking closely at the last term on the above equation, we can factor out C_{22}^{-1}. We further solve some brackets whereby we use $A = A^T$.

$$\begin{aligned} p(\mathbf{x_1}, \mathbf{x_2}|\boldsymbol{\mu}, C) \;\propto\; \exp\Big[-\frac{1}{2}\Big[&(\mathbf{x_1}-\boldsymbol{\mu_1})^T A(\mathbf{x_1}-\boldsymbol{\mu_1}) \\ &-(\mathbf{x_1}-\boldsymbol{\mu_1})^T A\widetilde{B}(\mathbf{x_2}-\boldsymbol{\mu_2}) \\ &-(\mathbf{x_2}-\boldsymbol{\mu_2})^T \widetilde{B}^T A(\mathbf{x_1}-\boldsymbol{\mu_1}) \\ &+(\mathbf{x_2}-\boldsymbol{\mu_2})^T \widetilde{B}^T A\widetilde{B}(\mathbf{x_2}-\boldsymbol{\mu_2})\Big]\Big] \\ &\cdot \exp\left[-\frac{1}{2}(\mathbf{x_2}-\boldsymbol{\mu_2})^T C_{22}^{-1}(\mathbf{x_2}-\boldsymbol{\mu_2})\right] \end{aligned}$$

Now we rewrite the argument of the first exponential as a matrix multiplication. If this step is not obvious, it can easily be verified by solving the matrix multiplication.

$$\begin{aligned} p(\mathbf{x_1}, \mathbf{x_2}|\boldsymbol{\mu}, C) \;\propto\; &\exp\Big[-\frac{1}{2}\Big((\mathbf{x_1}-\boldsymbol{\mu_1})^T - (\mathbf{x_2}-\boldsymbol{\mu_2})^T\widetilde{B}^T\Big) A\Big((\mathbf{x_1}-\boldsymbol{\mu_1}) - \widetilde{B}(\mathbf{x_2}-\boldsymbol{\mu_2})\Big)\Big] \\ &\cdot \exp\left[-\frac{1}{2}(\mathbf{x_2}-\boldsymbol{\mu_2})^T C_{22}^{-1}(\mathbf{x_2}-\boldsymbol{\mu_2})\right] \end{aligned}$$

We simplify,

$$\begin{aligned} p(\mathbf{x_1}, \mathbf{x_2}|\boldsymbol{\mu}, C) \;\propto\; &\exp\Big[-\frac{1}{2}\Big(\mathbf{x_1}-\boldsymbol{\mu_1}-\widetilde{B}(\mathbf{x_2}-\boldsymbol{\mu_2})\Big)^T A\Big(\mathbf{x_1}-\boldsymbol{\mu_1}-\widetilde{B}(\mathbf{x_2}-\boldsymbol{\mu_2})\Big)\Big] \\ &\cdot \exp\left[-\frac{1}{2}(\mathbf{x_2}-\boldsymbol{\mu_2})^T C_{22}^{-1}(\mathbf{x_2}-\boldsymbol{\mu_2})\right] \end{aligned}$$

and finally find, that equation 8 can be rewritten in the following way.

$$p(\mathbf{x_1}, \mathbf{x_2}|\boldsymbol{\mu}, C) \;\propto\; \exp\left[-\frac{1}{2}(\mathbf{x_1}-\bar{\boldsymbol{\mu}})^T A(\mathbf{x_1}-\bar{\boldsymbol{\mu}})\right] \cdot \exp\left[-\frac{1}{2}(\mathbf{x_2}-\boldsymbol{\mu_2})^T C_{22}^{-1}(\mathbf{x_2}-\boldsymbol{\mu_2})\right] \tag{85}$$

Appendix B Chemical Potential of the Ideal Gas

The Free energy is related to the partition function in the following way:

$$F = -kT\ln(Z) \tag{86}$$

The derivative of the Free energy with regard to the number of particles provides us with the chemical potential.

$$\mu = \frac{\partial F}{\partial N} \tag{87}$$

To calculate the chemical potential, we start with the Hamiltonian for similar, non-relativistic, classic particles.

$$H = \sum_{i=1}^{3N} \frac{p_i^2}{2m} + V(x_i) \tag{88}$$

Since we are considering an ideal gas, the particles do not interact with each other. Hence, the potential becomes $V(x_i) = 0$. Now we can calculate the partition function.

