Michael Scherbela, BSc.

Structure Prediction at Organic/Inorganic Interfaces using Machine 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

Institut für Festkörperphysik

Graz, April 2017

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

Acknowledgments

I want to thank the people that helped me do the research outlined in this thesis. First and foremost I owe my deepest gratitude to Oliver Hofmann for supervising my work. He gave me the freedom to explore ideas and methods that were new to both of us, while still ensuring I would not stray afar from the path towards a finished thesis.

Little can be achieved by one person alone, so I thank my coworkers Veronika, Lukas and Andi. Veronika introduced me to the topic, while Lukas and Andi partly helped to build the code that is at the heart of this thesis.

Furthermore I want to thank all the members of the group - including Thomas, Gernot and Christian - that made working on my master-thesis such an enjoyable time. I will miss the many discussions and good times that we shared at university and Café Zapo.

For the many CPU-hours that went into this project I want to thank the FWF for their financial support, the Vienna Scientific Cluster 3, its brave supporting team and its tall representative in our corridor.

My family has supported me through all my studies and has always been a safe haven, for which I am very grateful.

In the end I want to thank the person that has gone with me through all the years and has shared my ups and downs: Thank you Sonja!

Abstract

Structure Prediction at Organic/Inorganic Interfaces using Machine Learning

Michael Scherbela
Institute of Solid State Physics, Graz University of Technology

Many properties of thin films, such as their solubility or conductivity, depend strongly on the crystal structure of the adsorbed molecules. A major step towards designing new materials is therefore understanding and predicting the crystal structures that form at interfaces.

This could in principle be done by finding low energy structures using ab-initio methods. However, the many degrees of freedom lead to a rich polymorphism that prohibits an exhaustive search for the global minimum using expensive ab-initio calculations. In this work it is shown on the example of tetracyanoethylene (TCNE) on coinage metal surfaces that this challenge can be tackled with a combination of coarse-graining and machine learning:

First the adsorption geometries that isolated molecules would adopt on the substrate are found. Supercells with multiple molecules per unit cell are built by combining these isolated adsorption geometries to generate a set of possible “guess polymorphs”. This discretizes the configurational space to a huge, but finite size. Using optimal design methods a small, representative subset is selected and its energies calculated using Density Functional Theory (DFT). By training a Gaussian Process Regression model on these first-principle calculations, effective interactions between adsorbates are obtained. This provides an efficient and accurate energy prediction for all remaining guess polymorphs which is exploited by sampling the energetically most promising structures and iterated relearning.

Using this method we explore the potential energy landscape of TCNE on Cu(111) and Ag(100) and show that the machine learning model can achieve the accuracy of the expensive DFT calculations after having been trained only on a small fraction of the total dataset.

Kurzfassung

Struktursuche an Organisch-/Inorganischen Grenzflächen mittels Machine Learning

Michael Scherbela
Institut für Festkörperphysik, Technische Universität Graz

Viele Eigenschaften von Materialien von organischen Dünnschichten hängen stark von der Kristallstruktur der adsorbierten Moleküle ab. Um neue Materialien mit gewünschten Eigenschaften zu designen ist es daher von großem Vorteil die Struktur ohne Experimente - lediglich durch theoretische Berechnungen - bestimmen zu können.

Im Prinzip könnten thermodynamisch stabile Phasen durch Suche von niederenergetischen Strukturen mittels ab-initio Rechnungen bestimmt werden. In der Praxis dagegen führen die vielen Freiheitsgrade zu einer riesigen Vielzahl an möglichen Polymorphen die eine vollständige Suche nach dem globalen Minimum mittels teurer ab-initio Rechnungen unmöglich macht. In dieser Arbeit wird an Hand des Beispiels von Tetracyanoethen (TCNE) auf Münzmetalloberflächen gezeigt, dass sich dieses Konfigurationsproblem durch eine Kombination aus Diskretisierung und Machine Learning Methoden lösen lässt:

Zuerst werden die Adsorptionsgeometrien von einzelnen adsorbierten Molekülen auf dem Substrat bestimmt. Anschließend werden diese Adsorptionsgeometrien zu Superzellen mit mehr Molekülen pro Einheitszelle kombiniert um eine Liste aller möglichen Polymorphe aufzustellen. Dadurch wird der Konfigurationsraum auf eine sehr große, aber endliche Anzahl an Strukturen diskretisiert. Mittels Optimal-Design-Methoden wird ein repräsentatives Subset an Konfigurationen ausgewählt und deren Energien mittels Dichtefunktionaltheorie (DFT) berechnet. Anschließend wird ein Gauß-Prozess-Modell mit diesen Energien trainiert um effektive Wechselwirkungen zwischen den Molekülen zu bestimmen. Diese Wechselwirkungen bilden ein effizientes und genaues Modell der Adsorptionsenergien für alle verbleibenden Polymorphe und erlaubt es kostengünstig die energetisch niedrigsten Strukturen zu finden.

Mit diesem Modell werden in dieser Arbeit die Energien für Strukturen von TCNE auf Cu(111) und Ag(100) ermittelt. Das Machine Learning Modell liefert für diese Systeme Genauigkeiten, die mit den teuren DFT Rechnungen vergleichbar sind, obwohl es lediglich auf einem kleinen Bruchteil des gesamten Datensatzes trainiert wurde.

Life is full of mysteries, yeah,
but there are answers out there
and they won’t be found
by people sitting around
looking serious
and saying isn’t life mysterious?

TIM MINCHIN

Contents

1 Introduction

1.1 Polymorphism

Thermodynamics states that a material will form the structure of lowest free energy, but in many cases several competing low energy structures exist. These competing polymorphs can either become stable at different external conditions, like pressure or temperature, or be kinetically trapped out of equilibrium phases. Changing the structure of a molecule or crystal can dramatically alter its properties: While both diamond and graphite consist purely of carbon, their different crystal structures lead to completely different optical, mechanical and electrical properties. This strong dependence of properties on the crystal structure is also true for molecular crystals: As an example, the organic semiconductor rubrene has an extraordinarily high charge carrier mobility, but exhibits this desirable property only in one of four different polymorphs [1]. Finding polymorphs is therefore a necessary step to fully understand the possible properties and applications of any material.

While polymorphism and structure search has been extensively studied for 3D materials, limited progress has been made for surface structures. However thin molecular films on substrates are of great interest, in particular for the development of organic electronics, such as OLEDs and OFETs. In particular the structure of the first monolayer of deposited material on the substrate can crucially impact the properties of the material. On the one hand, this can be a direct effect of the monolayer, such as a change in workfunction leading to possible tuning of band alignments for organic electronic devices [2]. On the other hand the first monolayer can act as a template for subsequent layers and can change their structure and orientation: Pentacene, another organic semiconductor, can form a polymorph on surfaces that is not observed in bulk samples and controlling its crystallinity and structure has a critical impact on properties like the charge carrier mobility. It is hypothesized [3] that templating by the first monolayer plays the dominant role in the formation of this surface induced phase.

1.2 Motivation behind Structure Search

To find new materials and combinations of materials that show promising properties for future applications such as organic photovoltaics or more efficient fuel cells it is crucial to guide research efforts towards promising candidate materials. A long standing goal of materials science is therefore to calculate structures and properties for a large class of materials in a high-throughput manner and automatically search the results for desirable properties. Although this has not yet been achieved, rapid progress is made in this field [4] and the continuous improvements in ab-initio methods and computational power make ever more complex systems accessible.

As outlined in Sec. 1.1, calculated properties do strongly depend on the geometric structure of the system. Therefore accurate structure prediction is the crucial first step when calculating the properties of any new material or material combination. The aim of this work is to build a structure prediction algorithm for molecules adsorbed on surfaces and test it exemplary for the adsorption of tetracyanoethylene on coinage metal surfaces.

1.3 Configurational Explosion

Structure search is the search for the configuration with lowest free energy F. A typical search procedure consists of two basic steps: At first a structure must to be proposed and then in a second step its energy needs to be evaluated. For simple systems, that depend on very few degrees of freedom, generating structures is easy: For example, the geometric structure of any diatomic molecule only depends on a single degree of freedom, the bond length d. To generate structures one could try a grid of bond distances within reasonable limits d_{min} and d_{max}. However for systems that have more degrees of freedom, for example crystal structures with several molecules per unit cell, the configurational space grows too fast to allow simple grid based searches in practice.

While low energy polymorphs could in principle be found by exhaustively calculating the energy of all possible crystalline polymorphs this is computationally unfeasible. The impossibility of brute-forcing all possible polymorphs can readily be seen when looking at the example of a crystalline 2D structure with three molecules per unit cell. Since we constrain the molecules to be on the surface, each molecule has two translational degrees of freedom and at least one rotational (around the axis perpendicular to the surface). If we roughly discretize each degree of freedom by ten steps this yields a total number of $(10 \cdot 10 \cdot 10)^3 = 10^9$ configurations for this unit cell, which is clearly far beyond what can be calculated on a quantum mechanical level. Since this configurational space grows exponentially with the number of molecules per unit cell it is imperative to limit the configurational space to the regions of interest.

1.4 Structure Search: Generating Structures

There is a wide variety of different techniques to propose structures in the configurational space, many of which are in active use, as can be seen from the submissions for the Crystal Structure Prediction Blind Test 2016. [5]. This section will give a brief outline of a few commonly used algorithms to search for low energy configurations in the parameter space.

1.4.1 Grid Search

For a sufficiently small parameter space it can be feasible to discretize each parameter and exhaustively try all parameter combinations. This approach is limited to systems where there are few degrees of freedom, because the total number of grid points grows exponentially as a function of the number of degrees of freedom.

Grid searches have a second disadvantage: If a parameter turns out to be irrelevant, all calculations that only vary this parameter generate redundant information. This can be seen in Fig. 1 for the grid search: When the grid points are projected along one of the axis, the points accumulate densely on discrete point along the remaining axis and poorly cover the lower dimensional subspace.

1.4.2 Random Search

In a random search, values for each degree of freedom are chosen at random in the hope of covering a large part of the configurational space in an unbiased way. If values for each degree of freedom are chosen independently, also any lower dimensional projection will yield randomly distributed points, in contrast to the grid approach. For the crystal structure prediction blind test in [5], several groups preoptimized molecular geometries and then used random translations and rotations of these preoptimized molecules to build the full unit cell consisting of multiple molecules.

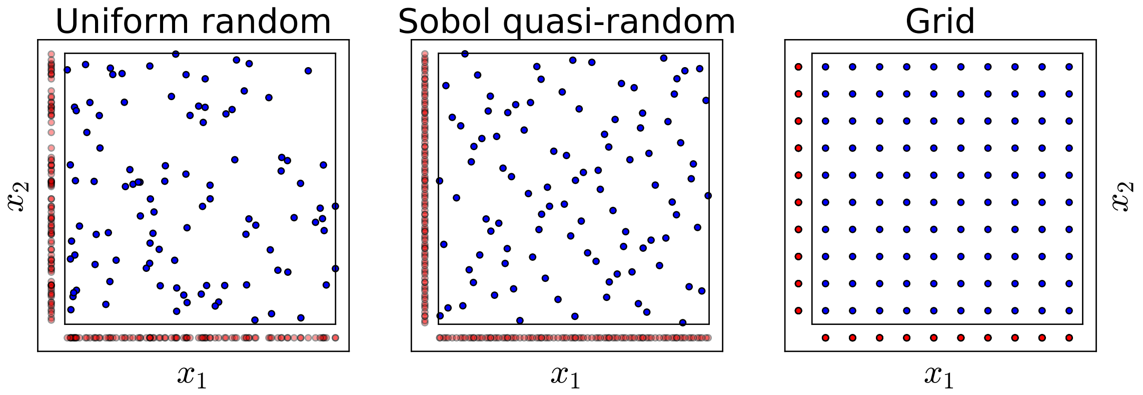


Figure 1: *100 points (blue) generated by a uniform random distribution, a quasi-random Sobol sequence and a regular grid. The Sobol sequence leaves fewer "unexplored holes" in the configurational space compared to the uniform random distribution, just like the regular grid. When projected onto one of the coordinates (red) the Sobol sequence and the uniform distribution still give a uniform distribution in 1D while the grid gives a sharply peaked distribution.*

1.4.3 Quasi-Random Sobol Sequence

When independently drawing random points, in many cases some areas of the configurational space will be covered more densely than others, leading to an uneven exploration and risking oversight of energetically low minima. Quasi-random Sobol sequences generate points that are correlated and more evenly distributed throughout space. Furthermore the points are also still evenly distributed when projected onto a lower-dimensional subspace, as can be seen in Fig. 1 where 100 uniformly distributed random points in the 2D plane are compared to 100 quasi random points from a Sobol sequence and 100 points on a regular grid. Sobol sequences try to combine the best features of grid searches - even coverage of parameter space - with the best features of random searches - unbiased search and good distribution of projections.

1.4.4 Markov-Chain-Monte-Carlo Simulated Annealing

Grid-searches or random searches sample the parameter space independently of the value of the objective function. If the parameter space is high dimensional, an increasingly large fraction of function evaluations are spent on regions that are of little interest. To improve convergence of the optimization it is desirable to spend more time in regions of the parameter space that are targeted by the optimization, e.g. low energy polymorphs. A possible way to do so is given by the family of Markov-Chain Monte-Carlo algorithms. The basic idea is to have a "walker" that moves through the parameter space and preferentially walks towards regions of low energy. At every iteration a random movement of the walker is proposed and its resulting change in energy ΔE is evaluated. If the change in energy was favorable, $\Delta E < 0$, the walker moves to this new position. If $\Delta E > 0$, the walker moves to this new position with a probability that decreases with larger energy changes. A common choice of probability function is given by the Boltzmann weight:

$$p_{accept} = e^{-\frac{\Delta E}{T}} \tag{1}$$

The "temperature" parameter T defines the energy scale at which changes in energy are considered to be too large to accept. Choosing large values of T will allow large movements of the walker in parameter space, even if these movements lead to intermittent energetically unfavorable points. Low values of T prevent large increases in energy and lead to convergence to the closest local minimum.

Simulated annealing exploits this behavior by changing this "temperature" during the optimization. Starting at high values of T fast exploration of the parameter space is possible. The temperature is then gradually decreased to focus more and more on local optimization of the potential energy surface.

1.4.5 Basin Hopping

One significant problem for Monte Carlo methods, described in Sec. 1.4.4, are energetic barriers between local minima. If the walker is trapped in a local minimum surrounded by high energetic barriers it is unlikely that this minimum can ever be left. Any proposed step out of this minimum will result in a large increase in energy E and therefore a small acceptance probability p. To address this problem of getting stuck in local minima, Basin Hopping combines a Monte-Carlo approach with a local optimization. For each proposed point in the parameter space a local optimization (e.g. gradient descent) is conducted and the energy assigned to the point is given by the local minimum [6]. This effectively transforms the potential energy surface into a staircase of energies that correspond to the energies of the local minima, as depicted in Fig. 2. Alternatively one can also move the position of the walker to the position of the local minimum that was found by the optimization, so that subsequent steps start from a low energy configuration.

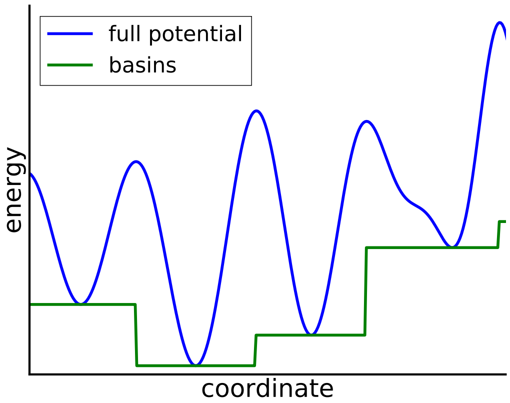


Figure 2: *Schematic of Basin-Hopping transformation. The original, full potential is transformed into a staircase of minima energies, allowing faster traversal of the potential energy landscape.*

Basin Hopping can in some cases vastly improve the speed at which the potential energy surface can be sampled, compared to traditional Monte-Carlo methods, due to faster traversal between local minima. Basin Hopping has been applied to the problem of structure search for Lennard-Jones Clusters [6] and the structures of Si and Cu clusters [7]. While Basin Hopping aims to facilitate faster transitions between local minima, it can still suffer from often revisiting the same energetic basin many times. This will happen in particular if a basin is very large in the parameter space, so that many random moves will land in it. In many cases wide basins are also particularly "deep" and thus of interest, but there are also cases where a wide, and thus often visited, basin does not lead to the global minimum. Several techniques have been tried to minimize revisits of already known basins, for example by energetically penalizing regions of the parameter space that have already been visited.

1.4.6 Genetic Algorithm

Genetic Algorithms (GA) are inspired by nature and its evolutionary improvement of species by mutations and natural selection. In a Genetic Algorithm a set of possible configurations in the parameter space is seen as a “population”. In each iteration the following steps are performed:

1. **Evaluation of fitness**: The quality of each member of the population is evaluated. For the case of structure search a simple, exemplary fitness function is given by $fitness = -E$.
2. **Selection**: Members with low fitness are removed from the population. The selection can be a sharp thresholding or a probabilistic choice that gives a higher chance to be eliminated to low-fitness-configurations.
3. **Crossover**: High-fitness members of the population are now paired to create “offspring”. The exact way of combining two members to create a new one depends on the application. Since members of the next generation consist of combinations from high-fitness "parents", their average fitness should be higher than the average fitness of the previous generation.
4. **Mutation**: To bring new features into the population, some members of the new generation are randomly altered by performing variations on part of their descriptors.

These steps are iterated until the population converges to a high-fitness solution of the optimization problem. Compared to traditional Markov-Chain-Monte-Carlo methods, genetic algorithms benefit from two important differences: Crossovers and mutations can lead to large movements in the parameter space, allowing faster escape from local minima. Secondly, crossovers can combine the best traits of two configurations, leading to faster convergence towards the global optimum. Mutations ensures that the population stays diverse during the optimization and should prevent getting stuck in local minima. Genetic algorithms alone are not well suited for fine tuning of parameters and can therefore be coupled with local optimization steps, referred to as hybrid-GA.

Like any stochastic optimization algorithm, also Genetic Algorithms do not guarantee to find the global optimum but they have been successfully employed to find sufficiently good solutions to difficult optimization problems. Genetic Algorithms have been applied to a wide range of problems, often involving large parameter spaces: Folding of proteins [8], tuning of parameters to make simulations fit experimental results [9] and automated design of a special antenna for the NASA space mission ST5 [10].

1.5 Calculating Energies

Each of the optimization methods outlined in Sec. 1.4 relies on being able to evaluate the function value at any point in the parameter space. A central part in structure search - and in many cases the computationally most demanding one - is therefore a routine that yields the total energy of a given geometry. This section will briefly list a few common computational methods for obtaining energies and follows closely the book *Introduction to Computational Chemistry: Second Edition* by Frank Jensen [11].

1.5.1 Force-Fields

Force-field methods solve the problem classically (i.e. not quantum mechanically) by treating a molecule as a set of atoms that are connected by bonds. Deforming a molecule causes the bonds to stretch, twist and deform, all contributing towards the energy of a system. The parameters to describe this energy model are usually obtained by experiments or ab-initio quantum chemistry methods. A typical force-field, like the AMBER force-field [12] calculates the energy as:

$$E = E_{stretch} + E_{bend} + E_{torsion} + E_{coloumb} + E_{vdW} \tag{2}$$

The first term $E_{stretch}$ models the energy required to change a bond length l. A common choice is

$$E_{stretch} = E_0 + \frac{k}{2}(l - l_0)^2 \tag{3}$$

which describes a harmonic potential with "spring constant" k, equilibrium bond length l_0 and equilibrium bond energy E_0. This can be viewed as the lowest contributing terms of a Taylor expansion of the true bond potential. This model can be refined further by introducing higher order terms like $k_n(l - l_0)^n$ at the expense of introducing additional parameters. As opposed to the exact bond potential, any polynomial expansion will not converge towards zero for $l \to \infty$ and therefore will only be valid for small perturbations of the bond length. A potential that does obey the correct boundary conditions is for example the Morse potential

$$E = E_0 \left(1 - e^{-a(l-l_0)}\right)^2 - E_0 \tag{4}$$

The second term E_{bend} describes distortions from equilibrium bond angles (e.g. the $\theta = 104.45°$ for the water molecule). As for the stretching energy, a common choice is a quadratic expansion around the minimum:

$$E_{bend} = E_0^{bend} + \frac{k^{bend}}{2}(\theta - \theta_0)^2 \tag{5}$$

The torsion energy arises in a system of four atoms A-B-C-D when there is a rotation around the B-C axis. It is characterized by an angle ω which, as opposed to bond angles, often has several minima and in some cases deviates strongly from the equilibrium value. Therefore, the energy associated with the angle must correctly obey the rotational symmetry that adding a rotation of 360° to ω should not alter the energy. This property can automatically be enforced by expanding the energy in a periodic Fourier series:

$$E_{torsion} = \sum_n E_n \cos\left(n(\omega - \omega_n^0)\right) \tag{6}$$

While the energies $E_{stretch}$, E_{bend} and $E_{torsion}$ only appear for atoms that are connected by bonds, there are also non-bonded energy contributions between all pairs of atoms. The Coulomb energy describes the electrostatic attraction or repulsion between fully or partially charged atoms. The monopole contribution between atoms i and j is given by

$$E_{coloumb} = \frac{q_i q_j}{r_{ij}} \tag{7}$$

where q_i and q_j denote their electric charges in atomic units. The charges can be obtained by fitting the electrostatic potential generated by these point charges to the potential calculated by an ab-initio method. The electrostatic potential can be further refined by including higher order contributions beyond the monopole, in particular the dipole and quadrupole moments of the electrostatic potential.

The last energy term is the van der Waals energy E_{vdW} which is a quantum mechanical interaction arising from charge density fluctuations. For large distances the interaction between atoms is attractive and decays as r^{-6}. At very small distances the wavefunction of atoms start to overlap, leading to Pauli repulsion. Since this repulsion is not yet included in any of the other energy terms, it is often included as a repulsive part of the van der Waals energy. A popular energy function describing these behaviors is given by the Lennard-Jones potential:

$$E_{vdW} = 4E_0 \left[\left(\frac{r}{r_0} \right)^{-12} - \left(\frac{r}{r_0} \right)^{-6} \right] \tag{8}$$

This function has a minimum at $r = r_0$ of depth $E = E_0$ and decays to zero for large distances.

Force-field methods are computationally cheap and therefore widely used when more accurate methods are computationally unfeasible. They have for example been applied to large organic molecules and in the field of protein folding. Because force-fields and other empirical methods rely on parametrization on known datasets, it is fundamentally difficult or even impossible to properly address new or rare interactions.

1.5.2 Wave Function Based Methods

Wave function based methods use the time-independent Schrödinger equation as their starting point:

$$\hat{H}\psi(\boldsymbol{R}, \boldsymbol{r}) = E\psi(\boldsymbol{R}, \boldsymbol{r}) \tag{9}$$

Here $\hat{H}$ denotes the Hamiltonian of the system and ψ and E are its eigenfunctions and eigenvalues respectively. The wavefunction ψ depends on the position of all nuclei $\boldsymbol{R}$ and all electrons $\boldsymbol{r}$. The square of the wavefunction $|\psi|^2$ gives the probability density of finding the nuclei at $\boldsymbol{R}$ and the electrons on $\boldsymbol{r}$. For a non-relativistic treatment, neglecting spin, the Hamiltonian of a molecule in atomic units is given by:

$$\hat{H} = -\sum_i \frac{1}{2}\nabla_i^2 - \sum_a \frac{1}{2M_a}\nabla_a^2 + \frac{1}{2}\sum_{i \neq j} \frac{1}{r_{ij}} + \frac{1}{2}\sum_{a \neq b} \frac{Z_a Z_b}{r_{ab}} - \sum_a \sum_i \frac{Z_a}{r_{ia}} \tag{10}$$

Here indices i, j denote electrons and indices a, b correspond to nuclei. The terms in the Schrödinger Equation describe the kinetic energy of the electrons and the kinetic energy of the nuclei, the Coulomb repulsion between nuclei-nuclei, electrons-electrons and the attraction of electrons to the nuclei. Solving this equation does yield all possible eigenstates ψ - the groundstate as well as all excited states - and their corresponding energies. However for a system consisting of N particles (nuclei and electrons) the wavefunction is a complex $3N$ dimensional function, the determination of which is unfeasible even for systems of moderate size. To simplify the problem, several approximations are being made, with the first one usually being the Born-Oppenheimer approximation: In the Born-Oppenheimer approximation the nuclei are not treated quantum-mechanically but as classical particles having a well defined position. Only the electrons that move in the electrostatic potential of these fixed nuclei are then treated quantum-mechanically. The approximation is in many cases well justified, because the nuclei are heavier then the electrons by about four orders of magnitude and appear quasi-stationary to the fast electrons. Treating the nuclei classically turns the kinetic energy of the nuclei and their electrostatic repulsion into constants that can be removed from the Hamiltonian and added to the eigen-energies after solving the Schrödinger equation.

