Aalto University
School of Science
Master's Programme in Physics of Advanced Materials

Yashasvi Singh Ranawat

Descriptor based adsorption-energy prediction on nano-clusters for catalyst selection

Master's Thesis
Espoo, July 13, 2018

Supervisor: Professor Adam Foster, Aalto University
Advisor: Marc Jäger M.Sc.
Eiakihonroeda Morooka M.Sc.

Aalto University
School of Science
Master's Programme in Physics of Advanced Materials

ABSTRACT OF MASTER'S THESIS

Author:	Yashasvi Singh Ranawat		
Title: Descriptor based adsorption-energy prediction on nano-clusters for catalyst selection			
Date:	July 13, 2018	**Pages:**	vi + 51
Major:	Physics of Advanced Materials	**Code:**	SCI3057
Supervisor:	Professor Adam Foster		
Advisor:	Marc Jäger M.Sc. Eiakihonroeda Morooka M.Sc.		

Nano-catalyst design, supplanting critical/rare metals with earth-abundant elements, for hydrogen evolution reactions (HERs) is a significant material-science and economic challenge. These design challenges can be significantly overcome by extensive first principle simulations. However, these simulations are computationally costly. With an introduction of descriptor, machine-learning methods afford significant advantages in such scenario. Descriptors: Smooth Overlap of Atomic Positions (SOAP) based on charge density, BLEACH , and Local Many-Body Tensor Representation (LMBTR) are proposed, and developed. These are evaluated on database of AuCu, and MoS_2 nano-clusters. A learning error of 0.05 eV in adsorption energy, and 1.7 me in charge on hydrogen are realised for LMBTR; while 0.08 eV and 13.4 me, respectively for BLEACH on the AuCu dataset. Although not as accurate as state-of-the-art SOAP-lite (3.36 meV and 0.077 me, respectively), these descriptors have their own benefits. While LMBTR allows for bi-directional operability, BLEACH provides element-agnosticism; both of which missing are from SOAP.

Keywords: Material science, catalyst, descriptor, energy prediction, machine learning

Language: English

Acknowledgements

I express my sincere gratitude to Prof. Foster for the opportunity and guidance, for the duration of thesis. His support and freedom to pursue the topics for thesis were encouraging. I would like to thank Marc and Aki for their unrelenting help in the minutest issue I faced. And also Dr. Filippo Canova, and Dr. Ondrej Krejci for their detailed discussions and valuable inputs.

Finally I extend my thanks to Department of Applied Physics, Aalto, and projects: NOMAD and CritCat.

Espoo, July 13, 2018

Yashasvi Singh Ranawat

Abbreviations and Acronyms

AAD	Average Absolute Deviation
ACSF	Atom Centred Symmetry Functions
BLEACH	A class of descriptors based on GTO, and orbital integrals
BoB	Bag of Bonds
CM	Coulomb Matrix
CP2K	atomistic simulation software package based on quantum chemistry and solid state physics
CritCat	Towards Replacement of Critical Catalyst Materials by Improved Nanoparticle Control and Rational Design
DFT	Density Functional Theory
pDOS	Partial density of states
DZVP-SR-GTH	Double Zeta Valence Polerised - Small Range - GTH (pseudo-potential)
GTO	Gaussian Type Orbital
HER(s)	Hydrogen Evolution Reaction(s)
LMBTR	Local MBTR
MAE	Mean Absolute Error
MBTR	Many-Body Tensor Representation
MI	Mutual Information
NOMAD	Novel Materials Discovery (NOMAD) Laboratory
SCF	Self-Consistent Field
SOAP	Smooth Overlap of Atomic Positions
SOAP-lite	Surfaces and Interfaces at Nanoscale, Aalto group's implementation of SOAP
STO	Slater Type Orbital