$$Z = \frac{1}{N!\ h^{3N}} \int \exp\left(\frac{H}{kT}\right) d^{3N}p d^{3N}x \tag{89}$$

$$Z = \frac{1}{N!\ h^{3N}} \int \exp\left(\frac{1}{kT}\left(\sum_{i=1}^{3N} \frac{p_i^2}{2m}\right)\right) d^{3N}p\ d^{3N}x \tag{90}$$

$$Z = \frac{1}{N!\ h^{3N}} \int \left[\prod_{i=1}^{3N} \exp\left(\frac{1}{kT}\left(\frac{p_i^2}{2m}\right)\right)\right] d^{3N}p\ d^{3N}x \tag{91}$$

$$Z = \frac{1}{N!\ h^{3N}} \prod_{i=1}^{3N} \left[\int \exp\left(\frac{1}{kT}\left(\frac{p_i^2}{2m}\right)\right) dp_i\ dx_i\right] \tag{92}$$

$$Z = \frac{1}{N!\ h^{3N}} \prod_{i=1}^{N} V(2\pi mkT)^{3/2} \tag{93}$$

$$Z = \frac{V^N}{N!}\left(\frac{2\pi mkT}{h^2}\right)^{3N/2} \tag{94}$$

If we define the thermal de Broglie wavelength as $\lambda = \frac{h}{\sqrt{2\pi mkT}}$ we can simplify the partition function.

$$Z = \frac{V^N}{N!}\left(\frac{1}{\lambda}\right)^{3N} \tag{95}$$

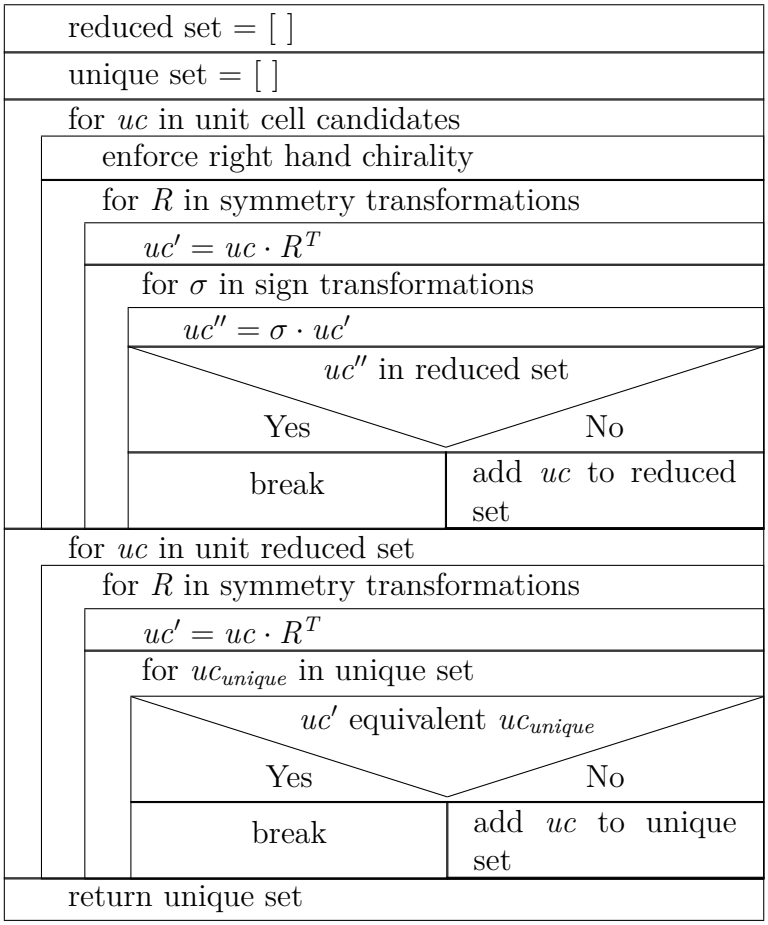
Now we can employ equations (86) and (87) to calculate the chemical potential of an ideal gas.

$$F = kT\left(\ln(N!) - N\ln\left(\frac{V}{\lambda^3}\right)\right) \tag{96}$$

$$\mu = kT\ln\left(\frac{\lambda^3}{VN}\right) = kT\ln\left(\frac{p\lambda^3}{kT}\right) \tag{97}$$

Appendix C Unit Cell Symmetry Reduction

Here, we present the code structure of the symmetry reduction algorithm.



Appendix D DFT Convergence Tests

D.1 Number of Slab Layers

The number of substrate layers is converged, using the adsorption energy of a singe naphthalene molecule in a $(4, 0, 0, 4)$ unit cell. The k-grid is set to 20 k-points in the x- and y-direction of the reciprocal unit cell and 1 k-point in the z-direction.