Even in the Born-Oppenheimer approximation the wavefunction does still depend on $3N_{electrons}$ coordinates and is difficult to calculate. The main difficulty arises from the electron-electron repulsion that correlates the movement of electrons. If the electron-electron interaction is neglected, the total Hamiltonian becomes a sum of Hamiltonians that only act on one of the electrons.

$$\hat{H} = -\sum_i \left(\frac{1}{2} \nabla_i^2 - \sum_a \frac{Z_a}{r_{ia}} \right) = \sum_i \hat{h}_i \tag{11}$$

This Hamiltonian can immediately be solved by a product ansatz:

$$\psi(\boldsymbol{r}) = \phi_1(\boldsymbol{r}_1)\phi(\boldsymbol{r}_2)...\phi(\boldsymbol{r}_N) \tag{12}$$

where the orbitals ϕ_i are solutions to the noninteracting Hamiltonian $\hat{h}$:

$$\hat{h}\phi_i = E_i \phi_i \tag{13}$$

Since electrons are fermions, the electronic wavefunction must additionally be antisymmetric with respect to the permutation of electrons. This can be achieved by using a so called Slater determinant as ansatz instead of the simple product state:

$$\psi = \begin{vmatrix} \phi_1(\boldsymbol{r_1}) & \phi_2(\boldsymbol{r_1}) & ... & \phi_N(\boldsymbol{r_1}) \\ \phi_1(\boldsymbol{r_2}) & \phi_2(\boldsymbol{r_2}) & ... & \phi_N(\boldsymbol{r_2}) \\ \vdots & \ddots & & \vdots \\ \phi_1(\boldsymbol{r_N}) & \phi_2(\boldsymbol{r_N}) & ... & \phi_N(\boldsymbol{r_N}) \end{vmatrix} \tag{14}$$

Expanding this determinant as a sum of permutations does include the simple product ansatz from Eq. 12 but also adds all possible permutations with the corresponding sign of the permutation. To find the full wavefunction in this non-interacting approximation, it is now possible to just solve the Hamiltonian $\hat{h}$ for a single electron, which is identical for all electrons, determine its eigenfunctions ϕ_i and construct the full wavefunction as a Slater determinant of the orbitals that have the lowest energy.

Fully neglecting the electron-electron interaction ignores a large part of the Hamiltonian and therefore yields very poor results for most materials, it is therefore necessary to include the Coulomb repulsion:

$$\hat{g}_{ij} = \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} \tag{15}$$

When including the Coulomb repulsion in the total Hamiltonian, the Slater determinant (Eq. 14) is no longer an eigenstate of the total Hamiltonian, however it can be used as a variational ansatz. An approximate solution can be found by minimizing the energy $E = \left\langle \psi \middle| \hat{H} \middle| \psi \right\rangle$ as a function of the orbitals ϕ_i. When evaluating the energy of a given Slater determinant, constructed out of orthonormal orbitals, only terms that have different orbitals in at most two coordinates contribute, while all other terms are zero due to the orthonormality of the orbitals. The energy of a Slater determinant is given by:

$$
\begin{aligned}
E &= \left\langle \psi \middle| \hat{H} \middle| \psi \right\rangle = && (16)\\
&= \sum_i \left\langle \phi_i \middle| \hat{h}_1 \middle| \phi_i \right\rangle + \sum_{i<j} \langle \psi \,|\, \hat{g}_{ij} \,|\, \psi \rangle && (17)\\
&= \sum_i \left\langle \phi_i \middle| \hat{h}_1 \middle| \phi_i \right\rangle + \sum_{i<j} \langle \phi_i(1)\phi_j(2) \,|\, \hat{g}_{12} \,|\, \phi_i(1)\phi_j(2) \rangle - \langle \phi_i(1)\phi_j(2) \,|\, \hat{g}_{12} \,|\, \phi_j(1)\phi_i(2) \rangle && (18)\\
&= \sum_i \left\langle \phi_i \middle| \hat{h}_1 \middle| \phi_i \right\rangle + \frac{1}{2}\sum_{ij} \langle \phi_i(1)\phi_j(2) \,|\, \hat{g}_{12} \,|\, \phi_i(1)\phi_j(2) \rangle - \langle \phi_i(1)\phi_j(2) \,|\, \hat{g}_{12} \,|\, \phi_j(1)\phi_i(2) \rangle && (19)\\
&= \sum_i \left\langle \phi_i \middle| \hat{h}_1 + \frac{1}{2}\sum_j \hat{J}_j - \hat{K}_j \middle| \phi_i \right\rangle && (20)\\
& && (21)
\end{aligned}
$$

In the last two steps the Coulomb operator $\hat{J}$ and the exchange operator $\hat{K}$ have been introduced:

$$\hat{J}_j \,|\phi_i(1)\rangle = \langle \phi_j(2) \,|\, \hat{g}_{12} \,|\, \phi_j(2) \rangle \;\; |\phi_i(1)\rangle \tag{22}$$

$$\hat{K}_j \,|\phi_i(1)\rangle = \langle \phi_j(2) \,|\, \hat{g}_{12} \,|\, \phi_i(2) \rangle \;\; |\phi_j(1)\rangle \tag{23}$$

Defining

$$\hat{F}_i = \hat{h}_1 + \frac{1}{2}\sum_j \hat{J}_j - \hat{K}_j \tag{24}$$

does formally lead to a problem of independent particles

$$E = \sum_i \left\langle \phi_i \middle| \hat{F}_i \middle| \phi_i \right\rangle \tag{25}$$

The orbitals that minimize Eq. 25 are the eigenstates of $\hat{F}$, however the operator $\hat{F}$ itself does depend on the orbitals ϕ_i via the Coulomb and exchange operators $\hat{J}$ and $\hat{K}$. Therefore Eq. 25 is usually solved self-consistently: Starting with an initial guess for the orbitals ϕ (e.g. starting with the non-interacting orbitals) the operator $\hat{F}$ is constructed. By diagonalizing $\hat{F}$, a new set of orbitals ϕ_i is obtained. This scheme is iterated until a self-consistent solution for the orbitals ϕ_i and $\hat{F}$ is found. This approach is called the Hartree-Fock method and yields the best possible (i.e. lowest in energy) wavefunction that can be obtained by a single Slater determinant.

To obtain better wavefunctions that lower the total energy further it is necessary to generalize the ansatz of Eq. 14 by writing the wavefunction as a sum of multiple determinants. Different determinants can be constructed from the orbitals ϕ by using orbitals with higher energy than the lowest N orbitals to construct the determinant. If all possible orbitals - which there are infinitely many of - and all possible combinations of these were used to construct all possible Slater determinants, this ansatz would be fully general and yield the exact wavefunction. In practice the number of orbitals is limited by the number of basis functions that are used to construct the orbitals. Furthermore in most cases only a subset of all possible Slater determinants is included, for example only the Hartree-Fock determinant and the doubly excited determinants. This ansatz of a truncated sum of excited determinants of the Hartree-Fock orbitals is known as Configurational Interaction (CI).

1.5.3 Semi-Empirical Methods

To evaluate the electron-electron interaction energy of a slater determinant in Hartree-Fock the following integral, besides others, must be evaluated:

$$E = \frac{1}{2}\sum_{i\neq j}\left\langle \psi \middle| \frac{1}{r_{ij}} \middle| \psi \right\rangle \tag{26}$$

Since the operator only acts on two (r_i and r_j) out of all the electron coordinates all other coordinates can immediately be integrated, thus leaving only the orbitals dependent on the coordinates r_i and r_j. The general form of these remaining two-electron integrals is therefore:

$$V^{ij}_{abcd} = \int\int \phi^*_a(\boldsymbol{r_i})\phi^*_b(\boldsymbol{r_j})\frac{1}{r_{ij}}\phi_c(\boldsymbol{r_i})\phi_d(\boldsymbol{r_j}) \tag{27}$$

If the orbitals are expanded in M basis functions than there are $\mathcal{O}(M^4)$ of these two-electron integrals that need to be evaluated, leading to a large computational cost at increasing system size.

Semi-empirical methods provide a computationally cheaper approach by neglecting all these two-electron integrals. This large simplification introduces errors than can partly be compensated by turning the remaining integrals into parameters that are fitted to experiments or more sophisticated computations. Further computational time can be saved by only treating the (chemically most interesting) valence electrons of a system by combining the core electrons together with the nucleus to a new nucleus that has a reduced effective electric charge. As with Force-Field methods also semi-empirical methods can only give reliable results for classes of materials that they have been parametrized for. An example of such a semi-empirical method is the AM1 method, which has been parametrized for small organic molecules. [13].

1.5.4 Density Functional Theory

In wavefunction based methods, as described above, the central object is the all-electron wavefunction which is complex and depends on $3N$ electron coordinates. Once the groundstate wavefunction is known, all groundstate properties can be calculated as expectation values of this wavefunction. Hohenberg and Kohn have shown that also the electron density is sufficient to calculate all groundstate properties of any system. This is in principle advantageous for calculation of properties, because the electron density is a much simpler object, being real valued and, more importantly, only dependent on three coordinates, independently of the number of electrons in the system.

If the external potential is fully known, for example the electrostatic potential generated by the nuclei, the Hamiltonian of the system is known, which - for a non-degenerate groundstate - uniquely defines the groundstate wavefunction, as depicted in Eq. 29. Integrating the square of the wavefunction over all but one coordinate yields the electron density:

$$n(\boldsymbol{r}) = \int \psi^*\psi \, d^{3N-3} \tag{28}$$

$$V(\boldsymbol{r}) \rightarrow \hat{H} \rightarrow \psi \rightarrow n(\boldsymbol{r}) \tag{29}$$

If it was possible to reverse Eq. 29, obtaining the potential from the electron density, then the wavefunction - and thus all groundstate properties - would be determined by the electron density.

The Hohenberg-Kohn theorem states that the potential can be inferred from the density and can be proven by *reduction ad absurdum*: Assume there are two different external potentials $V_1(\boldsymbol{r}), V_2(\boldsymbol{r})$ with different, non-degenerate groundstate wavefunctions ψ_1, ψ_2 that give rise to the same density n.

$$\begin{aligned} V_1(\boldsymbol{r}) \rightarrow \psi_1 \rightarrow n(\boldsymbol{r}) \\ V_2(\boldsymbol{r}) \rightarrow \psi_2 \rightarrow n(\boldsymbol{r}) \end{aligned} \tag{30}$$

The expectation value of the energy of a wavefunction ψ with respect to the Hamiltonian can be written as a sum of kinetic energy T, external potential V and electron-electron interaction U. Since ψ_1 is the groundstate, and thus the state of lowest energy, of the Hamiltonian corresponding to V_1 it holds that:

$$\begin{gathered} \langle \psi_1 \,|\, H_1 \,|\, \psi_1 \rangle < \langle \psi_2 \,|\, H_1 \,|\, \psi_2 \rangle \\ \langle \psi_1 \,|\, T + U \,|\, \psi_1 \rangle + \int V_1(\boldsymbol{r}) n(\boldsymbol{r}) \, d^3 r < \langle \psi_2 \,|\, T + U \,|\, \psi_2 \rangle + \int V_1(\boldsymbol{r}) n(\boldsymbol{r}) \, d^3 r \\ \langle \psi_1 \,|\, T + U \,|\, \psi_1 \rangle < \langle \psi_2 \,|\, T + U \,|\, \psi_2 \rangle \end{gathered} \tag{31}$$

However, the same argument can be made when calculating the expectation value of $\hat{H}_2$ leading to the contradiction:

$$\langle \psi_1 \,|\, T + U \,|\, \psi_1 \rangle < \langle \psi_2 \,|\, T + U \,|\, \psi_2 \rangle < \langle \psi_1 \,|\, T + U \,|\, \psi_1 \rangle \tag{32}$$

This proves that the assumption in Eq. 30 must have been wrong: It is not possible that two different potentials lead to the same ground state density. Therefore the groundstate density n uniquely determines the external potential V.

While the Hohenberg-Kohn-Theorem is a remarkable result, it is by itself of very limited use for practical applications: It proves that all groundstate properties, for example the total energy, of the system are a functional of the electron density, but unfortunately this functional is not known. Because the exact functional is not known, approximate functionals are used. In the development of orbital-free Density Functional Theory (DFT) the aim is to explicitly find the full functional that yields the energy for a given groundstate electron density. Some parts of the functional are known exactly, for example the potential energy of the electrons in the external potential V:

$$E[n] = \int V(\boldsymbol{r}) n(\boldsymbol{r}) d^3 r \tag{33}$$

However another large contribution to the total energy is given by the kinetic energy of the electrons, for which only approximate functionals such as the Thomas-Fermi functional exist:

$$E[n] \propto \int n(\boldsymbol{r})^{\frac{5}{3}} d^3 r \tag{34}$$

Because the kinetic energy is a large part of the total energy of a system, making rough approximations for it introduces large errors in the results. For this reason most DFT codes use the Kohn-Sham equations. In Kohn-Sham-DFT auxiliary orbitals ϕ are introduced that form a system of noninteracting electrons giving rise to the same electron density as the original, interacting system:

$$n(\boldsymbol{r}) = \sum_i \phi_i^* \phi_i \tag{35}$$

The kinetic energy is then given by the kinetic energy of these non-interacting auxiliary particles

$$T = -\frac{1}{2} \sum_i \int \phi_i \nabla^2 \phi_i \, d^3r \tag{36}$$

The energy of the electrons in the external potential and the Coulomb repulsion is given by

$$\begin{aligned} E_{potential} &= \int \left[V_{ext}(\boldsymbol{r}) + V_{hartree}(\boldsymbol{r}) \right] n(\boldsymbol{r}) \, d^3r \\ V_{hartree}(\boldsymbol{r}) &= \int \frac{n(\boldsymbol{r'})}{|\boldsymbol{r} - \boldsymbol{r'}|} \, d^3r' \end{aligned} \tag{37}$$

Using the Kohn-Sham orbitals, the only remaining unknown part of the energy functional is the exchange- and correlation-energy E_{xc}. The exact form of this functional is not known and a large variety of different approximate functionals exist. The advantage of the Kohn-Sham approach over orbital-free DFT is that the exchange and correlation energy typically only accounts for a small fraction of the total energy and even crude approximations can give acceptable accuracies for the total energy.

1.5.5 DFT Exchange-Correlation Functionals

Different exchange-correlation-functionals can be classified according to Perdew's Ladder [14] which ranks functionals from basic and computationally cheap versions, to more sophisticated and computationally expensive functionals.

1. **LDA**: In the Local Density Approximation E_{xc} is given by an integral over d^3r of a function that only depends on the electron density $n(\boldsymbol{r})$. This exchange functional is exact for a uniform electron gas and the approximation is therefore good for systems that have a slowly varying electron density, such as the valence electrons of alkali metals. LDA does perform poorly for strongly varying electron densities, as they occur in molecules. Common parametrizations are for example given by Vosko-Wilk-Nusair (VWN-LDA) [15] or Perdew-Wang (PW-LDA) [16].

2. **GGA**: In the Generalized Gradient Approximation (GGA) the exchange-correlation potential depends also on $\frac{|\nabla n(\boldsymbol{r})|}{n(\boldsymbol{r})}$ in addition to n, improving the accuracy for electron densities that vary in space. A common functional of this kind was developed by Perdew, Burke and Ernzerhof (PBE) [17].

3. **Meta-GGA**: Meta-GGA functionals take the logical next step after GGA by including the second derivative of the electron density in the exchange-correlation potential, or equivalently by including the kinetic energy density $\sum_i |\nabla \phi_i|^2$. A recent example of this class is given by the SCAN functional [18].

4. **Hybrid-Functionals**: Hybrid functionals mix a certain fraction of Hartree-Fock like exchange to the exchange-correlation functional. This comes at the cost of having a non-local exchange potential that depends not only on the density and its derivatives at

position $\boldsymbol{r}$, but on the density in the entire space, requiring an additional integration. While LDA and GGA typically underestimate band-gaps in semiconductors they are typically overestimated in Hartree-Fock calculations. Including of Hartree-Fock exchange allows tuning of the functional to fit experimental values. Such an empirical Hybrid-Functional that has shown to give good results for molecules is the B3LYP functional, which mixes LDA, GGA and Hartree-Fock exchange [19]. Another common functional is PBE0 which mixes 25% of Hartree-Fock exchange to the PBE functional, justified by perturbation theory results [20].

5. **RPA**: The random phase approximation (RPA)is the highest tier of Perdew's Ladder and additionally also includes information from the unoccupied Kohn-Sham Orbitals. While it has the advantage of being able to directly include van der Waals interactions, its computational cost is currently prohibitive for all but the smallest systems.

Density Functional Theory at or above the level of GGA functionals has been shown to yield good accuracies for a wide class of materials ranging from molecules to solids. A drawback of DFT, compared to wavefunction based methods, is that there is no clear and systematic way to converge towards an exact solution. While adding additional Slater determinants in a CI calculation is guaranteed to monotonically converge towards the exact total energy, moving through the rungs of Perdew's Ladder does not necessarily do so.

1.5.6 Van der Waals Corrections

Most DFT exchange-correlation functionals below the RPA level do not properly describe van der Waals (vdW) interactions. To get reliable interaction energies for systems where van der Waals interactions are dominant, such as physisorbed molecules on surfaces, ad-hoc van der Waals corrections are used. Since it is known that vdW interactions between atoms are proportional to r^{-6} for long distances r, many vdW schemes add a pairwise interaction energy between all atoms to the total energy:

$$E_{vdW} = \sum_{i \neq j} \frac{C_6^{ij}}{r_{ij}^6} \tag{38}$$

Grimme proposed multiplying an additional damping function to the r^{-6} distance dependency [21]. The C_6 coefficients are either fitted empirically [22] or can be determined from the electron density as proposed by Tkatchenko and Scheffler (TS) [23]. This TS-vdW correction has also been adapted to give good results for hybrid organic-inorganic interfaces [24]. All these van der Waals corrections are applied after the DFT self consistency cycle has converged and are computationally cheap since they only involve summation over pairs of atoms.

Another approach to van der Waals effects is given by the Many Body Dispersion (MBD) vdW correction which calculates the dynamic atomic polarizabilities by mapping the problem to coupled quantum harmonic oscillators sitting at each atomic site [25]. This more fundamental type of van der Waals correction comes at an increased computational cost, but has been shown to closely match experimental results when paired with the hybrid functional HSE [26].

1.6 Machine Learning

Ab-initio methods are computationally demanding, which can make it difficult to sample a sufficiently large part of the configurational space with any of the methods outlined in Sec.

1.4. However it is observed that the results of quantum mechanical calculations can often be explained in much simpler terms than the full electron density or the all-electron wavefunction: Many concepts of chemistry, for example chemical bonds or partial atomic charges, are not strict observables of the Hamiltonian but are rather descriptors that efficiently explain results for large classes of materials. The aim of machine learning methods is to find such descriptors from a given training set in an automated fashion. Once these descriptors and their correlation to material properties are known, this simplified model can be used to predict properties for samples that were not in the training set.

The following section will give a brief outline of a few commonly used machine learning methods and their application to the materials sciences.

1.6.1 Sparse Subset Selection

One straightforward way to find good descriptors is to simply build a large list of possible descriptors and then select a subset of these that give the best property prediction for a given training set. This has been done by Ghiringhelli et al [27] for the prediction of crystal structures of semiconductors. To build the list of possible descriptors they start with a few basic descriptors of each training sample (e.g. atomic radii, electronegativities, etc.) and expand this list of descriptors by combining theses simple descriptors by a variety of functions (adding descriptors, multiplying them, etc.). Once all N descriptors have been calculated for all training samples and stored in a matrix D, they select the subset of n descriptors that gives the best linear fit for the target properties $\boldsymbol{p}$ of the training set, in their case the energy difference between crystal structures. They show that even small descriptor sets with $n = 5$ can yield high prediction accuracies.

Because there are $\binom{N}{n}$ possible ways to select n descriptors from a list of N possibilities it is unfeasible to try all possible descriptor sets until the best one has obtained. This problem is circumvented by using an L1-regularized linear regression: Instead of performing linear fits on subsets, searching for the best subset, a linear fit is performed using all descriptors. To enforce sparsity of the model - only n out of N properties should have a nonzero fit coefficient - the L1 norm of the fit coefficients is added as regularization:

$$\arg\min_{\boldsymbol{c}} ||\boldsymbol{p} - D\boldsymbol{c}||_2^2 + \lambda ||\boldsymbol{c}||_1 \tag{39}$$

Here $||\boldsymbol{c}||_1$ denotes the L1 norm of the vector $\boldsymbol{c}$:

$$||\boldsymbol{c}||_1 = \sum_i |c_i| \tag{40}$$

This procedure is known as LASSO [28]: Least Absolute Shrinkage and Selection Operator. Because the fit model is sparse, it is often particularly well interpretable since the fit function is only given by a sum over few nonlinear functions of basic properties.

1.6.2 Artificial Neural Networks

Building the functions that map basic descriptors of training samples to their properties is difficult because the properties can depend on the input in a highly nonlinear fashion. Instead of finding a one-step solution to this regression problem, Artificial Neural Networks attempt to model a system layer by layer. The basic structure of a Feed-Forward-Neural-Network is

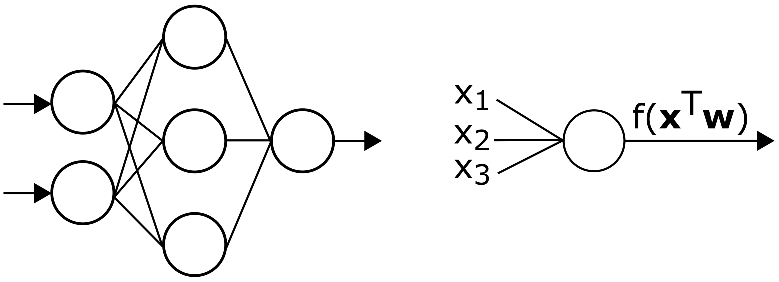


Figure 3: *Example of a feed-forward artificial neural network consisting of three layers: And input layer with two nodes, a hidden layer with 3 nodes and the output layer with one node. Each node receives input from previous layers and outputs the value of a function of the weighted sum of inputs.*

depicted in Fig. 3. Each neuron (represented by a circle in Fig. 3) receives inputs x_i from all nodes of the previous layer. The output of each node is then calculated as:

$$y = f\left(\sum_i x_i w_i\right) \tag{41}$$

where w_i are weights that can be different for each node and input and f is some nonlinear, usually saturating function. Common choices for $f(x)$ include $f(x) = tanh(x)$, the sigmoid function $f(x) = \frac{e^x}{e^x+1}$ or the rectifying linear unit $f(x) = max(0, x)$.