Contents

1 Introduction **1**

1.1 Structure of thesis . . . 4

2 Background **5**

2.1 Electronic structure theory . . . 5

2.1.1 Born-Oppenheimer approximation . . . 5

2.1.2 Basis set . . . 6

2.1.3 Hartree-Fock matrices . . . 7

2.1.4 Self-consistent field . . . 9

2.1.5 Density functional theory . . . 9

2.2 Data pre-processing and machine-learning techniques . . . 10

2.2.1 Mutual Information . . . 10

2.2.2 Principal component analysis . . . 10

2.2.3 Kernel Ridge Regression . . . 11

2.2.4 Multi-Layer Perceptrons . . . 12

2.2.5 Recurrent neural network . . . 12

2.3 Descriptors . . . 13

2.3.1 SOAP . . . 13

2.3.1.1 SOAP-lite . . . 15

2.3.2 MBTR . . . 15

3 Methods **17**

3.1 Descriptor design . . . 17

3.1.1 SOAP based on charge density . . . 17

3.1.2 BLEACH . . . 19

3.1.3 LMBTR . . . 21

3.2 Error analysis . . . 22

4 Implementation and evaluation **23**

4.1 Environment . . . 23

4.1.1 Descriptor design . . . 23

4.1.2 Machine learning 24
4.1.3 Basis set 24
4.1.4 Database 25
4.1.5 DFT simulations 27
4.2 SOAP based on charge density 27
4.2.1 Numerical 28
4.2.2 Charge density using density matrix with all and select MO 29
4.3 BLEACH 30
4.3.1 Overlap matrix 30
4.3.2 Coulomb repulsion, with density matrix made from all and select MO 31
4.3.3 Fock matrix 31
4.3.4 Fock tensor 32
4.4 LMBTR 34

5 Discussion 39
5.1 SOAP based on charge density 39
5.2 BLEACH 39
5.3 LMBTR 40
5.4 Learning curves 41
5.5 Error analysis 42

6 Conclusions 44

7 Outlook 45

Chapter 1

Introduction

Catalyst design is central to virtually all chemical processes[1]. Its applications range from heterogeneous: haber process[2], electro-catalysis[3], cracking[4], etc to homogeneous[5], and biocatalysis[6]. When designed appropriately, a catalyst lowers the activation energies of a reaction. This improves reaction kinetics, but doesn't affect its enthalpy; and potentially makes the reaction favorable — and economical — for widespread commercial use.

However, the mechanism of catalytic action is complex; it requires extensive study of every aspect of a material, to gain useful insight for development of new catalysts, better suited for the reaction. European Union's Horizon 2020 research and innovation programme funds one such project: CritCat. It emphasises electro-catalysis, specifically, catalysts for hydrogen evolution reactions(HERs)[7]. Many experimental[8–10] and theoretical[11–16] works have studied this reaction for its vital application in fuel cell, water splitting, etc. However, the focus, in the project, is on "...the substitution of critical metals, especially rare platinum-group metals (PGMs), used in heterogeneous and electrochemical catalysis..."[17], as seen in fig 1.1. Therefore, AuCu and MoS_2 nano-clusters are considered in the thesis for catalytic action, since they comprise earth-abundant elements.

To gauge the suitability in catalysis, a test parameter is required that is qualitative, readily observable, and swiftly calculable. The Gibbs free energy of adsorption (ΔG_H) — adsorption energy, henceforth — of hydrogen on a catalyst is considered to be a good parameter[12,13]. It is the difference between the total energy of the adsorbed system and that of the clean adsorbate and the adsorbent. The H_2 molecule adsorbing: (a) should dissociate on the surface of catalyst, thus, ready for further reaction; (b) should desorb hydrogen atom from the surface, when it is required. Thus, an adsorption energy between -0.1–0 eV/atom ensures a good catalytic behavior, comparable to that of PGMs[18].

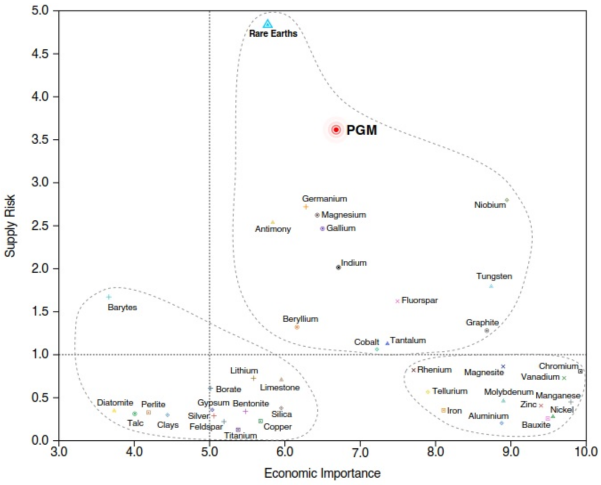
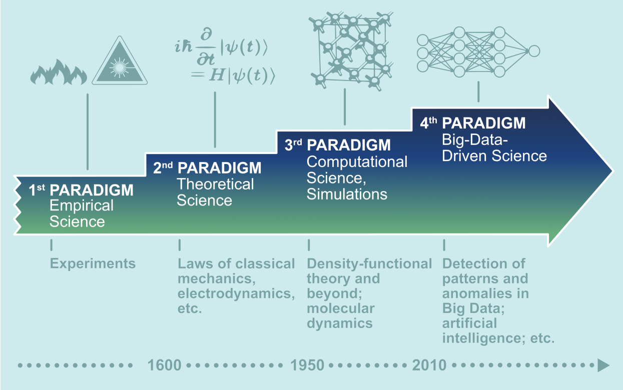