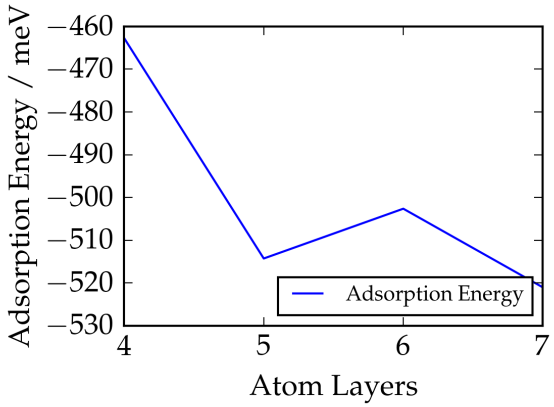


Figure 39: Layer convergence for slab

D.2 Lattice Constant

The lattice constant is converged for a primitive ffc-bulk unit cell with a k-grid containing 80 k-points in all three fractional dimensions of the reciprocal unit cell.

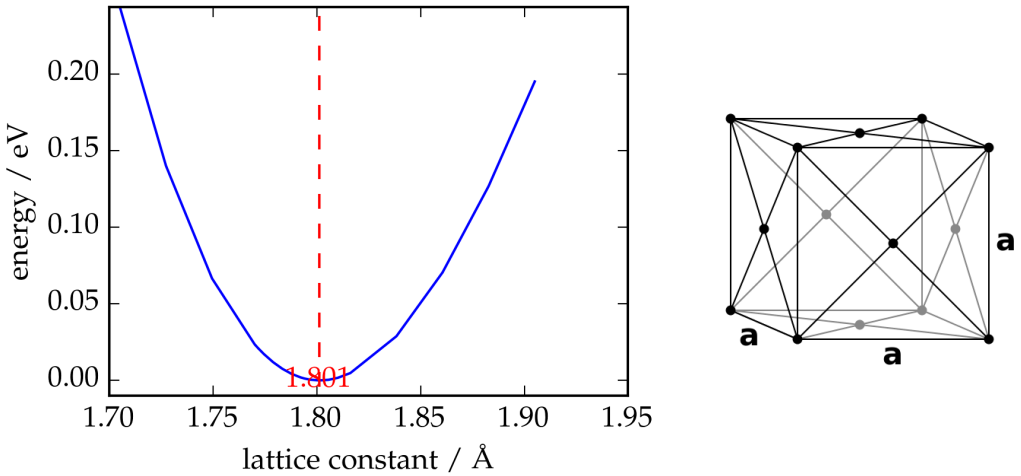


Figure 40: Lattice constant convergence:
The lattice constant in figure 42 is equivalent to $a/2$.

D.3 k-Grid

The k-grid is converged for a primitive ffc-bulk unit cell with a lattice constant of $a = 3.61$Å.

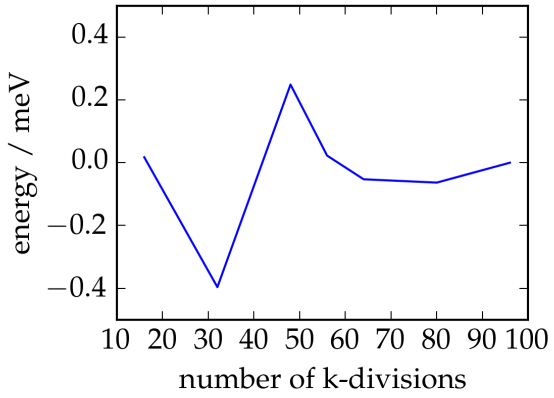


Figure 41: k-point convergence

D.4 Cutpot

The parameter cutpot is converged for a primitive ffc-bulk unit cell with a k-grid containing 80 k-points in all three fractional dimensions of the reciprocal unit cell.

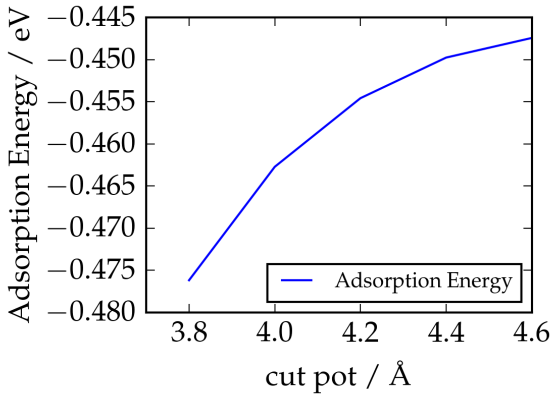


Figure 42: Cutpot convergence

By setting the parameter cutpot to 4.6Å the calculation time is increased by +4%.