Supervised training of neural networks is typically done using the backpropagation algorithm: A batch of training samples is fed trough the neural network and its output is computed for each training sample. Then the mean residual R between the prediction and the exact result is calculated. Using the chain rule of differentiation, the partial derivatives of the residual with respect to all parameters $\boldsymbol{w}$ of the model can be calculated. The weights are then updated to minimize the residual, for example by a steepest descent procedure:

$$\boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \eta \boldsymbol{\nabla}_w R \tag{42}$$

Due to the large number of parameters and their nonlinear structure neural networks are very flexible and can yield very good prediction accuracies given enough training data. Deep neural networks applied to large training sets have obtained unprecedented success in a variety of applications, such as recognition of handwritten digits [29], image classification [30] and even playing the board game Go.

For these classification tasks Deep Neural Networks, consisting of many hidden layers, have performed extremely well. One of the reason for the success of neural networks is their layered structure that extracts specific features at each layer of the network, forming an increasingly abstract representation of the input data. For the example of image classification the input data might simply be the intensity at each pixel of an image. The first layers of a multilayer network could then for example perform edge detection, the next layers might recognize basic shapes (circles, rectangles), the consequent layers more abstract concepts such as cars or faces.

Training these deep networks with the traditional backpropagation algorithm is hard because of the vanishing gradient problem: Small changes at the first layers of the network only weakly influence the final output of the network, in particular if some nodes of the network are in the saturated regime of the nonlinear function. This has partly been addressed by efficient GPU implementations of the backpropagation algorithm that are orders of magnitude faster than traditional CPU implementations and allow far more training iterations. This can eventually lead to convergence of the network despite small and noisy gradients.

A second problem of deep neural networks is that they contain a very large number of parameters: the input weights for all nodes. To avoid over-fitting onto the training dataset, usually large datasets are used to train neural networks. While large datasets are readily available for many classification challenges, for the problem of structure search generating large training sets might be computationally prohibitive.

1.6.3 Gaussian Process Regression

Gaussian Process Regression uses Bayes' theorem to derive a probability distribution for predictions, given some known training samples. For the prior distribution a multivariate Gaussian distribution is assumed, which is characterized by its mean value μ and its covariance C. Prior knowledge about the system can be encoded into in these quantities to achieve accurate predictions using few training samples. Gaussian Process Regression (GPR) and the closely related Kernel Ridge Regression (KRR) both employ regularization to avoid overfitting of the model. As opposed to neural networks that are extremely flexible but need large training sets to avoid overfitting, GPR can yield good results at smaller training sets at the expense of limited flexibility. While training on very large training sets ($>$ 100.000 samples) is straightforward and routinely done for neural networks, training a Gaussian Process can become computationally challenging. GPR and KRR have been successfully applied to the problem of predicting the atomization energies of small organic molecules [31] and been shown to achieve better prediction accuracies than neural networks [32], in particular for small training set sizes.

Gaussian Process Regression to predict interaction energies lies at the heart of this work and is described in detail in Sec. 2.6.

1.6.4 Cluster Expansion

Cluster Expansion is a method that has originally been developed to model the energy of binary alloys. It maps a crystal onto an effective, generalized Ising-like model where the type of atom present on lattice site i is described by a "spin" variable σ_i. If the site is occupied by atom type A, $\sigma = +1$, if the site is occupied by B, $\sigma = -1$. The energy of any configuration of atom types is fully determined by specifying all "spin" variables and is given by a sum of all possible correlation functions [33]:

$$E(\boldsymbol{\sigma}) = J_0 + \sum_{\text{sites}} J_i \sigma_i \sum_{\text{pairs}} J_{ij} \sigma_i \sigma_j + \sum_{\text{triplets}} J_{ijk} \sigma_i \sigma_j \sigma_k + \cdots \tag{43}$$

A classical Ising model is obtained if triplets and higher contributions are omitted, and the pair-sum only runs over nearest neighbor atoms. Including all terms of the sum does in principle yield an exact model but becomes quickly unfeasible as the number of possible correlations rises quickly with the number of atoms considered within it. Therefore two approximations are typically made:

- Only correlations up to a maximum number of atoms within it are considered (e.g. stopping at quadruplets)
- Only correlations within a certain distance cutoff are considered.

This truncates this energy expansion to a finite number of fit coefficients J that must be determined by fitting to training data.

In many cases even this truncation leaves more open fit coefficients than number of training samples available. To avoid over-fitting it is therefore necessary to either further truncate the number of effective cluster interactions J or to introduce regularization. Selecting the expansion coefficients J has been done using Genetic Algorithms [33] or compressive sensing techniques exploiting the promotion of sparse solutions by the L1 norm [34].

2 Method for Efficient 2D Structure Search

To find low energy configurations for molecules on surfaces we use a three step procedure: At first we build a large list of guess configurations as outlined in Sec. 2.1 - 2.2. In a second step we calculate the energies of a few selected configurations from this list and train the machine learning model described in Sec. 2.3 - 2.7 on these configurations. As a last step we can then use the machine learning model to cheaply predict the energy of all remaining configurations.

2.1 Generating Guess Configurations

When molecules crystallize on a periodic substrate, there are in principle two distinct, periodic lattices: The lattice vectors of the substrate L_s, and the lattice of the adsorbate L_a. The two sets of lattice vectors are related by the so called epitaxy matrix E:

$$L_a = EL_s \tag{44}$$

If E has only integer elements, then the lattice vectors of the adsorbate are an integer multiple of the substrate lattice vectors. Theses structures are called commensurate and have the same periodicity as the adsorbate crystal structure. If the epitaxy matrix has non-integer elements there is no common periodicity for both systems and it is therefore not possible to describe the entire structure with periodic boundary conditions. This difference between commensurate and non-commensurate structures is shown for a 1D example in Fig. 4. Since accurate description of periodic substrates requires periodic boundary conditions, only commensurate structures can be simulated with any bandstructure code package. We can therefore, without loss of generality, restrict our structure search to commensurate structures, i.e. structures where the substrate provides a registry for the adsorbate.

Additionally we assume that the adsorbate-adsorbate interaction is weak compared to the adsorbate-substrate interaction. Under this assumption, molecules in a close-packed monolayer assume geometries that are similar to the geometries of isolated molecules on this substrate. It is therefore possible to build the monolayer structures in a two-step procedure: At first we find the local adsorption geometries that an isolated molecule assumes on the substrate.

For the example of TCNE on Cu(111) eleven different local adsorption geometries can be found, three of which are depicted in Fig. 5. In a second step we combine these local adsorption geometries to larger structures with multiple molecules per unit cell. A systematic method for exhaustively combining the local adsorption geometries to larger structures is described in Sec. 2.2. When the inter-molecular interactions are weak, this combination of isolated geometries provides a good guess structure that can be further refined using traditional, local geometry optimization.

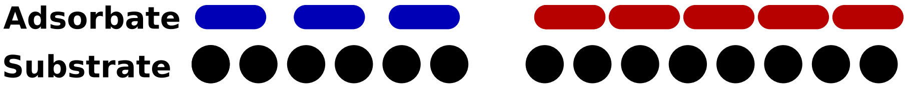


Figure 4: *The left, blue adsorbate has an epitaxy "matrix" of $E = 2$: The system is periodic every two substrate atoms. The right, red, adsorbate has an epitaxy matrix of $E = \frac{\pi}{2}$: There is no common periodic unit-cell.*

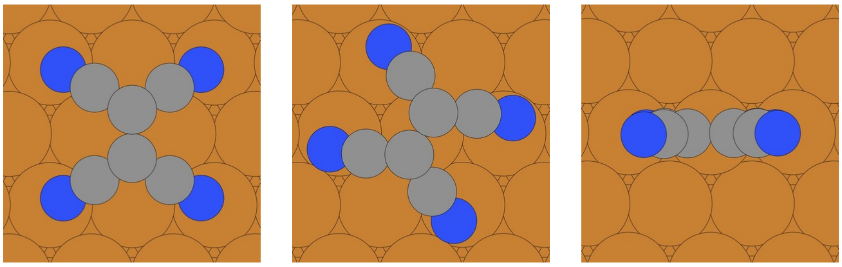


Figure 5: *Three out of eleven distinct local adsorption geometries of TCNE on Cu(111): The first two molecules lie flat on the surface, the third one is standing upright.*

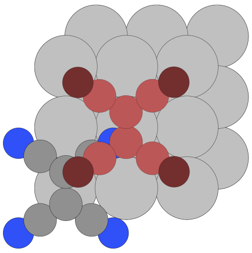


Figure 6: *Collision between two TCNE molecules on an Ag surface: The red molecule cannot be placed on its position because it is too close to the other TCNE molecule already present.*

2.2 Building Guess Polymorphs: TETRIS

When constructing guess polymorphs, a lot of configurations can immediately be dismissed due to collisions between adsorbates as seen in Fig. 6. To only generate configurations that are physically plausible the algorithm depicted in Fig. 7 is used.

Like in the 80s video game TETRIS, we build structures out of a few basic building blocks. As in TETRIS, we can rotate our molecules by discrete amounts and obtain different, but symmetrically equivalent geometries. Combining all these symmetrically equivalent geometries builds our complete list of local adsorption geometries. To exhaustively build a list of all possible configurations, we build up a tree (depicted in Fig. 8) that contains an increasing number of molecules per unit cell at every layer. The first layer consists of all configurations with one molecule per unit cell, the second layer all configurations with two molecules per unit cell and so forth. Each node contains a list of all possible configurations consisting of the given geometry indices. Consecutive layers of the tree are built by adding molecules to configurations from previous layers. When adding geometries to a configuration from a previous layer, only geometries having an index larger or equal to the highest geometry already present in the configuration are added. This ensures that the list of geometries contained is always ordered and therefore excludes permutations of the geometries that would lead to the same structure. When adding a molecule to a parent configuration, it is checked that the new molecule does not collide with any of the molecules already being present in the parent configuration. If a collision is detected this configuration is discarded. To speed up these collision checks, a table of all possible collisions between two molecules is precalculated so that the collision check is only a lookup within this dictionary. Finding the possible configurations of an individual node is independent of all other nodes within this layer and only depends on the parent node, therefore building the layers can easily be parallelized.

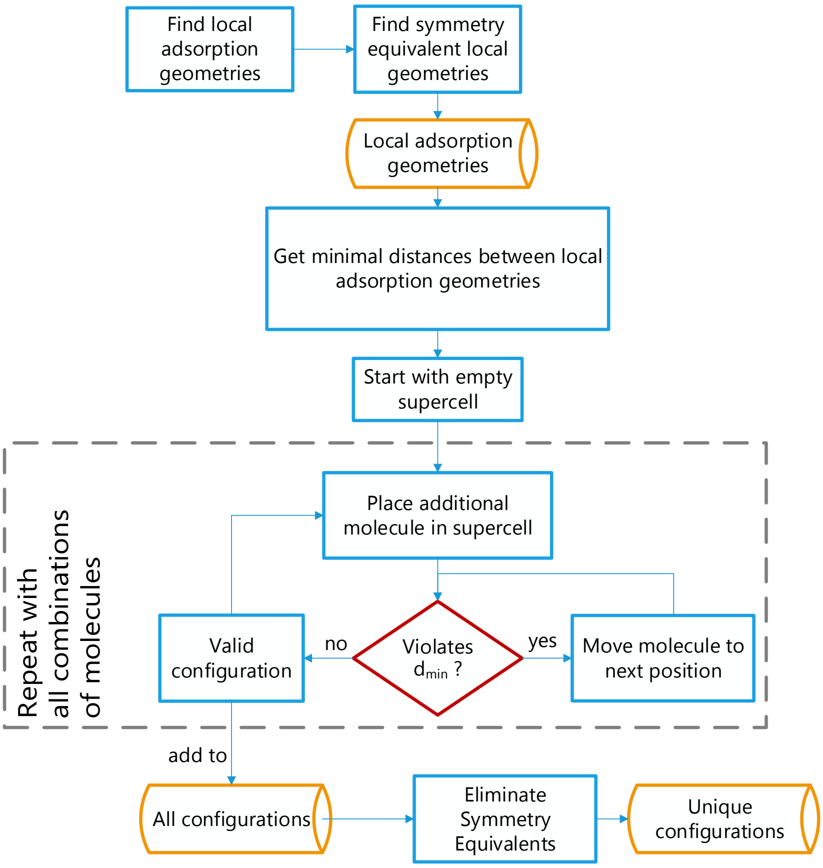

Find local adsorption geometries
Find symmetry equivalent local geometries
Local adsorption geometries
Get minimal distances between local adsorption geometries
Start with empty supercell
Repeat with all combinations of molecules
Place additional molecule in supercell
Valid configuration
no
Violates d_{min} ?
yes
Move molecule to next position
add to
All configurations
Eliminate Symmetry Equivalents
Unique configurations

Figure 7: *Flowchart for generating configurations using the "TETRIS" algorithm.*

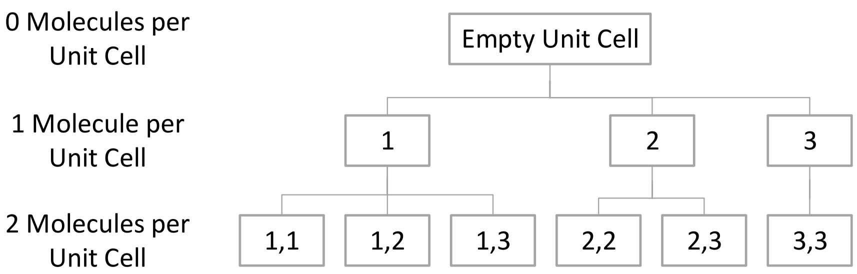


Figure 8: *Search tree for the TETRIS algorithm. Each box represents a sublist of configurations, containing fixed geometry indices. Each node only adds geometry indices that are larger or equal to the largest parent geometry index.*

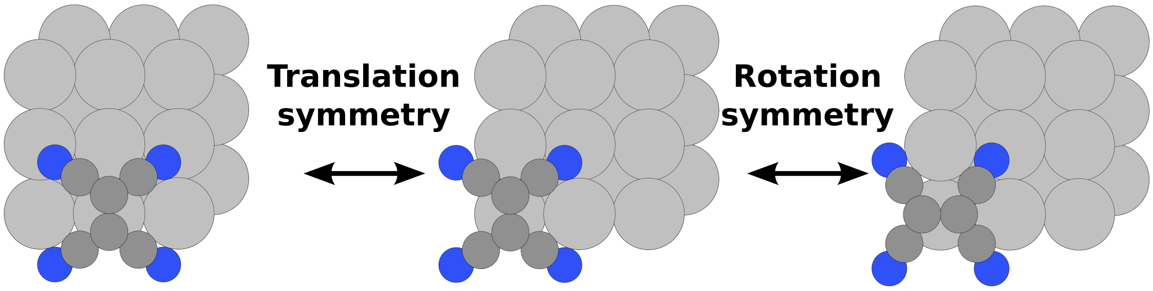


Figure 9: *Examples of three structures that are equivalent by rotational or translational symmetry.*

Several configurations that are found in this way might be equivalent by symmetry (Fig. 9). In particular translations, as well as rotational and inversion symmetry are exploited to reduce the total number of configurations. Due to the periodicity of the substrate, for every given structure there exist many other adsorbate structures that only differ by translation of all molecules by a multiple of the substrate lattice vectors (e.g. Fig. 9). To uniquely choose one structure among all its symmetry equivalent siblings a unique identifier called the “hash” is introduced. The hash is a list of numbers that fully determines a configuration. It consists of the shape of the substrate unit cell - given as a 2x2 epitaxy matrix - the indices of the local adsorption geometries used, as well as their positions. If multiple equivalent structures exist, the one that yields the smallest hash (interpreted as an integer number) is chosen. By applying all possible symmetry operations to all configurations found and selecting the equivalent configuration which minimizes the hash, a list of all symmetrically distinct configurations can be built.

The TETRIS procedure of eliminating colliding structures and exploiting substrate symmetries can in many cases reduce the configurational space by orders of magnitude, compared to a brute-force approach of putting every molecule in every rotation on every possible position. In particular eliminating colliding structures vastly decreases the search space as can be seen in Fig. 10. As more and more molecules are put into the unit cell the number of possible configurations rises exponentially at first, causing the straight line in the log-plot. But with increasing coverage more and more collisions eliminate configurations, decreasing the final number of configurations.

2.3 The Energy Model

Although the TETRIS approach outlined in 2.2 significantly reduces the size of the configurational space, compared to naively trying every combination of molecules on every lattice position, it is still unfeasible to exhaustively calculate the energies of all configurations using accurate quantum mechanical calculations. At this point, one could in principle resort to computationally cheaper methods such as force field based approaches, but since these models are usually not parametrized for the specific system at hand, their accuracy is not sufficient to find the subtle differences between various low energy structures. Instead, we want to perform a few, selected, highly accurate calculations and use the results to train a model that can then cheaply predict the energy of all remaining configurations. In this work, the high accuracy calculations were done using van der Waals corrected Density Functional Theory (DFT), as outlined in appendix A.1, but any other method that provides accurate energies for given configurations can be used as well. There are several properties that we would like to see in our energy model:

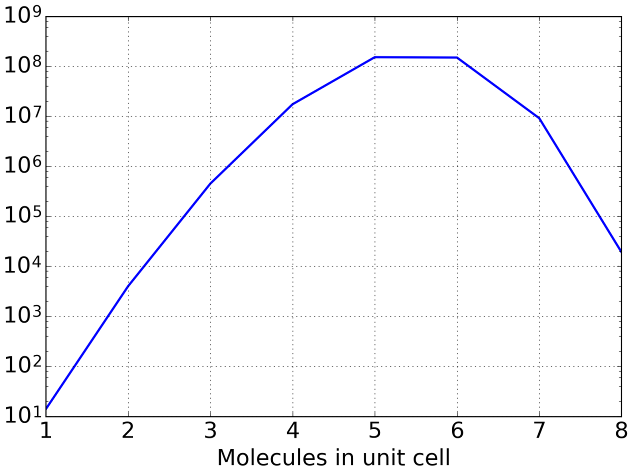


Figure 10: *Number of configurations for TCNE in a Ag(100) unit cell with 60 surface atoms and a diagonal epitaxy matrix of* 15×3*. With increasing number of molecules per unit cell the number of possible configurations initially rises exponentially (note the logarithmic scale) but is then decreasing again as configurations are ruled out due to collisions.*

- **Cheap evaluation**: Making predictions with the model should be computationally significantly cheaper than calculating the energy with the input method, e.g. DFT.
- **Accuracy**: Given enough training data, the model should be able to predict energies with an accuracy that is close to the accuracy supplied by the input method.
- **Efficiency**: The method should achieve this accuracy given as little training data as possible, thus minimizing computational effort for computing the training energies.
- **General Model**: The model should work for a wide class of molecules and substrate and not be specific to certain molecules or certain predominant interactions.
- **Adaptive Learning**: It should be possible for the model to incorporate the energies of all configurations that are calculated, including additional results after the initial training phase. Training the model should not require specific calculations that are only used for training but should preferably use energies of real configurations.
- **Interpretability**: The results and fitting parameters of the model should be human interpretable to allow the user to generate insights for a wider class of materials.

There is a large variety of possible energy models that could be used: Some simple models - such as an electrostatic model with fitted charges, dipole moments, etc. - can be readily interpreted but might not offer the flexibility to accurately predict energies of systems with varying or unknown interactions. Complex models, such as neural networks, have been shown to be highly accurate given enough training data, but their parameters, the input weights within the network, are difficult to interpret and they suffer from overfitting when trained on small datasets.

For this work a middle ground is chosen: We combine a well interpretable model, pairwise interactions between molecules, with a highly flexible model to describe these interactions. To model the energy of any configuration, we split the interaction into two parts: The interaction between the adsorbates and the substrate E_1 and the pairwise interactions between the adsorbates E_2.

$$E_{config} = \sum_{\substack{adsorbates\\a}} E_1^a + \sum_{\substack{pairs\\p}} E_2^p \tag{45}$$

Note that to fully take into account all possible interactions one must in principle also include higher order terms, like triplets of molecules. But for the systems investigated, we show in 4.7 that pairwise interactions suffice to describe the energy of the configurations with high accuracy.

The interaction strengths E_1, E_2 are a not known a priori. One could try to model them using a parametrized physical model, for example electrostatics, but to keep the model flexible we leave all interactions open as fit coefficients. If a certain interaction appears n times within a configuration, the total energy of a configuration is given by:

$$E_{config} = \sum_{\substack{adsorption\\geometries\\a}} n_a E_a + \sum_{\substack{pairs\\p}} n_p E_p \tag{46}$$

In Eq. 46 the sums now run over all possible adsorption geometries and all possible pairs of molecules. To limit the second sum to a finite number of terms, only pairs within a certain cutoff radius d_{max} are included. By combining the fit coefficients E_1 and E_2 into a vector of fit coefficients $\boldsymbol{E}$ and the multiplicities n into a matrix N, the energies for multiple configurations can be written in compact form:

$$\boldsymbol{E}_{configs} = N\boldsymbol{E} \tag{47}$$

If there are n_c configurations and n_E fit coefficients then N is a matrix of shape $[n_c \times n_E]$. Since for any given configuration usually only a few interactions contribute, the matrix N is mostly sparse.

2.4 Naïve Least-Square Inversion

Equation 47 is an inhomogeneous linear system of equations with known energies $\boldsymbol{E}_{configs}$ and unknown energies $\boldsymbol{E}$. In principle one could obtain a least squares fit for $\boldsymbol{E}$ by minimizing the residual:

$$\min_{\boldsymbol{E}} \|\boldsymbol{E}_{configs} - N\boldsymbol{E}\|^2 \tag{48}$$

This approach is problematic since obtaining n_E fit coefficients requires at least n_E data points. For a typical case there will be on the order of a thousand different pairwise interactions, thus requiring a thousand DFT calculations. While this can already be a large improvement over the millions of configurations that we started with, it is still unsatisfying. Furthermore, if the system of equations is ill-conditioned, small errors in the computed energies $\boldsymbol{E_{configs}}$ will lead to large errors in the interaction energies $\boldsymbol{E}$.

2.5 Introducing Prior Knowledge

Although Eq. 47 has in principle n_E fit coefficients, we can introduce additional (prior) knowledge that allows to solve the system, despite having fewer than n_E data points. Even without knowledge about the specific interactions, there are very general assumptions that can be made, which should hold for a large class of materials:

1. **Interaction decay**: Interactions decay to zero at large distances: If molecules are in close proximity their interactions might be large or small, but as the distance between the molecules increases, the interaction must decay to zero.
2. **Smoothness**: The potential energy surface will be somewhat smooth. In particular if molecules are far apart, small variations in their distance and orientation will only cause small changes in the interaction.
3. **Molecule-substrate interactions**: Molecule-substrate interactions in a densely packed monolayer will be similar to the adsorption energy at low coverage. While surrounding molecules will somewhat change the adsorption energy - for example via depolarization - it will still be relatively similar to the original adsorption energy at low coverage.