Figure 1.1: Economic importance v/s supply risk, emphasizing the need to replace critical material[17]

Adsorption energies are typically calculated by using ab-initio Density Functional Theory (DFT)[19]. It is widely adopted for electronic structure calculation, and is a vital tool in quantum chemistry[20]. Ground-state energies and densities of electrons in a potential field can be calculated by solving the Schrödinger equation — with approximations. The ground-state energy of nano-cluster, with and without adsorbed hydrogen, and that of a hydrogen molecule is used for adsorption energy calculation.

To find an appropriate nano-cluster catalyst thousands — ten thousand in our case — of calculations, with different nano-cluster composition and hydrogen positions, need to be performed. Such calculations can be done by high-throughput[18,21,22] workflows, that can sift through DFT simulations at an unprecedented scale. These results can also be queried from the existing database of HERs' catalyst; this simplifies storing and querying, and can remove redundancy in calculations. NOMAD[23] is one such repository of open-access data. It gathers computational materials science data, in the order of "several millions"[23]. Further, this abundance of data moves us into a new paradigm, as illustrated in 1.2. Here, big-data analysis[24] can identify correlations in these datasets, giving deeper insights to material design. Results from such calculations can hint at catalyst candidates befitting our requirements.

However, those myriad DFT calculations — including complementary NOMAD database entries — take immense computational and human re-



$i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = H|\psi(t)\rangle$
1st PARADIGM
Empirical Science
2nd PARADIGM
Theoretical Science
3rd PARADIGM
Computational Science, Simulations
4th PARADIGM
Big-Data-Driven Science
Experiments
Laws of classical mechanics, electrodynamics, etc.
Density-functional theory and beyond; molecular dynamics
Detection of patterns and anomalies in Big Data; artificial intelligence; etc.
1600
1950
2010

Figure 1.2: Material-design paradigms, over the years[23]

sources. Even for advanced super-computing clusters, DFT simulations are intense. Thus, machine-learned energy predictors are introduced to ease the design process. These machine-learning methods, once considered black boxes, have recently been extensively used for physics and quantum chemistry problems, to predict, for example, band gaps, NMR spectra, energies, phase diagrams, etc[25–28]. Thus, these methods can reliably be extended to our use case.

But, notwithstanding the utility and efficiency of said predictors, these can't merely read coordinates of atoms — input for DFT simulations — and reliably predict energy; translational, rotational, and permutational variance of the coordinates of atoms adds to the difficulty in training. Thus, a system needs to be implemented that accommodates such requirements, while being lossless.

Enter: Descriptors[24]. These are the special systems that define the atoms in a format amenable for machine learning. They fit the criteria described in 2.3. Many descriptors have been proposed, for example, Atom-Centered Symmetry Functions (ACSF)[29], Bag of Bonds (BoB)[30], Coulomb Matrix (CM)[31], MBTR[32], and SOAP[33]. The performance of these descriptors varies with use cases; for our applications SOAP and MBTR perform well, when compared to other techniques[34]. Hence, they are chosen for adding further complications.

Descriptor based learning is often employed in material research; this overcomes the inherent lack of geometric invariance in machine-learning techniques. MBTR[32], SOAP[33] (with SIN-Group's SOAP-lite[34]) are two such successful descriptors for machine-learning. The descriptors designed in this thesis are complications of these descriptors; these complications aim to add functionality — MBTR from local perspective — and generality — element-agnostic approach to SOAP.

1.1 Structure of thesis

With the background above, the thesis delves into descriptor designing for effective machine-learning in catalyst application. The following chapter presents a comprehensive background for the thesis. It explains SOAP and MBTR descriptors, theories of physics, and machine learning, which are the basis for newer descriptors. It is followed by methods, which discusses the newer descriptors developed, and the environment of the thesis. The next chapter delineates implementation and evaluation of said descriptors; and the chapter following that comprises discussions arising from the implementations. The thesis ends with the conclusions drawn from this exercise and an outlook for further work.

Chapter 2

Background

A significant pre-requisite knowledge is necessary, to design the descriptors described in the thesis. This comprises knowledge of the theories of electronic-structure methods, machine-learning — and data-science — practices, and the descriptors on which the newer are based. These are further discussed in this chapter, in that order.