D.5 Removal of the 5g Basis Function for Cu

Removing the 5g basis function is tested, using the adsorption energy of a singe naphthalene molecule in a $(4, 0, 0, 4)$ unit cell. The k-grid is set to 20 k-points in the x- and y-direction of the reciprocal unit cell and 1 k-point in the z-direction.

Adsorption Energy (tier1 tight)	$-0.46273\ eV$
Adsorption Energy (tier1 tight without 5g)	$-0.47816\ eV$
difference	$0.01543\ eV$

The reduction in calculation time is found to be -54%.

D.6 Radial Multiplier

Reducing the radial multiplier is tested, using the adsorption energy of a singe naphthalene molecule in a $(4, 0, 0, 4)$ unit cell. The k-grid is set to 20 k-points in the x- and y-direction of the reciprocal unit cell and 1 k-point in the z-direction.

Adsorption Energy (Multiplier set to 2)	$-0.46273\ eV$
Adsorption Energy (Multiplier set to 1)	$-0.46406\ eV$
difference	$0.00133\ eV$

Reducing the radial multiplier to 1, reduces the calculation time by -16%.

Appendix E Bayesian Linear Regression Convergence Tests

E.1 Feature Decay Length Convergence

The decay length σ is converged for the test system using 100 structures as training set. The RMSE is calculated from the comparison with 4122 structures. For the remaining parameters of the Bayesian linear regression model we use the values from chapter 3.5.3.

For the naphthalene structures on the Cu(111) substrate we converge the decay length σ using 189 structures as training set. The RMSE is calculated from the comparison with 20 structures.

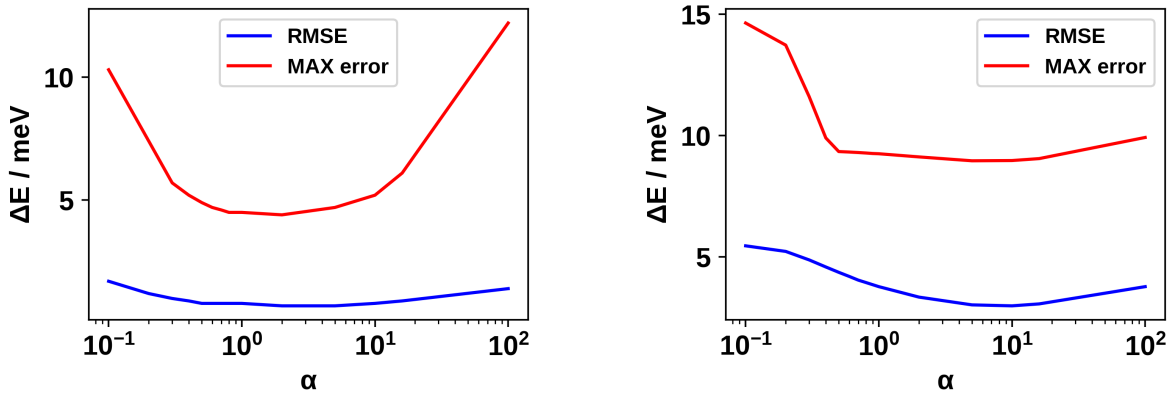


Figure 43: Left: Feature decay length convergence for naphthalene layer in vacuum
Right: Feature decay length convergence for naphthalene on the substrate

E.2 RMSE over Number of Training Points

We calculate the root mean square error (RMSE) for the test system, using the settings from chapter 3.5.3. The RMSE is calculated from the comparison with all 4122 structures in the test system.

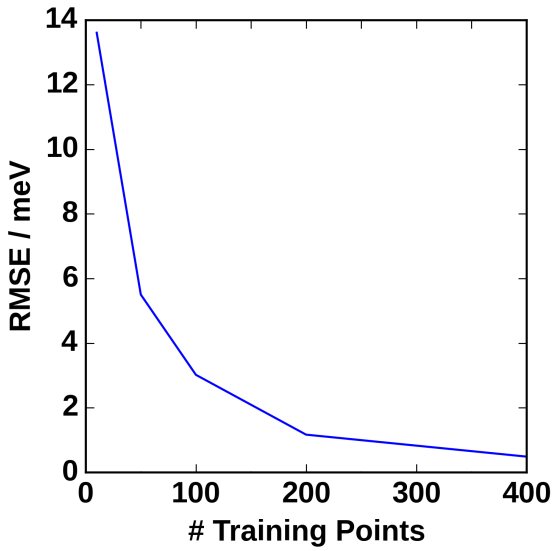


Figure 44: Naphthalene layer in vacuum:
root mean square error (RMSE) for different numbers of training data