2.6 Gaussian Process Regression

This prior knowledge can be incorporated into the fit using Gaussian Process Regression (GPR). GPR builds on Bayes' theorem:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{49}$$

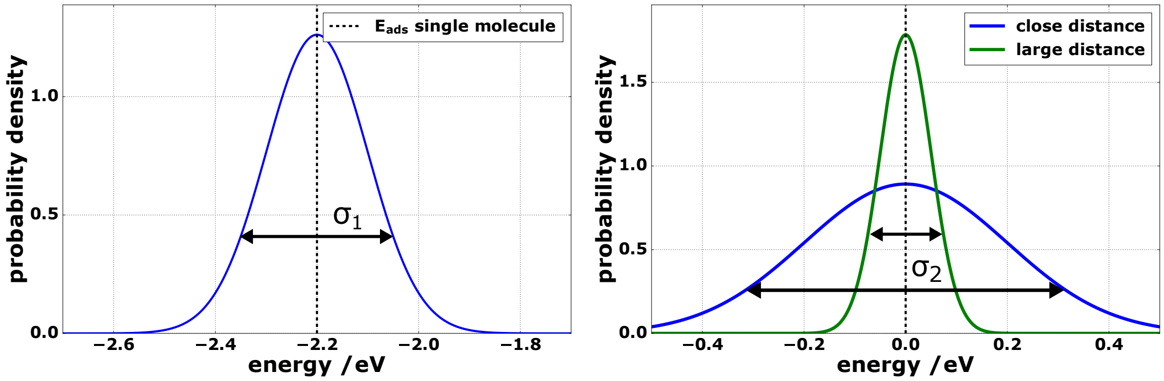
Here $p(A|B)$ denotes the probability for A given B, called the *a posteriori* probability. The term $p(B|A)$ is the probability to obtain B given A, and is called the likelihood. The term $p(A)$ is the probability for A without any additional information and is therefore called the *prior* probability. The denominator $p(B)$ serves to normalize the probability and is of little relevance for us: We will from now on only refer to it as $\mathcal{Z}$. For our application, the probability distribution of the fit coefficients' values $\boldsymbol{E}$, given the energies of some observed configurations $\boldsymbol{E_{configs}}$, is thus given by:

$$p(\boldsymbol{E}|\boldsymbol{E_{configs}}) = \frac{1}{\mathcal{Z}} p(\boldsymbol{E_{configs}}|\boldsymbol{E})\, p(\boldsymbol{E}) \tag{50}$$

Once we have calculated the distribution $p(\boldsymbol{E}|\boldsymbol{E_{configs}})$ we can find the most likely values for the fit coefficients $\boldsymbol{E}$. The probability distributions on the right hand side, the likelihood and the prior, are not known exactly but we can model them using multivariate Gaussian distributions. The likelihood is the probability to obtain certain energies for configurations, assuming we knew the fit coefficients $\boldsymbol{E}$. Since our energy model is not exact - because it only includes interactions to the substrate and pairwise interactions - the results we obtain using DFT will deviate from the results of our model. The likelihood is therefore expressed as:

$$p(\boldsymbol{E_{config}}|\boldsymbol{E}) \propto e^{-\frac{1}{2\sigma_M^2}\left\|\boldsymbol{E_{config}}-N\boldsymbol{E}\right\|^2} \tag{51}$$

The likelihood is maximal for $\boldsymbol{E_{config}}$ being exactly as predicted by the model, but still yields large probabilities as long as the residual is small compared to the expected uncertainty of the model σ_M. For the systems investigated typical values are on the order of $\sigma_M = 1 - 10$ meV.



(a) *The prior is peaked around the adsorption energy for a single molecule.*

(b) *The prior is centered around zero and broad for pairs at close distances, while narrow for long distances.*

Figure 11: *Prior energy distributions for molecule-substrate-interactions (a) and molecule-molecule-interactions (b).*

Also the prior is assumed to be a multivariate Gaussian distribution, which in the most general case is given by:

$$p(\boldsymbol{E}) \propto e^{-\frac{1}{2}(\boldsymbol{E}-\boldsymbol{E_p})^T C^{-1}(\boldsymbol{E}-\boldsymbol{E_p})} \tag{52}$$

The prior in Eq. 52 is fully specified by its mean values $\boldsymbol{E_p}$ and covariance matrix C. Using this mean and covariance we can encode the general prior assumptions we stated in 2.5. The mean values $\boldsymbol{E_p}$ encode our best guess for the fit coefficients before including any data from the periodic DFT calculations. Since we assume no particular type of interaction we do a-priori not know any interaction strengths, not even whether they are repulsive (positive) or attractive (negative). We therefore choose the prior mean $\boldsymbol{E_p}$ to be zero for all pairwise interactions.

For the molecule-substrate interactions however we have a reasonable guess - the adsorption energies obtained for a single molecule - and we can therefore encode assumption 3 by setting $\boldsymbol{E_p} = E_{ads}$ for the entries of the vector that describe the molecule-substrate interactions (see Fig. 11a). The diagonal of the covariance matrix contains our uncertainty of our prior guess for each fit coefficient. For the molecule-substrate interactions we are not certain that they will exactly be the same as in the isolated case ($E_p = E_{ads}$) but we assign a certain energy width σ_1 to the probability distribution.

Implementing assumption 1 - interactions decay to zero at large distances - can be done by choosing a different variance for each fit coefficient. For fit coefficients that describe long-distance interactions, where we are certain that the interaction will be close to zero, we can thus assign a small variance to it (see Fig. 11b). For fit coefficients that describe short distance interactions we expect potentially large values and are thus very uncertain that our initial guess $\boldsymbol{E} = \boldsymbol{0}$ is correct: we assign a large variance to it. For our implementation we used an exponentially decaying prior variance as a function of inter-molecular distance d:

$$C_{ii} = \left(\sigma_2 e^{-\frac{d_i - d_{min}}{\lambda}}\right)^2 \tag{53}$$

Here σ_2 is the typical interaction strength we expect at close packing, when the molecules have a distance of d_{min} and λ is the typical length scale on which the interactions decay to zero. For the investigated system of TCNE, typical values for these priors are interaction strengths of $\sigma_2 \approx 100\ meV$ at $d = 2.6$ Å and decay lengths of $\lambda \approx 3$ Å.

The remaining values to be specified are the off-diagonal elements of C which describe the correlation between pairs of interactions. Assumption 2 vaguely states that "similar pairs of molecules have similar interactions". To put this more formally we need a distance measure between pairs of molecules. This distance is not a physical distance, e.g. measured in Å, but rather a mathematical concept to measure the dissimilarity of two pairs of molecules. Given such a distance r, we can express the off-diagonal elements of the covariance matrix C as:

$$C_{ij} = \sqrt{C_{ii}C_{jj}}e^{-\frac{r}{\alpha}} \tag{54}$$

If pairs of molecules are very similar and thus have a distance $r \to 0$, the correlation will become maximal. The interactions will be fully correlated, meaning that large values of one interaction also necessitate large values of the other interaction. There is no unique way to construct such a distance r and the choice of distance function does have a crucial impact on the machine learning. One possible way to build such a distance function is discussed in Sec. 2.7.

Now that prior and likelihood are specified, Eq. 50 can be used to calculate the posterior distribution. Combining Eq. 51 and 52 yields:

$$p(\boldsymbol{E}|\boldsymbol{E_{configs}}) \propto e^{-\frac{1}{2\sigma_M^2}\left(\boldsymbol{E_{config}}-N\boldsymbol{E}\right)^T\left(\boldsymbol{E_{config}}-N\boldsymbol{E}\right)}e^{-\frac{1}{2}(\boldsymbol{E}-\boldsymbol{E_p})^TC^{-1}(\boldsymbol{E}-\boldsymbol{E_p})} \tag{55}$$

Calculating the logarithm of the probability and expanding the squares yields:

$$\begin{aligned}
\log p(\boldsymbol{E}|\boldsymbol{E_{configs}}) &= -\frac{1}{2}\left[\frac{1}{\sigma_M^2}(\boldsymbol{E_{config}} - N\boldsymbol{E})^T(\boldsymbol{E_{config}} - N\boldsymbol{E}) + (\boldsymbol{E} - \boldsymbol{E_p})^TC^{-1}(\boldsymbol{E} - \boldsymbol{E_p})\right] \\
&= -\frac{1}{2}\left[\boldsymbol{E}^T\left(\frac{N^TN}{\sigma_M^2} + C^{-1}\right)\boldsymbol{E} - 2\left(\boldsymbol{E_p}^TC^{-1} + \frac{\boldsymbol{E}_{\boldsymbol{config}}^TN}{\sigma_M^2}\right)\boldsymbol{E} + \right. \\
&\qquad \left. + \frac{1}{\sigma_M^2}\boldsymbol{E}_{\boldsymbol{config}}^T\boldsymbol{E_{config}}\right] \\
&= -\frac{1}{2}\left[(\boldsymbol{E} - \boldsymbol{E_{int}})^TA^{-1}(\boldsymbol{E} - \boldsymbol{E_{int}})\right] + R
\end{aligned}$$

Here R contains all the terms that do not depend on E and can therefore be moved into a new normalization $\mathcal{Z}$. The posterior distribution for the fit coefficients $\boldsymbol{E}$ is thus again a Gaussian distribution with mean interaction energies $\boldsymbol{E_{int}}$ and covariance A.

$$A^{-1} = \frac{N^TN}{\sigma_M^2} + C^{-1} \tag{56}$$

$$\boldsymbol{E_{int}} = A\left(C^{-1}\boldsymbol{E_p} + \frac{N^T\boldsymbol{E_{config}}}{\sigma_M^2}\right) \tag{57}$$

$\boldsymbol{E_{int}}$ is therefore the most likely set of fit parameters given the prior knowledge and input data. Once the interaction energies $\boldsymbol{E_{int}}$ are known, the energies for unknown configurations, can immediately be calculated. Just as the training samples were described by the occurring interactions via the matrix N, the new configurations can be described by a matrix $\tilde{N}$. Just as for the test data, each row corresponds to one configuration and each column lists how often a

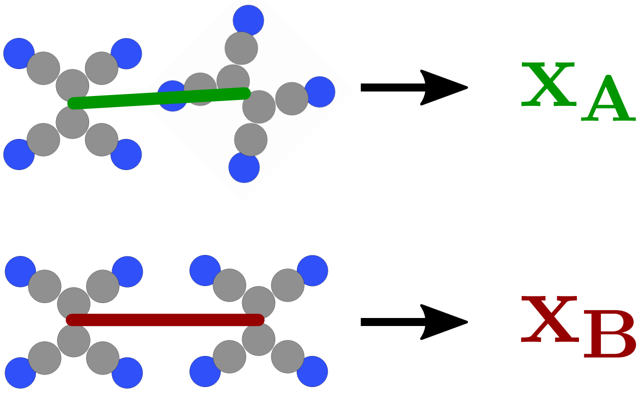


Figure 12: *Similarity between two pairs of molecules, A and B, is calculated as the distance between their feature vectors $\vec{x}$.*

specific interaction occurs within each configuration. The energies for the configurations to be predicted can then simply be calculated via:

$$\boldsymbol{E_{predict}} = \tilde{N}\boldsymbol{E_{int}} \tag{58}$$

2.7 Feature Vectors

To be able to enforce smoothness of the inter-molecular potential energy surface it is necessary to obtain a numerical representation for each pair of molecules. Once a pair of molecules is numerically represented by a feature vector $\boldsymbol{x}$, the distance between two different pairs of molecules, A and B, can be calculated with any norm, such as the L_1 norm $r = |\boldsymbol{x_A} - \boldsymbol{x_B}|$. This idea is sketched in Fig. 12.

There are several properties which are desirable for the numeric representation of a molecular pair:

- **Uniqueness**: Pairs that are physically different should map to different feature vectors.
- **Symmetries**: Pairs that are equivalent by symmetry should map to the same feature vectors. These symmetries include rotation and inversion, but also relabeling the atoms within the molecules. Changing the order in which the atoms are stored should not impact the feature vector.
- **Smoothness**: Similar pairs should map to similar geometries. Discontinuities in the representation can deteriorate the fit quality.

One possible feature vector that has been used for the representation of molecules is the Coulomb Matrix [35]. In this representation all inter-atomic distances r_{ij} are calculated and the molecule is represented as the matrix X:

$$X_{ij} = \begin{cases} \frac{Z_i Z_j}{r_{ij}} & \text{for } \ i \neq j \\ 0.5 Z_i^{2.4} & \text{for } \ i = j \end{cases} \tag{59}$$

This has been applied successfully to predict the atomization energies of small molecules [35]. Since unitary transformations (inversion, rotation) do not affect distances, the Coulomb Matrix

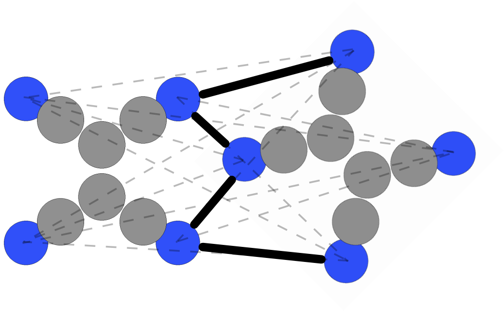


Figure 13: *The feature vector is built as a list of inter-molecular atom distances. For TCNE we only consider the distances between the (blue) nitrogen atoms.*

is invariant under these symmetries. However reordering the atoms within the molecule does also reorder the entries of the matrix and thus causes an unwanted change in the feature vector. This problem has been addressed by various means: One can sort the matrix, for example by the norm of the rows, to ensure that the final feature vector is independent of the original order of atoms. This does however introduce discontinuities whenever the sorting order changes. One can also augment the training data by training the model with multiple, randomly sorted Coulomb Matrices which has been shown to be effective in combination with algorithms that deal well with large training sets, such as artificial neural networks [32].

To represent pairwise interactions between two molecules, A and B, we use a representation similar to the Coulomb Matrix: We compute the inter-atomic distances between the atoms of molecule A and the atoms of molecule B. We do not include the interatomic distances within each individual molecule since these do not carry information about the relative arrangement of the molecules. For the investigated molecule of TCNE we also only compute the distances between the four “cornerstones” of the molecule, which are the four nitrogen atoms, as depicted in Fig. 13. The distances are than sorted to avoid the aforementioned ambiguity that arises from permuting atoms. This yields a vector of $4 \times 4 = 16$ elements. Instead of choosing r^{-1} as a feature vector, we choose r^{-2} which empirically did improve the prediction accuracy.

Choosing the feature vector to be proportional to a negative power of the interatomic distances is useful, because it yields high resolution at small distances - small changes in the distance cause large changes in the feature vector - and more smoothness at larger distances. As a last step we optionally truncate the feature vector from its full size of 16 elements to a feature vector length of smaller length. In some cases this could be useful since the relative arrangement of the molecules is already well determined using only some of the smallest inter-atomic distances and additional elements in the feature vector might rather add noise than useful information.

2.8 Choosing the Training Samples

In traditional machine learning applications the training of the model is done after a training dataset has been acquired. This is in particular the case when benchmarking new machine learning models on existing datasets, such as the MNIST database for image classification of handwriting or the QM7 dataset for the atomization energies of small molecules. On the contrary, when searching for low energy structures of a specific system, training data is usually not available and must be supplied by the user. The training structures could be chosen randomly out of all possible structures to avoid biasing the machine learning algorithm towards a certain model. However it might be advantageous to use the liberty of choice of training set to explicitly select the data-points which offer the “highest gain of information” and are thus most useful for the following machine learning procedure. To put this more formally we define a loss function that we aim to minimize by systematically selecting highly informative training samples.

The goal of Gaussian Process Regression for our application is to accurately estimate the fit coefficients $\boldsymbol{E_{int}}$ which in turn will allow accurate prediction of energies of configurations. The uncertainty of the fit coefficients is given by the covariance matrix A of the posterior distribution (Eq. 56). Minimizing A (or alternatively maximizing A^{-1}) will therefore lead to a small uncertainty for the fit parameters $\boldsymbol{E_{int}}$. Since A is a matrix, there is no unique definition for a loss function, but instead there is a variety of different optimality-criteria in the field of Optimal Design Theory. Some popular criteria are:

- **A**-optimality: Minimizes the trace of A and therefore minimizes the average of the variance of the fit parameters. The variance for each parameters is given by the diagonal elements of A and thus captured by $tr(A)$.

- **D**-optimality: Minimizes the determinant of A. This criterion also takes into account the off-diagonal elements of A, which contain the correlation between fit coefficients.

- **E**-optimality: Minimizes the largest eigenvalue, corresponding to the direction and size of the maximum uncertainty, of A. Since it only minimizes the largest eigenvalue this criterion focuses on "worst-case" performance.

For this work the criterion of D-optimality was used to select the training data-set. Minimizing the determinant of A by selecting n_{train} structures from a list of N candidates is a challenging task, even for relatively small n_{train} and N. The number of possible choices is given by the binomial coefficient $\binom{N}{n_{train}}$. For a typical example of choosing 50 configurations out of 10 000 possible ones there are approximately 10^{135} possible choices. It has been proven that optimal subset selection for regression is in fact NP-hard [36] but heuristic algorithms can yield satisfactory results for practical applications [37]. One particular method is Fedorov’s algorithm [37], which starts with a random subset selection and iteratively improves this initial guess by greedily swapping training samples if this swap increases the determinant of A^{-1}. The general structure of the algorithm is depicted in Fig. 14.

The core part of the algorithm is to check whether exchanging one row in the matrix N for a different row increases the determinant of $B = A^{-1} = N^T N$. Calculating the determinant of a matrix is computationally demanding, but calculating the relative change that arises from a small change to the matrix can be calculated much more efficiently. If a row $\boldsymbol{r}$ is appended to the matrix N, the matrix B becomes $B' = N^T N + \boldsymbol{r}\boldsymbol{r}^T$. Equivalently, removing a row r' from the matrix N leads to a new matrix $B' = N^T N - \boldsymbol{r}\boldsymbol{r}^T$. Exchanging row $\boldsymbol{r}$ for a different row $\boldsymbol{r'}$ can be interpreted as appending $\boldsymbol{r'}$ and removing $\boldsymbol{r}$. Using the matrix determinant lemma, B' can then be calculated by:

$$|B'| = |B \pm \boldsymbol{r}\boldsymbol{r}^T| = |B| \left(1 \pm \boldsymbol{r}^T B^{-1} \boldsymbol{r}\right) \tag{60}$$

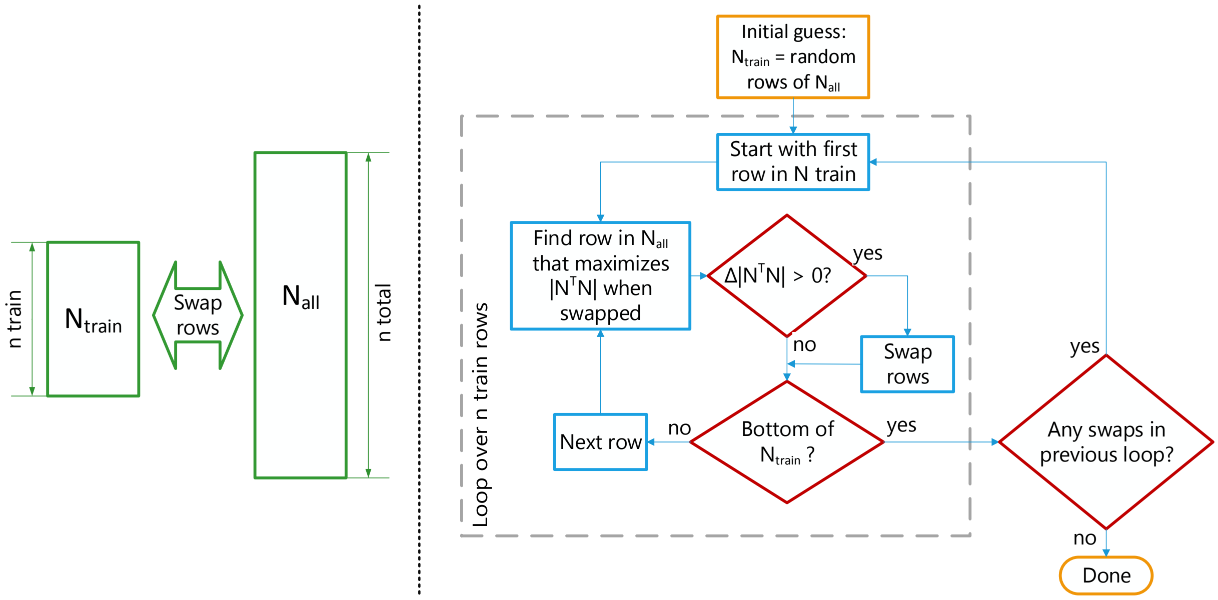

n train
N_{train}
Swap rows
N_{all}
n total
Initial guess: N_{train} = random rows of N_{all}
Start with first row in N train
Find row in N_{all} that maximizes |N^TN| when swapped
Δ|N^TN| > 0?
yes
Swap rows
no
Bottom of N_{train} ?
no
Next row
yes
Any swaps in previous loop?
yes
no
Done
Loop over n train rows

Figure 14: *Flowchart for choosing optimal training samples N_{train} from a matrix of possible training samples N_{all} by maximizing the determinant $|N_{train}^T N_{train}|$. Fedorov's algorithm semi-greedily swaps rows until no further improvement is found.*

Calculating B^{-1} at every step would be expensive, but by using the Sherman-Morrison-Formula also updating the inverse of a matrix can be done cheaply:

$$\left(B + \boldsymbol{r}\boldsymbol{r}^T\right)^{-1} = B^{-1} - \frac{B^{-1}\boldsymbol{r}\boldsymbol{r}^T B^{-1}}{1 + \boldsymbol{r}^T B^{-1}\boldsymbol{r}} \tag{61}$$

Combining Eq. 60 and Eq. 61 allows to evaluate the change of the determinant by simple matrix multiplications. Since in our application any configuration typically only consists of a few interactions, the matrix N is usually sparse, which further accelerates the implementation of Eq. 61.

The algorithm is non-deterministic, because the initial training samples are chosen at random and can converge towards local maxima of the determinant. To improve performance, the algorithm can be restarted with different initial guesses and the best design matrix achieved is chosen for the final selection. When implementing this algorithm in practice, it can be difficult to calculate the initial determinant of the matrix, because the determinant can quickly become too small for standard floating point arithmetic. To avoid this underflow problem one can instead calculate the logarithm of the determinant by summing up the logarithms of the eigenvalues of the determinant:

$$\log |A| = \sum_i \log a_i \tag{62}$$

2.9 Hierarchical Search Strategy

When searching for low energy periodic structures there are several unknowns: The coverage (i.e. the number of molecules per substrate area), the shape of the unit cell (angles and aspect

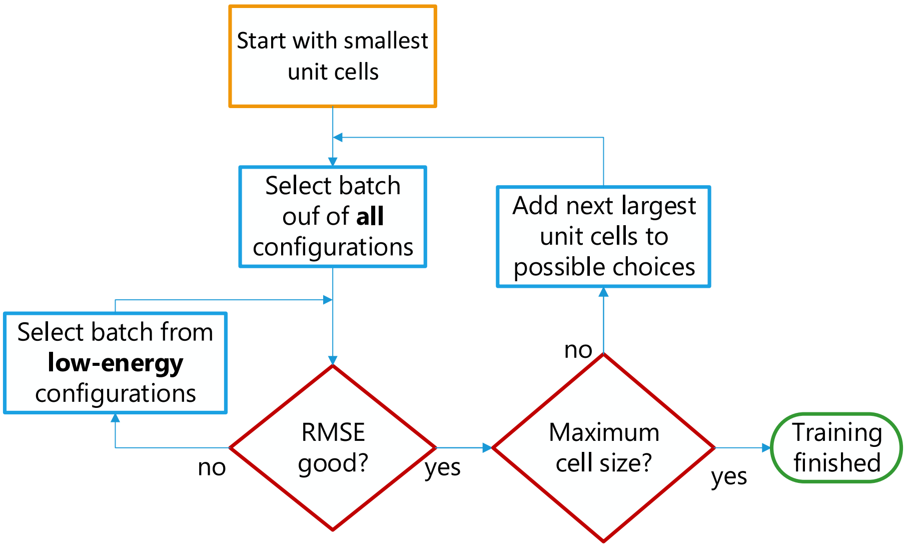

Start with smallest unit cells
Select batch ouf of **all** configurations
Add next largest unit cells to possible choices
Select batch from **low-energy** configurations
RMSE good?
no
yes
Maximum cell size?
no
yes
Training finished

Figure 15: *Flowchart for selecting configurations from increasingly larger unit cells.*

ratio), the size of the unit cell and the positions and orientations of the molecules within the unit cell. In this work the coverage and shape of the unit cell are restricted to values obtained from experiment. Extending the algorithm to arbitrary unit cell however is straightforward.

For example, for the case of TCNE on Ag(100) the search space is restricted to rectangular unit cells that have a coverage of two TCNE molecules per 15 surface substrate atoms. This does still leave a variety of possible unit cells that need to be explored, ranging from small unit cells with few molecules in it, to large cells with many molecules in it. Because smaller unit cells are computationally much cheaper than large unit cells with many atoms in it, it is preferential to generate as much information as possible on smaller unit cells, before moving to cells with many molecules in it.

To systematically search for low energy structures in different unit cells the following search strategy, depicted in Fig. 15 is used:

1. **Initial training**: At first, configurations from the smallest unit cells are selected using Fedorov's algorithm in small batches of about 10 configurations. After each training batch the quality of the fit is assessed by checking the Root Mean Square Error of the prediction.

2. **Unconstrained search**: Once a good prediction accuracy for a set of unit cells is obtained, or the configurational space for small unit cells is exhausted, the next biggest set of unit cells is tackled. We select a training batch of configurations from all possible configurations from this next unit cell and all previous unit cells and continue to monitor progress by calculating the RMSE.

3. **Low energy search**: Because we search for low energy structures, we prefer good prediction accuracy for low energy structures at the expense of higher uncertainty for high energy configurations. To improve the prediction for low energy structures we now only select training data from the configurations that are predicted to be within a fixed threshold of the predicted minimum energy structure. When high accuracy for the current set of unit cells is reached we continue with step 2 on the next biggest set of unit cells.

These steps are iterated up to a given maximum size of the unit cell. At this point high prediction accuracy should be obtained that can now be exploited by evaluating the energy predictions for all configurations.

The rationale behind splitting the training phase in an unconstrained search and a low energy search is a trade-off between exploration and exploitation: Quickly focusing on the low energy structures will often yield faster improvements of the energy predictions for the configurations of interest, but risks oversight of potentially attractive interactions. Therefore a training batch from the complete set of configurations is calculated whenever configurations from a new unit cell shape are included.

Since the energy model is purely aperiodic - it only consists of pairwise interactions, but does not include periodic features - the generated models should perform equally well, independent of the unit cell size. It is therefore possible to train on small, cheap unit cells and make predictions for large, expensive unit cells that were not part of the training set. This possibility to extrapolate the predictions from small unit cells to larger unit cells is extraordinary and is demonstrated for the system of TCNE on Ag(100) in Sec. 5.

The explicit focus on low energy configurations is introduced by restricting the search space for Fedorov's algorithm during selection of the training samples. Another, in principle more elegant way, might be to explicitly encode this importance of low energy configurations into the utility function during training set selection. The current loss function that is minimized during training set selection is the variance of the interaction energies (see Sec. 2.8). It might be advantageous to assign different weights to the interaction energy variances based on their expectation value. Low energy interactions could be assigned a high weight and large energy interactions could be assigned smaller weights, so that the algorithm tries to maximize information gain for low energy interactions which presumably lead to low energy configurations.

Another possibility would be to minimize the variance of the prediction energies, instead of the variance of the model parameters. Also in this case, one could opt for high prediction accuracies for low energy configurations while tolerating larger uncertainties for the less interesting part of the configurational space.

3 Summary of Assumptions

The training procedure outlined in Sec. 2 does at several points make assumptions and choices that can influence the results obtained. The purpose of this short section is to clearly outline the parameters and justify their choice.

3.1 Limitations of the Model

The TETRIS approach to generating guess structures only allows structures that have (unrelaxed) inter-molecular distances larger than a cutoff distance d_{min}. Since it assembles structures out of local adsorption geometries it can also not find structures that are purely stabilized by the molecule-molecule interaction. For example the structure of a classical house of cards could not be found using the TETRIS approach, because the necessary buildings blocks - tilted cards - are not stable local geometries on their own.

The energy model only includes interactions of one molecule with the substrate and pairwise interactions between molecules. It does not include three-body interactions (e.g. one molecule perturbs the interaction between two other molecules) and higher order terms and can therefore never fully model all effects, even when presented with an arbitrarily large training set. However in Sec. 4.7 it can be seen that (at least without substrate) higher order effects are small for the studied systems. Furthermore only interactions within a distance d_{max} are considered by the model.

3.2 Choice of Feature Vector

To encode the pairwise interactions a feature vector is assigned to each pair of molecules. The choice of feature vector implicitly specifies the underlying assumptions about smoothness of the interaction potential. In this work inter-atomic distances have been used to construct the feature vectors but other representations could in principle be used.

Note that when modeling the interaction between molecules we do only include the relative position of the molecules with respect to each other, but do not include the local adsorption sites. For example two parallel molecules sitting at a fixed distance, each on a "top" site, are assumed to have the same interaction as two parallel molecules, both sitting on a "hollow site" as long as their relative position is identical.

On the contrary, for dealing with standing and lying molecules we explicitly break this similarity by effectively constructing three different potential energy surfaces: One for the interaction between lying-lying, one for lying-standing pairs and one for standing-standing molecule pairs. The rationale is that standing molecules might show qualitatively different interactions compared to flat lying molecules, for example due to different charge transfer. We therefore explicitly set to zero any correlations between these three classes of interactions.

When choosing inter-atomic distances, additionally several choices have to be made:

- **Atoms**: The distance between which atoms is used? All inter-atomic distances? Only the "cornerstones" of the molecule? Should different atoms be assigned a different weight in the feature vector? For TCNE we have only used the distances between the nitrogen atoms of different molecules.
- $\mathbf{f(d)}$: When choosing inter-atomic distances d as a features, any function $f(d)$ may be used as well. Decaying functions have been found to be advantageous, because they put

more weight at closer distances where interaction strengths are stronger and the potential energy surface is usually less smooth. When $f(d) = d^{-n}$, the power n has to be chosen.

- **Cutoff**: When generating a list of all feature vectors, many of them will be equivalent by symmetry and the number of fit coefficients can be reduced by removing redundant feature vectors. When $f(d) = d^{-n}$, feature vectors decay to zero at large distances d and are therefore always equivalent (even if not by symmetry) for long distances. A sensible cutoff must therefore be chosen that specifies when to consider two feature vectors to be equivalent.

The feature vector used in this work is currently not immediately transferable to other systems, since it only includes the distances between the cornerstones of the TCNE molecule which are the nitrogen atoms. To apply this method to other systems a modified feature vector must be designed.

3.3 Hyperparameters for Prior Distribution

The design of the covariance matrix as outlined in Sec 2.6 requires several choices of hyperparameters, in particular:

- $\boldsymbol{\sigma_1}$, $\boldsymbol{\sigma_2}$: The expected interaction strengths for 1-body and 2-body interactions respectively. Typically values of 100 meV per TCNE molecule were chosen.

- $\boldsymbol{\sigma_M}$: The expected deviation of the DFT results from the pairwise-interactions-model. For TCNE on metal surfaces this was typically chosen as a few meV. Small values of σ_M put emphasis on exactly fitting the training data, which can lead to over fitting.

- **Interaction decay length** λ: Typical length scale of the pairwise interactions, which for the case of TCNE is typically a few Å.

- **Correlation length** α: Length scale in (normalized) feature space at which pairwise interactions are correlated. For the investigated systems this is typically around 0.1 - 0.5 for features normalized to $[0, 1]$. Large correlation lengths lead to smoother potential energy surfaces.

4 Results for TCNE on Cu(111)

As a first test system we choose to search for the structure of TCNE on Cu(111). As outlined in Sec. 2 we will at first find the local adsorption geometries, then list all possible configurations and train the machine learning model on a small subset calculated via Density Functional Theory. The model will then make predictions that can be validated using further DFT calculations.

4.1 Local Adsorption Geometries on Cu(111)

The TETRIS algorithm needs as input a list of possible local adsorption geometries. In this work the local adsorption geometries were calculated using a brute-force approach, but in principle more advanced methods that have been developed could be used. To find all local energy minima on the Cu(111) surface, geometry optimizations were started from a variety of different starting positions and orientations of the molecule. The following parameters of the starting geometry were varied:

1. **Orientation**: The molecule was put onto the surface in 3 distinct orientations. Lying flat, standing upright with the C=C bond vertical and standing with the C=C bond parallel to the surface.

2. **Position**: The molecule was initially placed on all 4 symmetry distinct adsorption sites: top, bridge, fcc hollow and hcp hollow.

3. **Angle**: The molecule was rotated around the z-axis (perpendicular to the substrate) by 4 different angles: 0°, 15°, 30°, 45°.

This yields a total of 48 starting points for the geometry optimization which was performed in a 6x6 supercell to minimize interactions between molecules of neighboring supercells. The 48 starting points converged towards the 11 distinct local adsorption geometries depicted in Fig. 16. There are 3 flat lying configurations, 4 *wide* (C=C bond parallel to the substrate) standing geometries and 4 *tall* standing geometries (C=C bond perpendicular to the substrate).

The adsorption energies of these single molecules are depicted in Fig. 17. The adsorption is strongest when the molecule lies flat on the surface, because this maximizes the van der Waals interactions with the substrate. Geometries 1 and 2 are especially advantageous because they place the cyano groups directly on top of Cu atoms which act as strong docking sites.

The stability of the found energetic minima was confirmed by applying a small perturbation to the geometries and confirming that a subsequent geometry optimization would indeed lead back to the same minimum.

4.2 Building supercells for TCNE/Cu(111)

To limit the search space we only consider supercells with a coverage of 1 TCNE molecule per 12 surface Cu atoms which was taken from experimental data. Since the local adsorption geometries of flat lying molecules are significantly lower in energy than their standing counterparts we only include the three flat lying geometries and optionally one of the standing geometries into the search procedure. Additionally we limit the supercell shapes to cells that have a diagonal epitaxy matrix and consider supercells with one to three molecules within it. This yields the

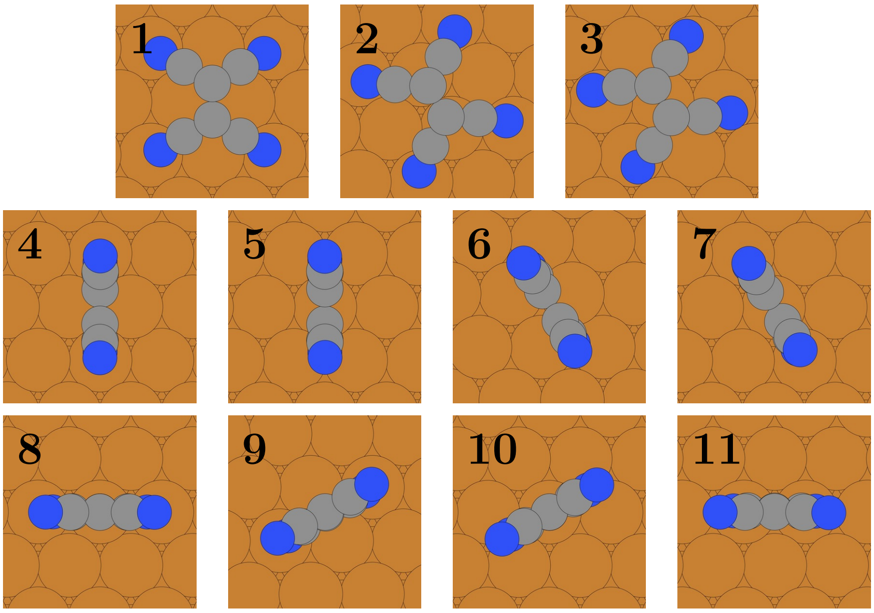


Figure 16: *Local adsorption geometries of TCNE on Cu(111). The geometries in the first row lie flat down, the second row is standing upright with the C=C bond parallel to the surface, the last row has the C=C bond perpendicular to the surface. Adsorption strength (Fig. 17) decreases from top to bottom and from left to right.*

Table 1: *Considered supercells for TCNE on Cu(111). The unit cell shape is given in multiples of the primitive (1x1) substrate unit cell.*

TCNE / supercell	Supercell shapes	Nr of config. (flat only)	Nr of config. (flat + 1 wide)
1	3x4	5	8
2	3x8, 6x4	143	604
3	6x6	218	2873

supercells listed in Tab. 1. For the TETRIS assembly of configurations with multiple molecules per supercell we choose a distance threshold of 2.6 Å between TCNE molecules. This distance is about 10 % smaller than twice the van der Waals radius of nitrogen which is approximately 1.55 Å [38].

4.3 TCNE monolayer in Vacuum

In Sec. 4.8 we will apply the developed method to the problem of TCNE on the Cu(111) metal surface, but to benchmark and study the quality of the energy prediction we will first test it on a computationally much cheaper system: A monolayer of TCNE without a metal substrate. This system does not exist in nature but it is an interesting model system for two reasons: First, it is computationally very cheap since we do not have to calculate the electron density of the substrate. For typical coverages of TCNE on Cu(111) around 95% of the electrons in the system belong to the substrate. Omitting the substrate speeds up the calculation by about a factor of 40x when using the *FHI-aims* DFT code. The second reason are periodic boundary conditions: In order to properly model the metallic substrate, periodic boundary conditions are used. This makes it computationally expensive to accurately calculate the interaction energy

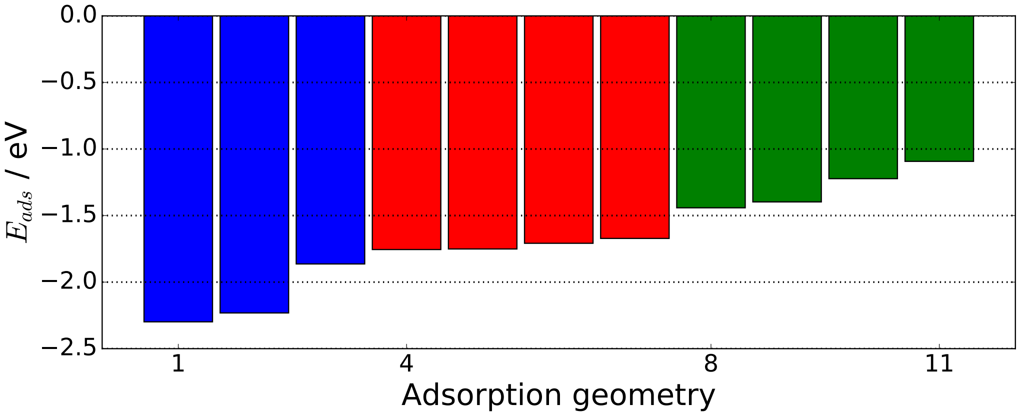


Figure 17: *Adsorption energy of a single molecule for each local adsorption geometry found. Low values correspond to strong adsorption. Blue: flat lying, Red: C=C bond parallel to surface, Green: C=C bond perpendicular to surface.*

between a pair of molecules, since this would require setting up a very large supercell to eliminate the interaction with the periodic replicas in neighboring cells. Once the metallic substrate is removed, there is no fundamental need to use periodic boundary conditions. One can easily do nonperiodic calculations of TCNE-dimers (in the gas phase) and calculate their interaction energies. This allows to directly validate the underlying model that is generated by the fitting procedure, because the fit coefficients directly describe pairwise interaction energies.

To set up the configurations we will start with TCNE on Cu(111). We use the configurations listed in the previous section and simply remove the Cu substrate. We then perform two sets of calculations: A large set of periodic configurations that we will train and validate our machine learning algorithm on. A second set of nonperiodic calculations of TCNE dimers will be used to compare the pairwise energies assigned by the machine learning model to the dimer energies obtained as fit coefficients. In total about 6000 periodic and 5000 nonperiodic single-point DFT calculations were performed with TCNE in vacuum. All DFT calculations were done using the *FHI-aims* code package [39] with the settings outlined in the Appendix A.1.

4.4 Interactions in Vacuum

The model prior assumes a Gaussian distribution of the interaction energies centered around $E = 0$ with a variance that decreases as the distance between molecules increases. To validate this assumption one can analyze the energies of TCNE dimers at various relative orientations and positions. Figure 18 shows the distribution of interaction energies for TCNE dimers for small and large distances. For both long and short distances the distribution is peaked around $E = 0$. As assumed by the prior it is broad for short distances but narrow for larger distances. Overlayed are Gaussian prior distributions for different distances as assumed in the Gaussian Process Regression model. The most notable deviation between the prior assumption and the observed distribution of interaction energies is that the observed distribution for short distances is significantly skewed towards repulsive, positive interaction energies which is not captured in the prior distribution.

As a next step it is of interest to see how well the interaction strengths can actually be inferred by the Gaussian Process Regression model from periodic DFT calculations. To check this we

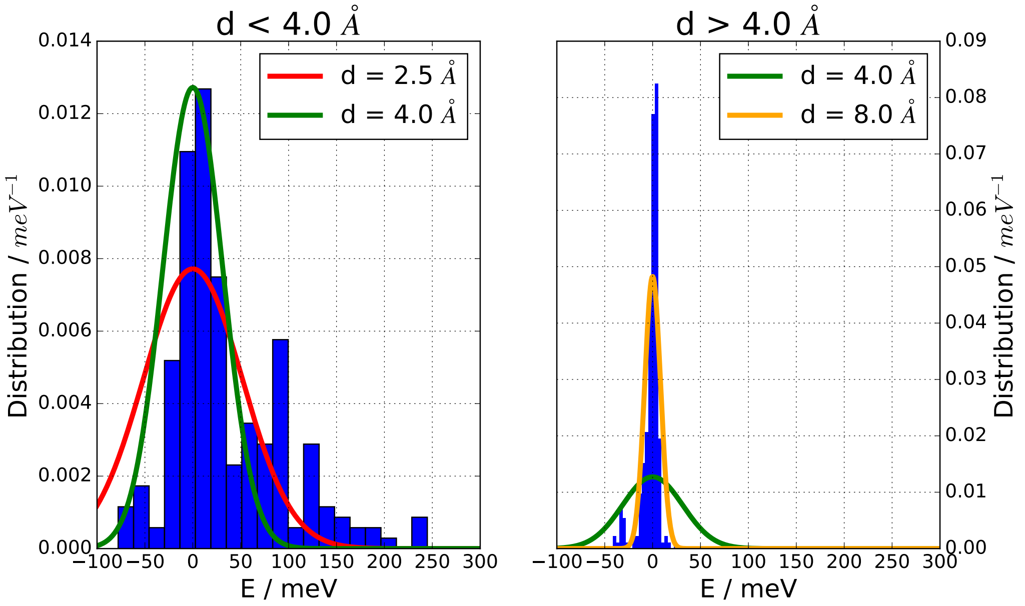


Figure 18: *Distribution of interaction energies in vacuum for pairs of molecules that are close to each other (distance d < 4 Å, left) and far apart (right). Interactions at large distances are clustered around 0, while interaction energies at close distances can have large magnitudes. This fact is encoded in the prior distributions that are plotted for various inter-molecular distances and become narrower with increasing intermolecular distance d*

trained the model on 50 and 300 periodic configurations of a TCNE monolayer sampled from the supercells listed in Tab. 1. The obtained fit coefficients - which represent pairwise interactions between TCNE molecules - are then compared to their reference values obtained by the dimer DFT calculations. This is shown in Fig. 19: For a small training set the model only predicts very few interactions to deviate significantly from its prior assumption of $E = 0$. When adding more training samples the model correctly learns all interaction energies without significant outliers. Note that the model was never directly given any of the interaction energies but only linear combinations of these interactions in the form of energies of periodic configurations.

4.5 Inspection of the Interaction Potential

Every point in Fig. 19 corresponds to one fit coefficient of the model and thus to one pairwise interaction energy. We can use this information to plot the discrete potential energy surface of the interaction between two molecules for fixed rotational angles. This is shown in Fig. 20: One molecule is held fixed at the origin, and a second molecule is moved around, at each point depicting the energy necessary to put the second molecule at this position.

When both molecules lie flat (Fig. 20 - top left) interactions are mainly repulsive, and particularly large when the intermolecular distance between atoms becomes small, justifying the initial assumption for only considering configurations with a certain minimum distance between molecules. When the molecules stand upright and parallel to each other, as in Fig. 20 - top right, large attractive interactions arise. The cause for these interactions is probably a combination of van der Waals and electrostatic interactions. Note that the interaction is particularly favorable in a staggered configuration where the negatively charged cyano-groups come close the positively charged center of the molecule. Bringing the cyano-groups of one molecule close to the

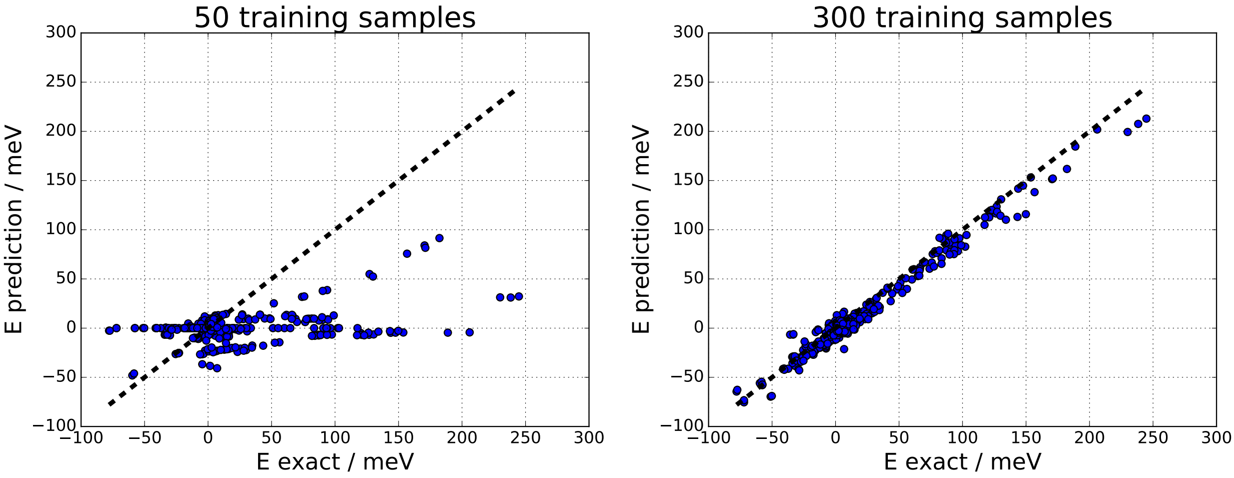


Figure 19: *Pairwise interaction energies in vacuum predicted from periodic calculations compared to reference dimer interaction energies for different training set sizes.*

cyano-groups of another molecule leads to strong electrostatic repulsion as can be seen in the top right figure as well as the bottom right figure where repulsive interactions of 200 meV are observed. Also in the bottom figures it is consistently observed that interactions mainly depend on the relative positions of the cyano-groups. These observations on the one hand explain why the chosen type of feature vector performs well in practice, but also hints at the possibility of improved fit convergence by including an electrostatic model as the prior assumption for the pairwise interactions.

4.6 Cross-Validation of Periodic Energies

Finally one can take a look at the actual problem of interest: Given the energies of n_{train} out of N_{total} periodic configurations, how well can we predict the remaining energies? This is shown in Fig. 21 where the predicted energies for configurations are plotted against the energies calculated by DFT, both for the data that was selected for training (red), as well as all other samples that were used for validation (blue). Ideally we want all points to fall onto the dashed line were the prediction energy equals the first-principles energy. The top left image of Fig. 21 shows the results obtained using zero training data and therefore relying pureley on the prior mean information: Local adsorption energies for the 1-body terms and zero for the interaction energies. As expected this leads to a very poor prediction accuracy. As the number of training samples rises, the fit quality improves, until a near perfect fit is reached using 200 training samples, which only make up about 3 % of the entire sample population. This rapid improvement in fit quality can also be seen in Fig. 22 where the Root Mean Square Error (RMSE) of the prediction is plotted as a function of training sample size. The RMSE is calculated as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(E_{\mathrm{predict},i} - E_{\mathrm{DFT},i}\right)^2} \tag{63}$$

Depicted in red is the RMSE for the data points that were used during training. The prediction error is low, which is unsurprising, since the model has already seen these data points. More interesting however is the RMSE for the testing data that were not presented to the model

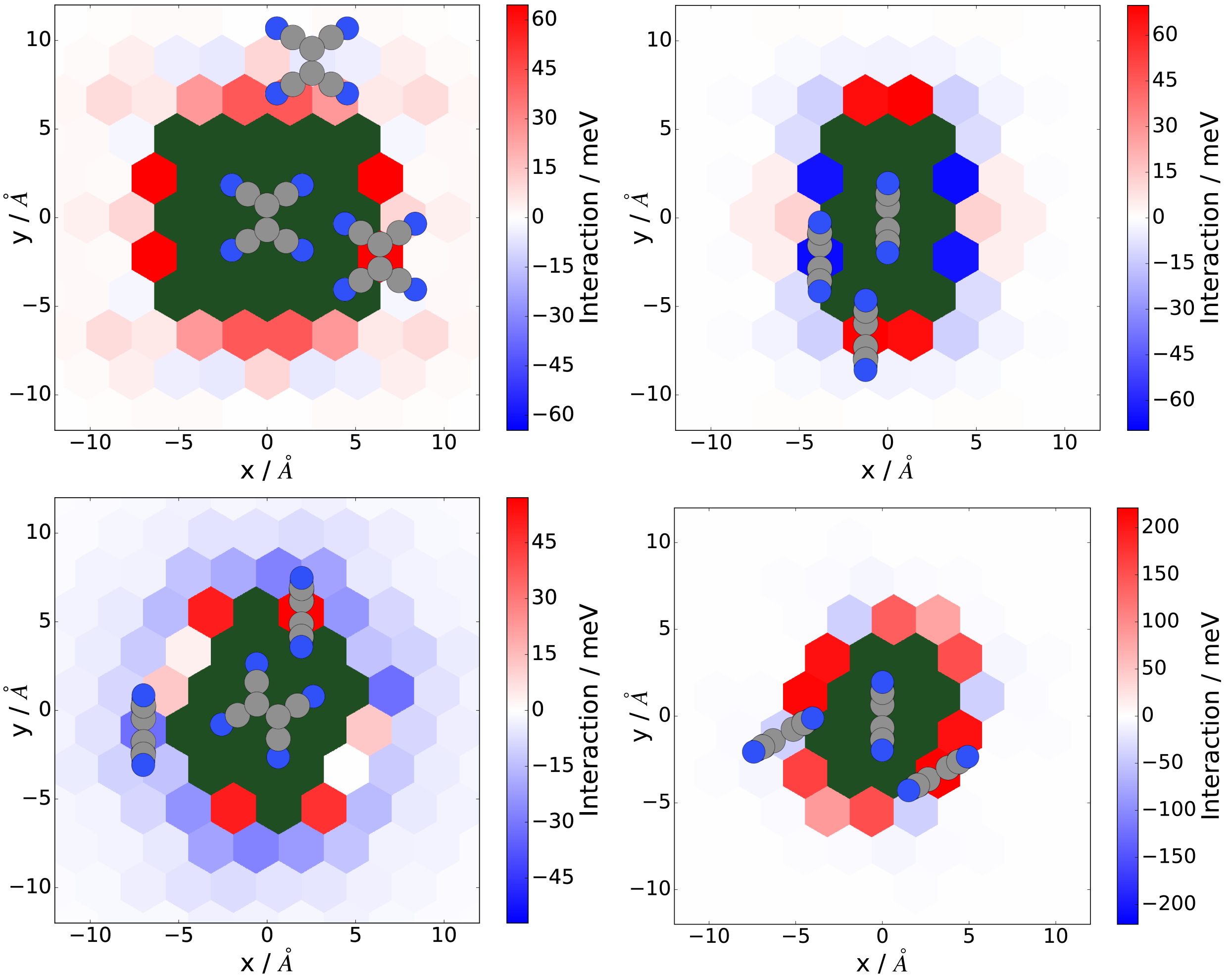


Figure 20: *Pairwise potential energy surfaces. These images show the interaction energy between two molecules as a function of their relative position: One molecule is held fixed at the origin a second molecule is (virtually) moved around it. Their interaction energy is depicted by the color-code. Additional molecules are drawn for the position of maximum and minimum energetic position of the second molecule. Note the varying energy scale.*

during training. It can be seen that the mean error per molecule drops rapidly as the number of training samples rises and levels off at a few meV. As a reference scale, thermal energy k_BT at room temperature (300 K) is about 25 meV.

4.7 Validation of Pairwise Interaction Model

Fig. 21 shows that high prediction accuracy can be obtained using about 200 training samples. When using many training samples the accuracy will ultimately be limited by a basic assumption of the model: the assumption that the energy of any configuration can be expressed as a sum of 1-body and 2-body interactions. For this model system of a TCNE monolayer without substrate all pairwise interaction energies can also be calculated directly using DFT. When we split a periodic configurations into pairs of molecules and sum up their interaction energies from the dimer calculations (as indicated in Fig. 23), how close can we come to the DFT energy from the periodic calculations? The answer to this question is depicted in Fig. 24.

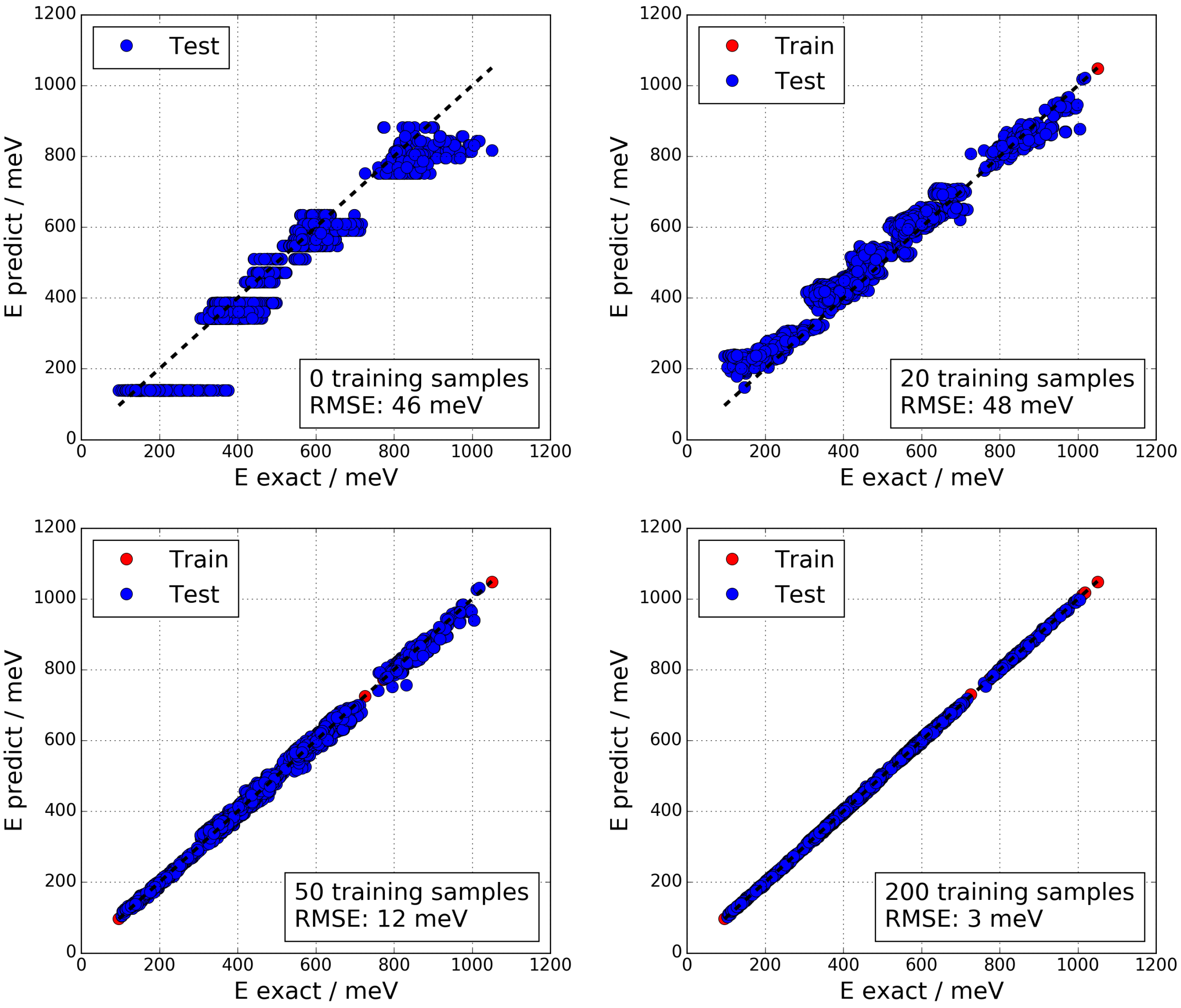


Figure 21: *Predicted energies per molecule compared to reference DFT energies for increasing number of training samples. The model already gives a near perfect fit after having been trained on only 200 samples (≈ 3 %) of the total 6288 configurations.*

It can be seen that (at least for the vacuum case) the model can reach accuracies of a few meV per molecule. Higher order effects and (very) long range interactions that span distances larger than the cutoff distance are shown to lead to deviations of only a few meV. The distribution of residuals in Fig. 24 shows a notable bimodal structure: For the majority of the configurations the model fits well and gives mean energy errors of about $\pm 2\ meV$. However for a minority of configurations the model does systematically underestimate the periodic energy by about $7\ meV$. This might be attributed to long range interactions beyond the cutoff radius or higher order effects.

This comparison shows the limit of prediction accuracy that can be achieved as the number of training sample increases. No matter how much training data we present to the machine learning algorithm, it will be fundamentally limited by the underlying model and no accuracy better than the one depicted in Fig. 24 can be reached. The only exception to this are the fit coefficients of the molecule-substrate interactions that are also left open as fit parameters in the full model, but were held fixed for this validation.

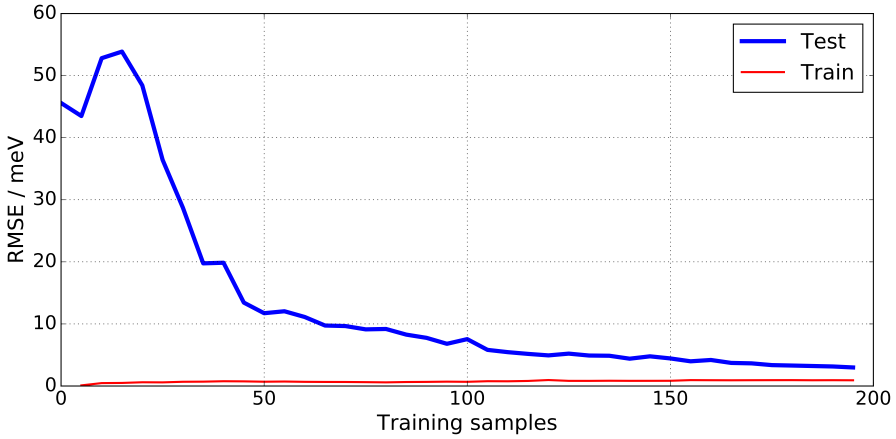


Figure 22: *Root Mean Square Error (RMSE) of predicted energies over number of training samples used. The mean error on the testing data decreases rapidly as a function of training sample size.*

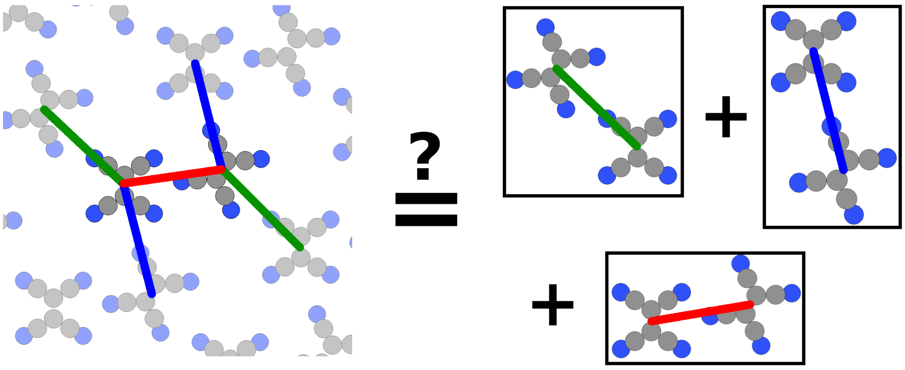


Figure 23: *Schematic comparison of periodic structure with the sum of pairwise interaction.*

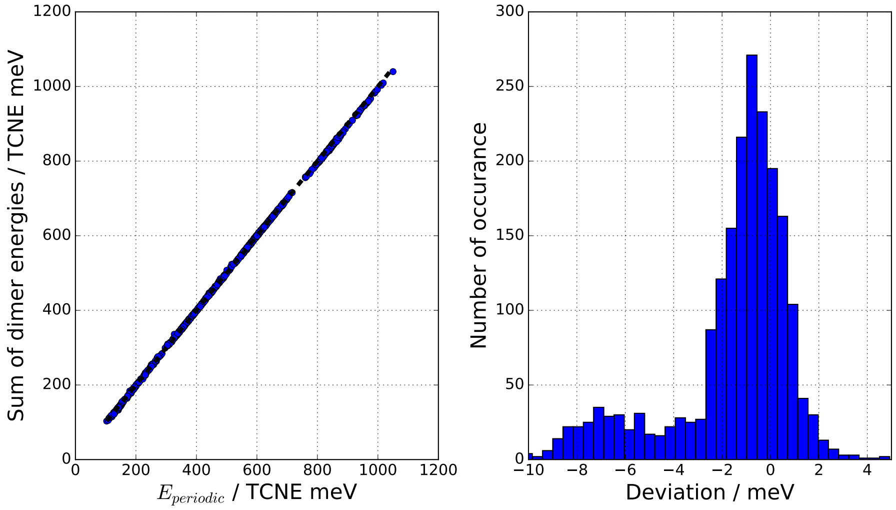


Figure 24: *Left: comparison of the energies of periodic configurations (in vacuum) and the sum of interaction energies obtained from dimer calculations. Right: Histogram of the energy deviation between the two energies in left figure. The histogram shows a doubly peaked structure: Most configurations are very well described by the model while a few configurations show a small bias of about 7 meV.*

4.8 Predictions for TCNE on Cu(111) surface

We now apply the method to the real physical system: TCNE adsorbed on the copper (111) surface. For these calculations we use the configurations listed in Tab. 1 as starting points for a local geometry optimization. Because the configurations proposed by the TETRIS algorithm lie close to the local energetic minima, these local optimizations usually converge within less than 10 geometry steps. When training the model we learn the relationship between the unrelaxed configuration and the relaxed adsorption energy. The model therefore does not only have to learn the interactions for the given geometry but also how these interactions will change upon geometry optimization. This will of course only work if the geometric changes occurring during the local geometry optimization are small, as they are in our case.

Figure 25 shows the fit quality obtained after training on 100 configurations and how it evolves as a function of training sample size. The prediction quality initially improves rapidly, mostly due to more accurate determination of local adsorption energies. Then the prediction quality improves more slowly towards a prediction uncertainty of about 10 meV, being significantly below thermal energies at room temperature.

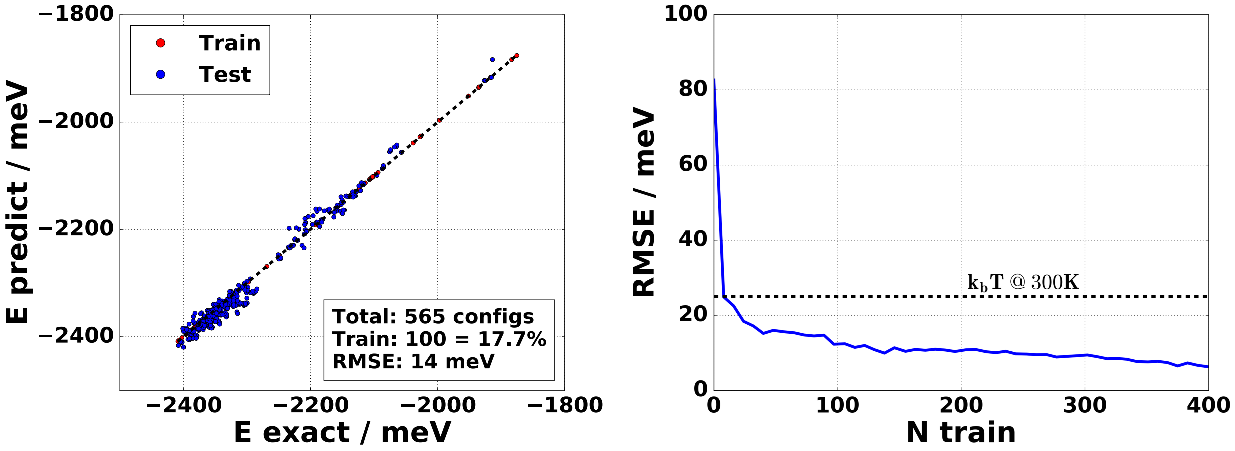


Figure 25: *Left: Predicted adsorption energies for configurations of TCNE/Cu(111) compared to their energies given by DFT. Right: Root Mean Square Error of predictions as a function of number of training samples.*

5 Results for TCNE on Ag(100)

As a second system to study we chose tetracyanoethylene on Ag(100). TCNE on the silver (100) surface presents an interesting system to test the structure search code on because it forms crystalline structures with multiple molecules per unit cell in it. This can be seen from the STM image in Fig. 26.

Experimentally, diagonal stripes of molecules are reported on clean parts of the surface that easily form chevrons of varying size: In regions with many defects the chevrons contain only two molecules per unit cell but in other regions chevrons with up to eight molecules per unit cell were found [40]. This provides an interesting system for our structure search method. Particular questions we would like to answer are: Can we find from first principles the experimentally observed structures? How can we treat large unit cells with up to eight molecules per unit cell where even very few DFT calculations might be computationally prohibitive?

5.1 Local Adsorption Geometries on Ag(100)

According to our TETRIS approach we start by finding the local adsorption geometries. The local adsorption geometries for TCNE on Ag(100) were determined with the same procedure as for the Cu(111) surface by optimizing the adsorption geometry of a single molecule from multiple starting points. As opposed to TCNE/Cu(111), for this system only flat lying molecules were considered. The found adsorption geometries and their corresponding adsorption energies are depicted in Fig. 27 and Fig. 28 respectively.

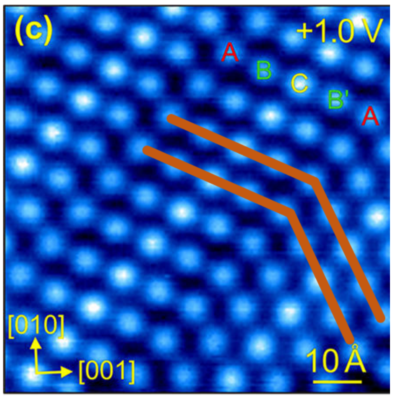


Figure 26: *STM image of TCNE on Ag(100) surface, adapted from [40]. The orange lines highlight the chevron structure of the adsorbed molecules.*

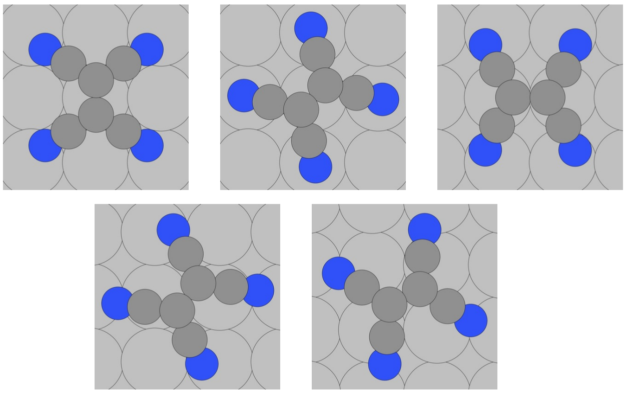


Figure 27: *Local adsorption geometries for TCNE/Ag(100).*

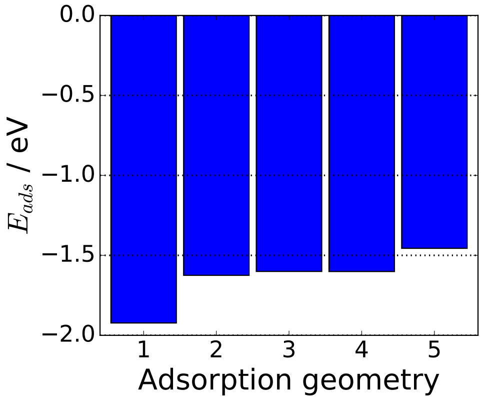


Figure 28: *Adsorption energies for TCNE on Ag(100). Lower means stronger adsorption.*

The three geometries depicted in the top row have one symmetry equivalent geometry each (90° rotation), the geometries in the bottom row have 3 symmetry equivalent geometries each. Geometry 1, with the molecule sitting on a top site, corresponding to the top left image in Fig. 27, is bound significantly stronger to the substrate than all other local adsorption geometries.

As for the system of TCNE/Cu(111), also here we restrict our search space by two assumptions: We only include configurations that have the experimental coverage of 2 molecules per 15 surface substrate atoms and we only allow rectangular supercells. This still leaves several different supercell shapes as listed in Tab. 2.

Table 2: *Considered supercells for TCNE on Ag(100). The supercell shape is given in multiples of the substrate unit cell.*

Molecules per supercell	Supercell shapes	Number of configurations
2	3×5	8
4	3×10, 6×5	243
6	3×15, 5×9	3997
8	3×20, 4×15, 5×12, 6×10	199324

5.2 Validation in Vacuum

As for the system of TCNE on Cu(111) in Sec. 4.3 we also validate our machine learning approach by calculating energies for monolayers of TCNE molecules without any substrate underneath. This hypothetical system can be cheaply calculated and serves to test the prediction accuracies that can be obtained. We have already validated that the pairwise interaction energies, obtained by the model, accurately represent real interaction energies (Sec. 4.7). We now want to see whether it is possible to accurately predict the energies of large unit cells, with many molecules in them, from training data that only contains few molecules.

All considered unit cells with 2, 4 or 6 molecules in them generate a combined number of 4248 configurations when enforcing a minimum distance between the molecules of 2.6 Å. The single point energies of all these configurations were calculated using DFT for validation purposes.

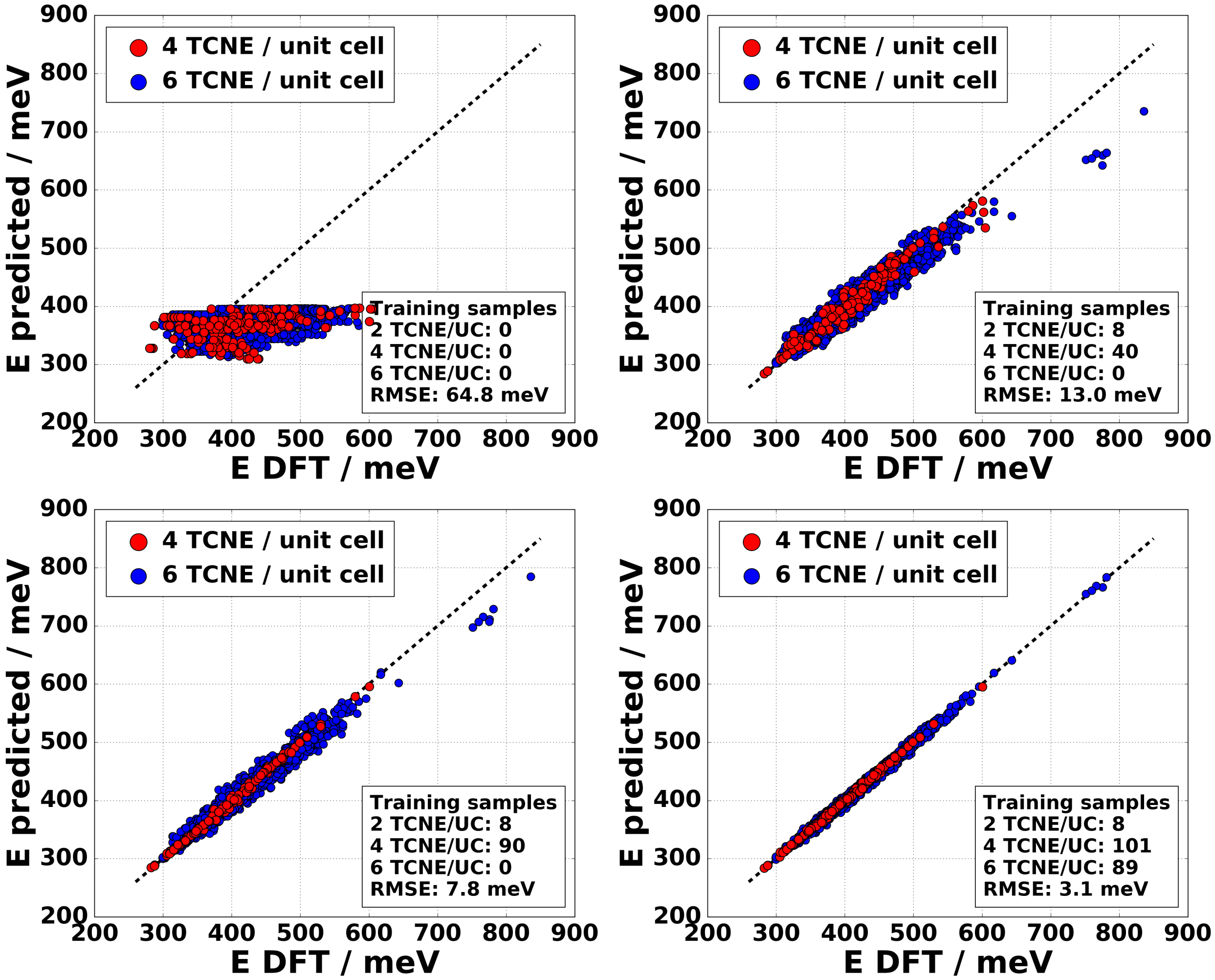


Figure 29: *Predicted energies per molecule compared to DFT energies for increasing number of training samples for a molecular monolayer without substrate on a 100 grid. Even when training only on small unit cells reasonable prediction accuracies can be obtained for configurations in large unit cells (blue).*

Then training batches consisting of 10 configurations each were selected using Fedorov's algorithm (Sec. 2.8). For these training samples only the configurations consisting of 2 or 4 molecules per unit cell were initially allowed, no configurations with 6 or more molecules per unit cell were proposed for training.

The machine learning model was trained on an ever increasing number of training batches and its performance was evaluated by comparing its prediction to the DFT energies calculated for the remaining configurations. Just as for the case of TCNE/Cu(111) a rapid improvement of the prediction quality as a function of training dataset size is observed (Fig. 29). Note that in Fig. 29 the prediction quality for configurations with 4 molecules per unit cell becomes almost exact after training on about 100 configurations. Even more remarkable, however, is that also the prediction for configurations with 6 molecules per unit cell improves drastically and yields reasonable results after only training on configurations with 4 or fewer molecules per unit cell. The root mean square error (RMSE) for all predictions (which is dominated by the much larger sample size of configurations with 6 TCNE/UC) is 8 meV and no extreme

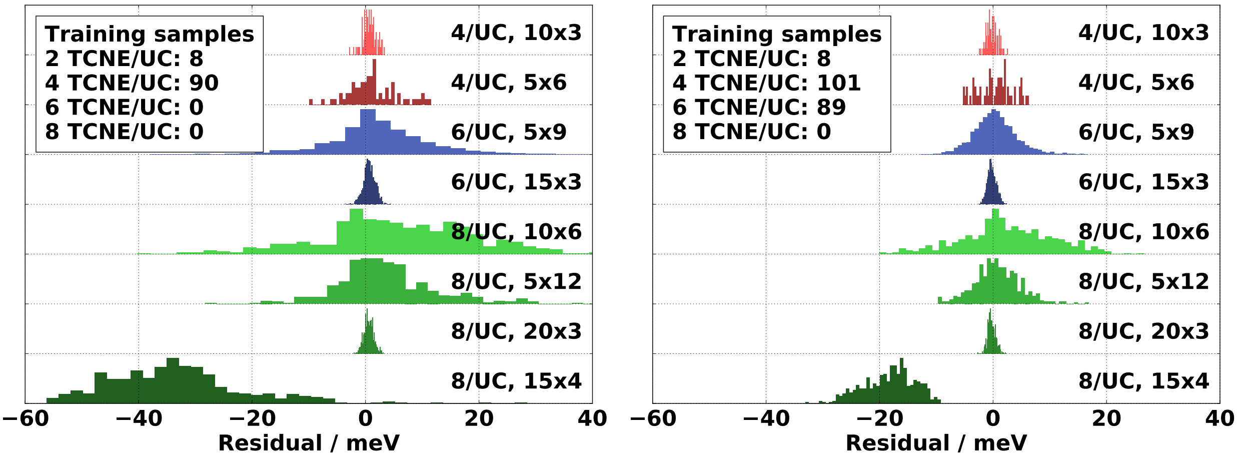


Figure 30: *Prediction error distribution for different unit cell shapes. In the left plot the model was only trained on unit cells with 2-4 molecules in it. For the right plot unit cells with up to 6 molecules in it where used for training.*

outliers are observed. Additionally including training samples with 6 molecules per unit cell further improves the prediction quality to a RMSE of 3 meV. Training the model only on small unit cells and using the model's predictions for larger unit cells is thus a viable strategy although significantly lower accuracy for unit cells that were not part of the training set has to be accepted. Training the model further including also unit cells with 6 molecules allows to improve this prediction accuracy at the expense of more expensive training data.

To gain more insight, it is instructive to further differentiate the configurations according to their unit cell shape. Figure 30 shows the distribution of residuals - the differences between predictions and DFT reference energies. For this plot not only all configurations with 2-6 molecules per unit cell were considered but also a random selection of 500 configurations for each unit cell with 8 molecules within it. Figure 30 shows that the prediction accuracy differs widely between different unit cell shapes. For all long and narrow unit cells (10x3, 15x3, 20x3) excellent prediction accuracies of about 5 meV are obtained, independent of the number of molecules within this cell. Other unit cell shapes yield poorer prediction accuracies but are unbiased: their average prediction is centered around the correct energy. The notable exception is the unit cell of shape 15x4 with 8 TCNE/UC which shows a bias of -20 to -40 meV, depending on the number of training samples used. One possible explanation is the fact that the 15x4 unit cell is the only unit cell shape that is not a multiple of the primitive 5x3 unit cell: All other unit cells in the considered test set can be described by stacking of this minimal 5x3 cell, while this is not possible for the 15x4 unit cell.

5.3 Prediction for TCNE on Ag(100)

For TCNE on Ag(100) a systematic training on increasingly large unit cells as described in Sec. 2.9 was done. As opposed to the case of TCNE/Cu(111) the proposed geometries were not relaxed but only single-point calculations of their energies were performed. This simplification was done because geometry optimization proved to be difficult for this system compared to TCNE/Cu(111): Using the BFGS trust radius method implemented in FHI-aims, many configurations required significantly more than 10 steps to even converge to relatively crude remaining forces of 0.1 eV/Å. In most cases geometry optimization did not significantly perturb the struc-

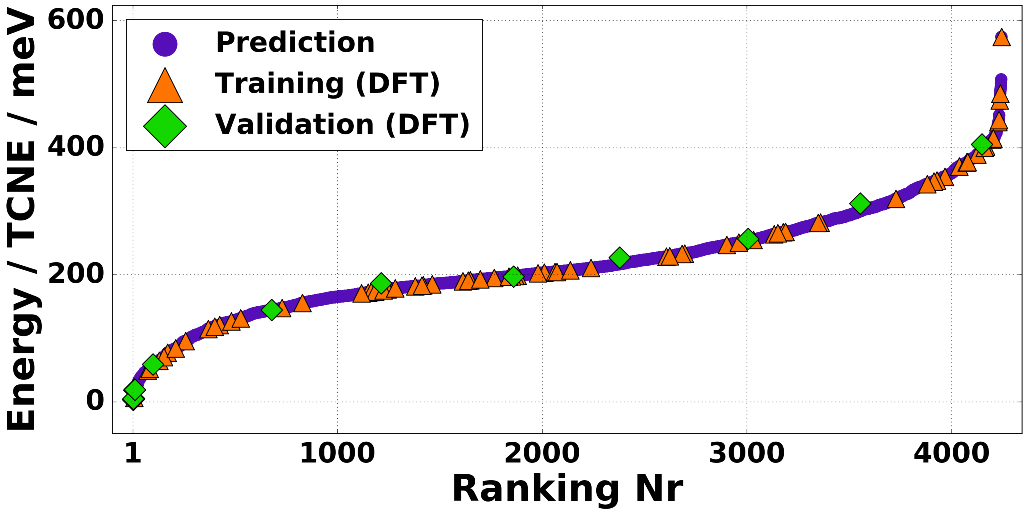


Figure 31: *Energy ranking for all configurations with 2-6 molecules per unit cell for TCNE/Ag(100). Green validation structures lie close to the predicted energies in violet across the entire energy range.*

tures obtained by the TETRIS assembly but in some cases geometry optimization transformed a starting geometry to a different configuration obtained by TETRIS. Since geometry optimization did in the cases tested only introduce energy deviations of about 50 meV between the configurations it was not done for the system of TCNE/Ag(100) to reduce the computational costs.

At first the model was trained (exhaustively) on all 8 configurations with only 2 molecules per unit cell. Then four training batches consisting of 10 configurations each were selected according to Fedorov's algorithm from all configurations with 4 molecules per unit cell. Two additional batches were chosen from all configurations that had at this point a predicted energy that was lower than 200 meV per TCNE molecule above the predicted energy minimum. A final training batch of four configurations with 6 molecules per unit cell was calculated.

At this point an energy prediction for all configurations with 2-6 molecules per unit cell was evaluated. To validate the prediction quality, the energy of 16 configurations was calculated: 8 configurations evenly distributed from low energy to high energy configurations and the 8 configurations that were predicted to be lowest in energy. The RMSE of these 16 configurations is 5.5 meV. The result of this prediction is shown in Fig. 31: All configurations are ranked according to their energy per molecule, ranging from low energy structures on the left via defect structures to high energy structures on the right.

Of most interest however is the low energy region which should contain the structures dominant at thermal equilibrium. Fig. 32 shows a zoom into the low energy region of Fig. 31. As expected the energy prediction is even better in the low energy region, compared to the prediction across the entire energy rage, yielding a RMSE of 2.1 meV. It is particularly notable that the algorithm correctly predicts six configurations to be lower in energy than the lowest configuration present in the training data-set.

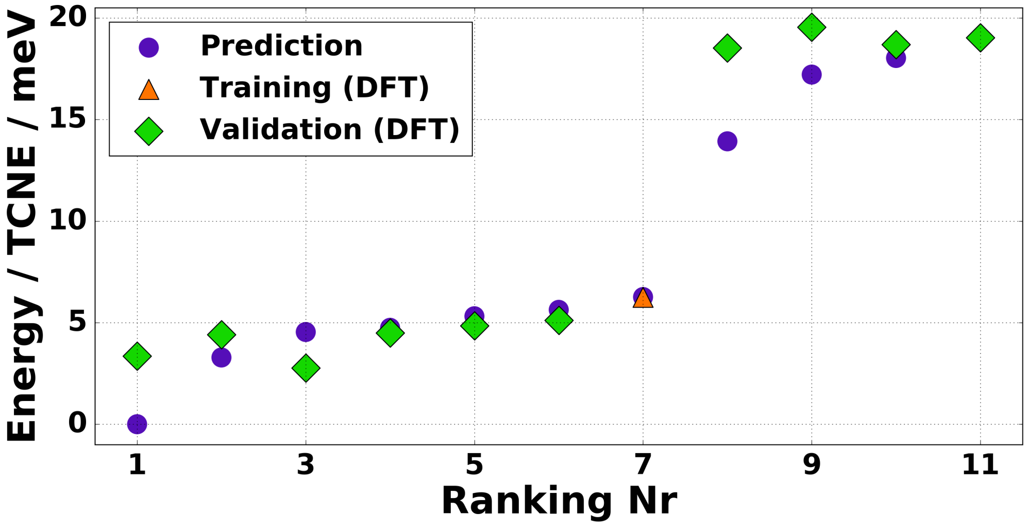


Figure 32: *Energy ranking of the lowest structures with 2-6 molecules per unit cell according to prediction and DFT.*

5.4 Predicted Low-Energy Structures on Ag(100)

Some of the low energy structures predicted by our algorithm are depicted in Fig. 33. All of these structures consist of alternating columns of top and bridge molecules. They are offset relative to each other to form diagonal lines. Introducing kinks in the diagonal lines to form chevron or zig-zag patterns is a low energy defect costing only a few meV. In all predicted low energy structures the backbones of the molecules are parallel to each other. Rotating a molecule from its parallel to an orthogonal orientation relative to its neighbors only appears as a defect of the ground state structure. Introducing this defect by rotating one molecule by 90° has an energy cost of about 100 meV. These predictions do qualitatively agree with the structure determined by Scanning Tunneling Microscopy [40]: Also the experimental study finds diagonal lines of molecules that can easily be perturbed to form zig-zag structures. They observe chevrons of varying size with the largest one consisting of eight molecules per unit cell (Fig. 26), while the smallest ones only consist of two molecules forming a simple zig-zag pattern. It is found that large chevrons form in pristine regions, while defects within the monolayer (e.g. missing molecules) lead to smaller structures. These findings are consistent with our prediction that introducing kinks costs very little energy and can thus easily be observed.

On the other hand, the experiment does also show some features that clearly differ from our prediction: While we predict the TCNE-backbones to be parallel to each other, the experiment observes alternating orientations of the TCNE molecules. While we predict the structures to consist of columns of molecules that alternate between bridge and top site, the experimental structure shows this alternating behavior only for the smallest structure consisting of two molecules per unit cell. Larger chevrons (with 4 or 8 molecules per unit cell) are formed by only placing one in four molecules on a top side and the remaining 3/4 on a bridge site.

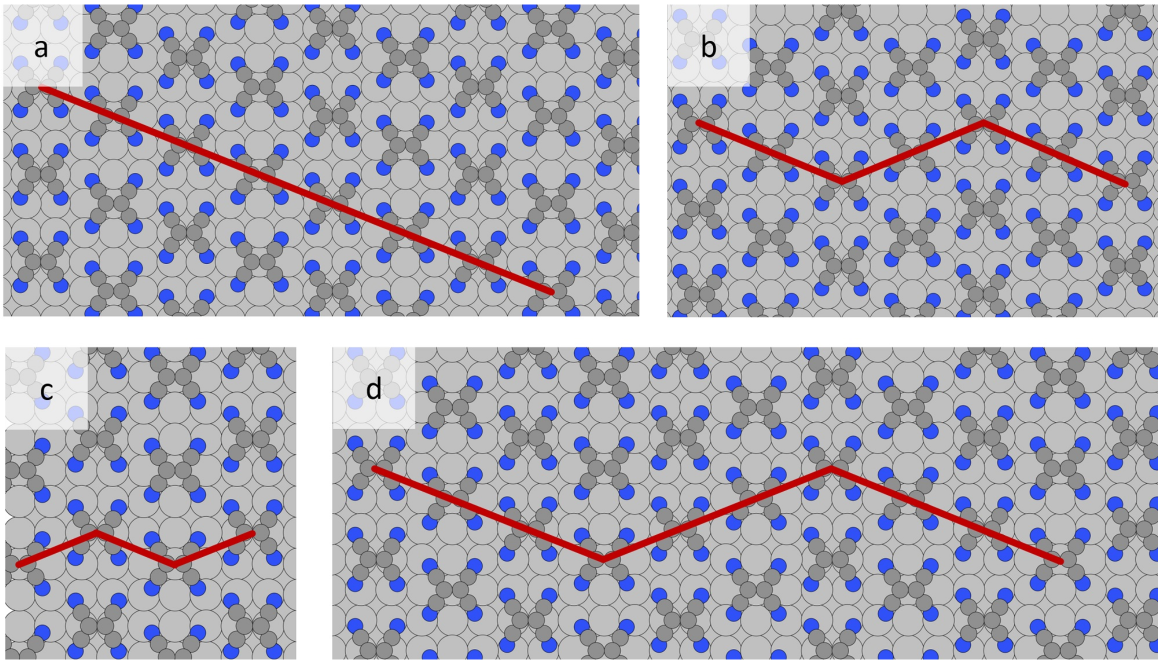


Figure 33: *Four examples of low energy structures.* ***a****: Diagonal stripes,* ***b-d****: Chevrons consisting of 2-6 molecules per unit cell. All these configurations are similar in energy according to DFT within 5 meV / molecule.*

At this point it has to be stressed that these deviations do not constitute a shortcoming of the machine-learning procedure, but rather a lacking accuracy of the energies calculated by DFT. There are several possible reasons for this discrepancy between Density Functional Theory and the experiment:

- **Accuracy of DFT functional**: Although Density Functional Theory is in principle exact, the functionals used for any practical calculation (in our case PBE + vdW corrections) are not. As opposed to wavefunction based methods there is unfortunately no systematic way to improve upon the results obtained by DFT to check the quality of the approximations made. Using a more sophisticated functional (e.g. hybrid-functionals) might shed more light on this issue.

- **Local Relaxation**: While the structures proposed by the TETRIS algorithm have shown to be good guess configurations, there will always be some additional relaxation of the structures once they are assembled on the surface. It might be necessary to feed the energies of locally optimized structures into the machine learning algorithm as opposed to single-point energies. However, we have seen this not to significantly alter the energetic ordering, in particular it does not affect the energy difference between orthogonal and parallel TCNE molecules.

- **Vibrational Enthalpy**: Even at $T = 0$, the total energy is not the only contribution that must be considered to find the ground-state structure. The zero point energy of the vibrational modes might be necessary to accurately rank the molecules. This is supported by the observation of Wegner et al [40] that the TCNE molecules show a shift of their vibrational frequencies depending on their adsorption site and surrounding.

- **Temperature**: Although the STM images were recorded at low temperatures ($T = 7\,K$) the deposition of the monolayer was done at room temperature. It is thus conceivable that a phase, other than the $T = 0$ ground-state, formed at room temperature and was "frozen in" when cooling to 7 K.

- **Kinetics**: The assumption that a phase of minimal free energy forms relies on the system to be in thermal equilibrium. Depending on the growth conditions this assumption might not hold for many experiments. It might therefore be of interest to study the dynamics of crystal growth on the surface. The fact that our method does not only yield the energy of the lowest energy structure but gives in fact an energy prediction for all local energy minima with an accuracy that comes close to the one obtained by DFT, should provide a good starting point for kinetic Monte Carlo simulations.

6 Summary and Outlook

In this work an efficient machine learning model to learn the adsorption energies of TCNE on metal surfaces has been developed and extensively tested on Cu(111) and Ag(100). At the current stage the model shows high prediction accuracy for a large dataset of configurations given a few hundred training data samples. Taking this work as a starting point, several questions will have to be addressed in the future:

Varying the supercell and coverage

At the current stage the code still needs experimental input in the form of the lattice parameters and coverage since we currently restrict the search space to configurations that show the experimental coverage. Introducing arbitrary supercell shapes should in principle be straight-forward since the code can already deal with configurations with different supercell sizes. However two important questions will have to be answered: How transferable are the learned interaction energies to configurations with a different coverage and how well are they transferable to supercell shapes that have never been trained on?

Better default model

A second large question is whether the prior assumptions used in the model could be improved. The model currently assumes zero interaction strength between the molecules if no DFT data is available but a computationally cheap physical model, such as electrostatics combined with empirical van der Waals interactions, might be able to yield a far better default guess. In systems that are dominated by interactions that can be described by simple, parametrized models this could help to further reduce the number of training samples needed.

Marginalization of hyper-parameters

Several parameters of the model are currently picked by hand and are thus in principle prone to human error and arbitrariness. For future work it is desirable to minimize the number of user picked hyper-parameters by determining their values self-consistently within the Bayesian framework. This should bring the additional benefit of more reliable posterior-covariances and thus access to error-bars for the predictions which were not discussed in this thesis.

Training data selection

Currently the training samples are initially only selected from small supercells before going to larger supercells. This is to steer the algorithm towards extracting information initially from the smaller, and thus cheaper, supercells before doing expensive calculations on large supercells. It might be advantageous to directly encode this difference in cost into the data selection process. Instead of asking Fedorov's algorithm to gain maximum information from exactly n_{train} DFT calculations, one could alter the algorithm to gain maximum information from a given number of CPU hours.

Additionally it could be desirable to also change the objective function that is optimized: Instead of minimizing the uncertainty of the interaction energies it might be better to directly minimize the prediction uncertainty obtained for the configurations of interest.

Bridging the gap to experiment

The case of TCNE on Ag(100) shows that the predicted structures do not always match experimental observations. It is currently not fully clear where this discrepancy is coming from: It might be a shortcoming of the first-principles method used to calculated the energies or a contribution currently not considered in the energy ranking such as vibrational and configurational entropy. It could however also be an experimental problem such as kinetically trapped structures or artifacts during STM imaging. For the future it will be necessary to gauge the influence of the various approximations made, in order to obtain reliable prediction uncertainties.

Simulating larger domains

Since the method outlined in this thesis cheaply yields energy predictions for any given configuration it is ideally suited to perform Monte Carlo simulations of larger domains. In particular if transition barriers between the predicted polymorphs were available one could use Kinetic Monte Carlo simulations to estimate the stability and lifetime of polymorphs and study their evolution and growth. This might eventually even allow to study phases which are not in thermodynamic equilibrium.

List of Figures

List of Tables

References

[1] 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. DOI: 10.1002/adfm.201503169. URL: http://onlinelibrary.wiley.com/doi/10.1002/adfm.201503169/abstract (visited on 12/09/2016).

[2] Norbert Koch. “Organic Electronic Devices and Their Functional Interfaces”. In: *ChemPhysChem* 8.10 (2007), pp. 1438–1455. ISSN: 1439-7641. DOI: 10.1002/cphc.200700177. URL: http://onlinelibrary.wiley.com/doi/10.1002/cphc.200700177/abstract (visited on 02/28/2017).

[3] Makoto Yoneya, Masahiro Kawasaki, and Masahiko Ando. “Are Pentacene Monolayer and Thin-Film Polymorphs Really Substrate-Induced? A Molecular Dynamics Simulation Study”. In: *The Journal of Physical Chemistry C* 116.1 (2012), pp. 791–795. ISSN: 1932-7447. DOI: 10.1021/jp208468b. URL: http://dx.doi.org/10.1021/jp208468b (visited on 02/23/2017).

[4] Stefano Curtarolo et al. “The High-Throughput Highway to Computational Materials Design”. In: *Nature Materials* 12.3 (2013), pp. 191–201. ISSN: 1476-1122. DOI: 10.1038/nmat3568. URL: http://www.nature.com/nmat/journal/v12/n3/abs/nmat3568.html (visited on 02/21/2017).

[5] A. M. Reilly et al. “Report on the Sixth Blind Test of Organic Crystal Structure Prediction Methods”. In: *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials* 72.4 (2016), pp. 439–459. ISSN: 2052-5206. DOI: 10.1107/S2052520616007447. URL: http://scripts.iucr.org/cgi-bin/paper?gp5080 (visited on 12/09/2016).

[6] David J. Wales and Jonathan P. K. Doye. “Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms”. In: *The Journal of Physical Chemistry A* 101.28 (1997), pp. 5111–5116. ISSN: 1089-5639. DOI: 10.1021/jp970984n. URL: http://dx.doi.org/10.1021/jp970984n (visited on 03/01/2017).

[7] Ralf Gehrke and Karsten Reuter. “Assessing the Efficiency of First-Principles Basin-Hopping Sampling”. In: *Physical Review B* 79.8 (2009), p. 085412. DOI: 10.1103/PhysRevB.79.085412. URL: http://link.aps.org/doi/10.1103/PhysRevB.79.085412 (visited on 02/20/2017).

[8] R. Unger and J. Moult. “Genetic Algorithms for Protein Folding Simulations”. In: *Journal of Molecular Biology* 231.1 (1993), pp. 75–81. ISSN: 0022-2836. DOI: 10.1006/jmbi.1993.1258.

[9] Peter Siepmann et al. “A Genetic Algorithm Approach to Probing the Evolution of Self-Organized Nanostructured Systems”. In: *Nano Letters* 7.7 (2007), pp. 1985–1990. ISSN: 1530-6984. DOI: 10.1021/nl070773m. URL: http://dx.doi.org/10.1021/nl070773m (visited on 02/14/2017).

[10] J. D. Lohn et al. “Evolutionary Design of an X-Band Antenna for NASA’s Space Technology 5 Mission”. In: *IEEE Antennas and Propagation Society Symposium, 2004.* IEEE Antennas and Propagation Society Symposium, 2004. Vol. 3. 2004, 2313–2316 Vol.3. DOI: 10.1109/APS.2004.1331834.

[11] Frank Jensen. *Introduction to Computational Chemistry: Second Edition.* 2nd ed. Chichester, England ; Hoboken, NJ: JW, 2011. 620 pp. ISBN: 978-0-470-01187-4.

[12] Wendy D. Cornell et al. “A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules”. In: *Journal of the American Chemical Society* 117.19 (1995), pp. 5179–5197. ISSN: 0002-7863. DOI: 10.1021/ja00124a002. URL: http://dx.doi.org/10.1021/ja00124a002 (visited on 02/01/2017).

[13] Michael J. S. Dewar et al. “Development and Use of Quantum Mechanical Molecular Models. 76. AM1: A New General Purpose Quantum Mechanical Molecular Model”. In: *Journal of the American Chemical Society* 107.13 (1985), pp. 3902–3909. ISSN: 0002-7863. DOI: 10.1021/ja00299a024. URL: http://dx.doi.org/10.1021/ja00299a024 (visited on 02/15/2017).

[14] “Jacob’s Ladder of Density Functional Approximations for the Exchange-Correlation Energy”. In: *AIP Conference Proceedings* 577.1 (2001), pp. 1–20. ISSN: 0094-243X. DOI: 10.1063/1.1390175. URL: http://aip.scitation.org/doi/abs/10.1063/1.1390175 (visited on 01/29/2017).

[15] S. H. Vosko, L. Wilk, and M. Nusair. “Accurate Spin-Dependent Electron Liquid Correlation Energies for Local Spin Density Calculations: A Critical Analysis”. In: *Canadian Journal of Physics* 58.8 (1980), pp. 1200–1211. ISSN: 0008-4204. DOI: 10.1139/p80-159. URL: http://www.nrcresearchpress.com/doi/abs/10.1139/p80-159 (visited on 02/27/2017).

[16] John P. Perdew and Yue Wang. “Accurate and Simple Analytic Representation of the Electron-Gas Correlation Energy”. In: *Physical Review B* 45.23 (1992), pp. 13244–13249. DOI: 10.1103/PhysRevB.45.13244. URL: http://link.aps.org/doi/10.1103/PhysRevB.45.13244 (visited on 02/27/2017).

[17] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. “Generalized Gradient Approximation Made Simple”. In: *Physical Review Letters* 77.18 (1996), pp. 3865–3868. DOI: 10.1103/PhysRevLett.77.3865. URL: http://link.aps.org/doi/10.1103/PhysRevLett.77.3865 (visited on 02/27/2017).

[18] Jianwei Sun, Adrienn Ruzsinszky, and John P. Perdew. “Strongly Constrained and Appropriately Normed Semilocal Density Functional”. In: *Physical Review Letters* 115.3 (2015), p. 036402. DOI: 10.1103/PhysRevLett.115.036402. URL: http://link.aps.org/doi/10.1103/PhysRevLett.115.036402 (visited on 02/27/2017).

[19] “Density-Functional Thermochemistry. III. The Role of Exact Exchange”. In: *The Journal of Chemical Physics* 98.7 (1993), pp. 5648–5652. ISSN: 0021-9606. DOI: 10.1063/1.464913. URL: http://aip.scitation.org/doi/abs/10.1063/1.464913 (visited on 02/27/2017).

[20] “Rationale for Mixing Exact Exchange with Density Functional Approximations”. In: *The Journal of Chemical Physics* 105.22 (1996), pp. 9982–9985. ISSN: 0021-9606. DOI: 10.1063/1.472933. URL: http://aip.scitation.org/doi/abs/10.1063/1.472933 (visited on 02/09/2017).

[21] Stefan Grimme. “Accurate Description of van Der Waals Complexes by Density Functional Theory Including Empirical Corrections”. In: *Journal of Computational Chemistry* 25.12 (2004), pp. 1463–1473. ISSN: 1096-987X. DOI: 10.1002/jcc.20078. URL: http://onlinelibrary.wiley.com/doi/10.1002/jcc.20078/abstract (visited on 02/27/2017).

[22] Qin Wu and Weitao Yang. “Empirical Correction to Density Functional Theory for van Der Waals Interactions”. In: *The Journal of Chemical Physics* 116.2 (2002), pp. 515–524. ISSN: 0021-9606. DOI: 10.1063/1.1424928. URL: http://aip.scitation.org/doi/abs/10.1063/1.1424928 (visited on 02/27/2017).

[23] Alexandre Tkatchenko and Matthias Scheffler. “Accurate Molecular Van Der Waals Interactions from Ground-State Electron Density and Free-Atom Reference Data”. In: *Physical Review Letters* 102.7 (2009), p. 073005. DOI: 10.1103/PhysRevLett.102.073005. URL: http://link.aps.org/doi/10.1103/PhysRevLett.102.073005 (visited on 02/27/2017).

[24] Victor G. Ruiz et al. “Density-Functional Theory with Screened van Der Waals Interactions for the Modeling of Hybrid Inorganic-Organic Systems”. In: *Physical Review Letters* 108.14 (2012), p. 146103. DOI: 10.1103/PhysRevLett.108.146103. URL: http://link.aps.org/doi/10.1103/PhysRevLett.108.146103 (visited on 12/12/2016).

[25] Alexandre Tkatchenko et al. “Accurate and Efficient Method for Many-Body van Der Waals Interactions”. In: *Physical Review Letters* 108.23 (2012), p. 236402. DOI: 10.1103/PhysRevLett.108.236402. URL: http://link.aps.org/doi/10.1103/PhysRevLett.108.236402 (visited on 02/27/2017).

[26] Wei Liu et al. “Quantitative Prediction of Molecular Adsorption: Structure and Binding of Benzene on Coinage Metals”. In: *Physical Review Letters* 115.3 (2015), p. 036104. DOI: 10.1103/PhysRevLett.115.036104. URL: http://link.aps.org/doi/10.1103/PhysRevLett.115.036104 (visited on 03/01/2017).

[27] Luca M. Ghiringhelli et al. “Big Data of Materials Science: Critical Role of the Descriptor”. In: *Physical Review Letters* 114.10 (2015), p. 105503. DOI: 10.1103/PhysRevLett.114.105503. URL: http://link.aps.org/doi/10.1103/PhysRevLett.114.105503 (visited on 02/19/2017).

[28] Robert Tibshirani. “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288. ISSN: 0035-9246. JSTOR: 2346178.

[29] Dan Claudiu Ciresan et al. “Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition”. In: *Neural Computation* 22.12 (2010), pp. 3207–3220. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/NECO_a_00052. URL: http://arxiv.org/abs/1003.0358 (visited on 02/20/2017).

[30] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-015-0816-y. URL: http://link.springer.com/article/10.1007/s11263-015-0816-y (visited on 02/20/2017).

[31] Matthias Rupp. “Machine Learning for Quantum Mechanics in a Nutshell”. In: *International Journal of Quantum Chemistry* 115.16 (2015), pp. 1058–1073. ISSN: 1097-461X. DOI: 10.1002/qua.24954. URL: http://onlinelibrary.wiley.com/doi/10.1002/qua.24954/abstract (visited on 02/21/2017).

[32] Katja Hansen et al. “Assessment and Validation of Machine Learning Methods for Predicting Molecular Atomization Energies”. In: *Journal of Chemical Theory and Computation* 9.8 (2013), pp. 3404–3419. ISSN: 1549-9618. DOI: 10.1021/ct400195d. URL: http://dx.doi.org/10.1021/ct400195d (visited on 01/13/2017).

[33] Volker Blum et al. “Using Genetic Algorithms to Map First-Principles Results to Model Hamiltonians: Application to the Generalized Ising Model for Alloys”. In: *Physical Review B* 72.16 (2005), p. 165113. DOI: 10.1103/PhysRevB.72.165113. URL: http://link.aps.org/doi/10.1103/PhysRevB.72.165113 (visited on 02/21/2017).

[34] Lance J. Nelson et al. “Cluster Expansion Made Easy with Bayesian Compressive Sensing”. In: *Physical Review B* 88.15 (2013), p. 155105. DOI: 10.1103/PhysRevB.88.155105. URL: http://link.aps.org/doi/10.1103/PhysRevB.88.155105 (visited on 02/21/2017).

[35] Matthias Rupp et al. “Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning”. In: *Physical Review Letters* 108.5 (2012), p. 058301. DOI: 10.1103/PhysRevLett.108.058301. URL: http://link.aps.org/doi/10.1103/PhysRevLett.108.058301 (visited on 01/02/2017).

[36] William J. Welch. “Algorithmic Complexity: Three NP- Hard Problems in Computational Statistics”. In: *Journal of Statistical Computation and Simulation* 15.1 (1982), pp. 17–25. ISSN: 0094-9655. DOI: 10.1080/00949658208810560. URL: http://dx.doi.org/10.1080/00949658208810560 (visited on 03/02/2017).

[37] Alan J. Miller and Nam-Ky Nguyen. “Algorithm AS 295: A Fedorov Exchange Algorithm for D-Optimal Design”. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 43.4 (1994), pp. 669–677. ISSN: 0035-9254. DOI: 10.2307/2986264. JSTOR: 2986264.

[38] *Van Der Waals Radius*. In: *Wikipedia*. Page Version ID: 753350551. 2016. URL: https://en.wikipedia.org/w/index.php?title=Van_der_Waals_radius&oldid=753350551 (visited on 03/30/2017).

[39] 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. DOI: 10.1016/j.cpc.2009.06.022. URL: http://www.sciencedirect.com/science/article/pii/S0010465509002033 (visited on 03/13/2017).

[40] Daniel Wegner et al. “Adsorption Site Determination of a Molecular Monolayer via Inelastic Tunneling”. In: *Nano Letters* 13.6 (2013), pp. 2346–2350. ISSN: 1530-6984. DOI: 10.1021/nl304081q. URL: http://dx.doi.org/10.1021/nl304081q (visited on 12/14/2016).

A Appendix

A.1 Computational Settings for TCNE on Cu(111)

For the DFT calculations of TCNE on Cu(111) a periodic 4 layer slab was used with a lattice constant of 2.553 Å.For geometry optimization all layers except the top two Cu layers were held fixed. The following computational settings (control.in) were used for the FHI-aims code:

```
xc pbe
vdw_correction_hirshfeld .true.
vdw_pair_ignore Cu Cu
relativistic atomic_zora scalar
RI_method lvl_fast
charge 0
spin none
use_dipole_correction .true.
compensate_multipole_errors .true.

# Define convergence
sc_accuracy_rho 1e-2
sc_accuracy_etot 1e-5
sc_accuracy_forces 1e-2
sc_iter_limit 100

# Geometry Optimization
relax_geometry trm 1.5e-1
max_relaxation_steps 10
output_level MD_light
occupation_type gaussian 0.01            # occupation_type type width,
   broadening scheme to define Fermi level and occupation of KS-eigenstates
k_grid 3 2 1                             # k_point grid n1 n2 n3

collect_eigenvectors .false.
evaluate_work_function

# optimized for 3 flat TCNE on 4 layer Cu 6x6
preconditioner kerker 2.2
charge_mix_param 0.15
################################################################################
#  FHI-aims code project
#  Volker Blum, Fritz Haber Institute Berlin, 2010
#  Suggested "tight" defaults for Cu atom (to be pasted into control.in file)
################################################################################
  species        Cu
  hirshfeld_param 59 10.9 2.4 #vdW surf
#
    nucleus       29
    mass          63.546
#
    l_hartree     6
#
    cut_pot       4.0  2.0  1.0
    basis_dep_cutoff    1e-3
#
    radial_base         53 7.0
    radial_multiplier   1
    angular_grids       specified
      division   0.3478   50
      division   0.6638  110
      division   0.9718  194
      division   1.1992  302
      division   1.5920  434
#      division   1.8557  590
#      division   2.0466  770
#      division   2.0877  974
#      division   2.4589 1202
      outer_grid  434
################################################################################
#
#  Definition of "minimal" basis
#
################################################################################
#     valence basis states
    valence      4  s   1.
    valence      3  p   6.
    valence      3  d  10.
#     ion occupancy
    ion_occ      4  s   0.
    ion_occ      3  p   6.
    ion_occ      3  d   9.
################################################################################
#
#  Suggested additional basis functions. For production calculations,
#  uncomment them one after another (the most important basis functions are
#  listed first).
#
#  Constructed for dimers: 1.8, 2.2, 3.0, 4.0 Ang
#
################################################################################
#  "First tier" - improvements: -211.42 meV to -9.17 meV
     ionic 4 p auto
#     hydro 4 f 7.4
     hydro 3 s 2.6
#     hydro 3 d 5
#     hydro 5 g 10.4
################################################################################
#  FHI-aims code project
#  Volker Blum, Fritz Haber Institute Berlin, 2009
#  Suggested "tight" defaults for C atom (to be pasted into control.in file)
################################################################################
  species        C
#     global species definitions
    nucleus             6
    mass                12.0107
#
    l_hartree           6
#
    cut_pot             4.0  2.0  1.0
    basis_dep_cutoff    1e-4
#
    radial_base         34 7.0
    radial_multiplier   2
    angular_grids specified
      division   0.2187   50
      division   0.4416  110
      division   0.6335  194
      division   0.7727  302
      division   0.8772  434
#      division   0.9334  590
#      division   0.9924  770
#      division   1.0230  974
#      division   1.5020 1202
#     outer_grid  974
      outer_grid  434
################################################################################
#
#  Definition of "minimal" basis
#
################################################################################
#     valence basis states
    valence      2  s   2.
    valence      2  p   2.
#     ion occupancy
    ion_occ      2  s   1.
    ion_occ      2  p   1.
################################################################################
#  Suggested additional basis functions. For production calculations,
#  uncomment them one after another (the most important basis functions are
#  listed first).
#  Constructed for dimers: 1.0 A, 1.25 A, 1.5 A, 2.0 A, 3.0 A
################################################################################
#  "First tier" - improvements: -1214.57 meV to -155.61 meV
     hydro 2 p 1.7
     hydro 3 d 6
     hydro 2 s 4.9
#  "Second tier" - improvements: -67.75 meV to -5.23 meV
     hydro 4 f 9.8
     hydro 3 p 5.2
     hydro 3 s 4.3
     hydro 5 g 14.4
     hydro 3 d 6.2
################################################################################
#  FHI-aims code project
#  Volker Blum, Fritz Haber Institute Berlin, 2009
#  Suggested "tight" defaults for N atom (to be pasted into control.in file)
################################################################################
  species        N
#     global species definitions
    nucleus             7
    mass                14.0067
#
    l_hartree           6
#
    cut_pot             4.0  2.0  1.0
    basis_dep_cutoff    1e-4
#
    radial_base         35 7.0
    radial_multiplier   2
    angular_grids       specified
      division   0.1841   50
      division   0.3514  110
      division   0.5126  194
      division   0.6292  302
      division   0.6939  434
#      division   0.7396  590
#      division   0.7632  770
#      division   0.8122  974
#      division   1.1604 1202
#      outer_grid  974
      outer_grid  434
################################################################################
#
#  Definition of "minimal" basis
#
################################################################################
#     valence basis states
    valence      2   s   2.
    valence      2   p   3.
#     ion occupancy
    ion_occ      2   s   1.
    ion_occ      2   p   2.
################################################################################
#  Suggested additional basis functions. For production calculations,
#  uncomment them one after another (the most important basis functions are
#  listed first).
#  Constructed for dimers: 1.0 A, 1.1 A, 1.5 A, 2.0 A, 3.0 A
################################################################################
#  "First tier" - improvements: -1193.42 meV to -220.60 meV
     hydro 2 p 1.8
     hydro 3 d 6.8
     hydro 3 s 5.8
#  "Second tier" - improvements: -80.21 meV to -6.86 meV
     hydro 4 f 10.8
     hydro 3 p 5.8
     hydro 1 s 0.8
     hydro 5 g 16
     hydro 3 d 4.9
```

A.2 Convergence Tests for TCNE on Ag(100)

To converge the computational settings for TCNE on Ag(100) at first a baseline adsorption energy was calculated. This was done for a flat lying TCNE molecule in a 4x4 Ag supercell on a 5 layer Ag slab with default FHI-aims tight settings and 16x16 k-points. For the Ag substrate tier 1 basis functions were used. For the organic adsorbate a tier 2 basis set was used. For the Ag slab a primitive lattice constant of 3.957 Å was used which deviates by about 3 % from the experimental lattice constant of 4.08 Å. Tests on a few selected configurations showed no significant influence of the lattice constant on the adsorption energies.

Using this baseline, several parameters were varied and their effect on adsorption energy and runtime tabulated in Tab. 3. For the calculations presented in Sec. 5 a k-point density of 30 was used: The number of k-points times the length of the supercell measured in multiples of the Ag unit cell was 30. The full control file used for the FHI-aims DFT code is listed below:

Table 3: *Convergence tests for TCNE/Ag100: All changes relative to baseline calculation*

Parameter	ΔE_{ads} / meV	Δ SCF-time / %	Accepted
Radial multiplier Ag 2 → 1	0	-8	yes
wave threshold 1e-6 → 1e-5	0	-2	yes
cut pot Ag 4.0 → 4.3	2	3	no
4 layers Ag slab	-68	-37	no
6 layers Ag slab	67	51	yes
7 layers Ag slab	-67	121	no
remove Ag 5g basis function	13	-35	yes
remove Ag 4d basis function	11	-18	yes
remove Ag 5g+4d basis function	27	-50	yes
remove Ag 5p basis function	-1539	-13	no
remove Ag 4f basis function	68	-27	no
remove Ag 4f+5g basis function	69	-56	no

```
xc pbe
vdw_correction_hirshfeld .true.
relativistic atomic_zora scalar
RI_method lvl_fast
charge 0
spin none
use_dipole_correction .true.
evaluate_work_function
compensate_multipole_errors .true.
sc_accuracy_rho 1e-2
sc_accuracy_etot 1e-5
wave_threshold 1e-5
sc_iter_limit 250
max_relaxation_steps 20
output_level MD_light
occupation_type gaussian 0.01
k_grid 6 5 1
collect_eigenvectors .false.
################################################################################
#  FHI-aims code project
#  Volker Blum, Fritz Haber Institute Berlin, 2009
#  Suggested "tight" defaults for Ag atom (to be pasted into control.in file)
################################################################################
  species        Ag
  hirshfeld_param 122 15.4 2.57 #vdW surf
#     global species definitions
    nucleus             47
    mass                107.8682
#
    l_hartree           6
#
    cut_pot             4.0  2.0  1.0
    basis_dep_cutoff    1e-4
#
    radial_base         62 7.0
    radial_multiplier   1
    angular_grids specified
      division   0.3947   50
      division   0.7739  110
      division   1.1156  194
      division   1.3117  302
      division   1.5936  434
#      division   2.0830  590
#      division   2.2341  770
#      division   2.8497 1202
#      outer_grid  974
      outer_grid  434
################################################################################
#  Definition of "minimal" basis
################################################################################
#     valence basis states
    valence      5  s   1.
    valence      4  p   6.
    valence      4  d  10.
#     ion occupancy
    ion_occ     5  s   0.
    ion_occ     4  p   6.
    ion_occ     4  d   9.
################################################################################
#  Suggested additional basis functions. For production calculations,
#  uncomment them one after another (the most important basis functions are
#  listed first).
#
#  Constructed for dimers: 2.1 A, 2.45 A, 3.00 A, 4.00 A
################################################################################
#  "First tier" - max. impr. -144.99  meV, min. impr. -3.42 meV
     ionic 5 p auto
     hydro 4 f 7.6
     hydro 3 s 2.6
#     hydro 5 g 9.8
#     hydro 4 d 8.4
################################################################################
#  FHI-aims code project
#  Volker Blum, Fritz Haber Institute Berlin, 2009
#  Suggested "tight" defaults for C atom (to be pasted into control.in file)
################################################################################
```

```
  species        C
#     global species definitions
    nucleus             6
    mass                12.0107
#
    l_hartree           6
#
    cut_pot             4.0  2.0  1.0
    basis_dep_cutoff    1e-4
#
    radial_base         34 7.0
    radial_multiplier   2
    angular_grids specified
      division   0.2187   50
      division   0.4416  110
      division   0.6335  194
      division   0.7727  302
      division   0.8772  434
#      division   0.9334  590
#      division   0.9924  770
#      division   1.0230  974
#      division   1.5020 1202
#     outer_grid  974
      outer_grid  434
################################################################################
#  Definition of "minimal" basis
################################################################################
#     valence basis states
    valence      2  s   2.
    valence      2  p   2.
#     ion occupancy
    ion_occ      2  s   1.
    ion_occ      2  p   1.
################################################################################
#  Suggested additional basis functions. For production calculations,
#  uncomment them one after another (the most important basis functions are
#  listed first).
#  Constructed for dimers: 1.0 A, 1.25 A, 1.5 A, 2.0 A, 3.0 A
################################################################################
#  "First tier" - improvements: -1214.57 meV to -155.61 meV
     hydro 2 p 1.7
     hydro 3 d 6
     hydro 2 s 4.9
#  "Second tier" - improvements: -67.75 meV to -5.23 meV
     hydro 4 f 9.8
     hydro 3 p 5.2
     hydro 3 s 4.3
     hydro 5 g 14.4
     hydro 3 d 6.2
################################################################################
#  FHI-aims code project
#  Volker Blum, Fritz Haber Institute Berlin, 2009
#  Suggested "tight" defaults for N atom (to be pasted into control.in file)
################################################################################
  species        N
#     global species definitions
```

```
    nucleus             7
    mass                14.0067
#
    l_hartree           6
#
    cut_pot             4.0  2.0  1.0
    basis_dep_cutoff    1e-4
#
    radial_base         35 7.0
    radial_multiplier   2
    angular_grids       specified
      division   0.1841   50
      division   0.3514  110
      division   0.5126  194
      division   0.6292  302
      division   0.6939  434
#      division   0.7396  590
#      division   0.7632  770
#      division   0.8122  974
#      division   1.1604 1202
#      outer_grid  974
      outer_grid  434
################################################################################
#  Definition of "minimal" basis
################################################################################
#     valence basis states
    valence      2  s   2.
    valence      2  p   3.
#     ion occupancy
    ion_occ      2  s   1.
    ion_occ      2  p   2.
################################################################################
#  Suggested additional basis functions. For production calculations,
#  uncomment them one after another (the most important basis functions are
#  listed first).
#  Constructed for dimers: 1.0 A, 1.1 A, 1.5 A, 2.0 A, 3.0 A
################################################################################
#  "First tier" - improvements: -1193.42 meV to -220.60 meV
     hydro 2 p 1.8
     hydro 3 d 6.8
     hydro 3 s 5.8
#  "Second tier" - improvements: -80.21 meV to -6.86 meV
     hydro 4 f 10.8
     hydro 3 p 5.8
     hydro 1 s 0.8
     hydro 5 g 16
     hydro 3 d 4.9
```