2.1 Electronic structure theory

An electronic structure is the state of electrons and nuclei in an atomic system[35]. It is calculated by representing the states into a complex-valued probability amplitude, called wave function, and solving the Schrödinger's wave equation (Time-independent, in the thesis). This allows calculation of physical observables, primarily energies.

$$\hat{H}|\psi\rangle = E|\psi\rangle \tag{2.1}$$

Where $\hat{H}$ is the Hamiltonian, E, energy, is the eigenvalue, and ψ, wave function, is the eigenvector. However, these equations are complex and their calculation is computationally intensive. Hence, several approximations are introduced to ease the computation. These approximations are detailed in following sub-sections.

2.1.1 Born-Oppenheimer approximation

Born-Oppenheimer approximation[36] assumes that the contribution of nuclei and electrons is separable, since the masses of ions and electrons are of different orders of magnitude. This implies the electronic and nuclear wave function can also be separated.

$$\psi_{total} = \psi_{electronic} \otimes \psi_{nuclear} \tag{2.2}$$

The latter term in equation 2.2 can be approximated in a pseudo-potential comprising nuclear core of the atomic charge, along with the inner-shell electrons. Thus, the solution of Schrödinger's wave equation relies primarily on a Hamiltonian based on the valence electrons, which greatly simplifies the computation.

2.1.2 Basis set

The electronic part of wave functions, cf. eq 2.2, is represented by a basis set. The Slater type orbitals (STO) appropriately represent atomic orbitals. However, they decay as $e^{-\alpha \cdot r}$ and the integrals of such functions are convoluted and computationally intensive. Thus, they are replaced by the gaussian type orbitals (GTO); since product of two gaussians is a gaussian that makes the integration straightforward. A GTO is given as:

$$\psi_i(\vec{r}) = R_i(r) \cdot Y_{l_i,m_i}(\theta, \psi) \tag{2.3}$$

Here Y_{l_i,m_i} is the spherical harmonics for angular part and $R_i(r)$ is the radial part, which is:

$$R_i(r) = r^{l_i} \cdot exp(-\alpha_j \cdot r^2) \tag{2.4}$$

Multiple gaussians are contracted together, to accurately mimic a STO.

$$\psi^{contracted}(\zeta) = \sum_{i=1}^{L} \beta_i \psi_i^{gaussian}(\alpha_i)$$

STO-3G basis, is an example of 3 contracted GTOs.

$$\begin{aligned}\psi^{contracted}(\zeta = 1, STO - 3G) &= 0.444635\psi^{gaussian}(0.109818)\\ &+ 0.535328\psi^{gaussian}(0.405771)\\ &+ 0.154329\psi^{gaussian}(2.22766)\end{aligned}$$

This closely fits the hydrogen's 1s STO ($\zeta = 1$):

$$\psi^{slater} \simeq (\zeta^3/\pi)^{\frac{1}{2}} e^{-\zeta|r-R_a|}$$

The fit is also shown in fig 2.1.

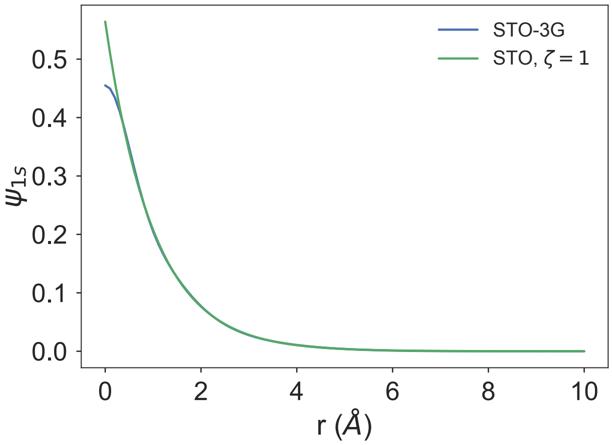


Figure 2.1: A fit of STO basis using STO-3G basis ($\zeta = 1$)

2.1.3 Hartree-Fock matrices

The Hamiltonian in equation 2.1 can be expressed as a Hartree-Fock matrix. This matrix includes contributions from the kinetic, core-potential, and electron-electron interactions (coulomb repulsion). These terms are a function of the wave function. If the chosen basis set is not orthogonal — which is not, in our case — it arises a need for an overlap matrix, during diagonalisation of the Hamiltonian. This overlap matrix is given by

$$S_{\mu,\nu} = \int dr \psi^*_\mu(r - R_A)\psi_\nu(r - R_B) \tag{2.5}$$

$$= \sum_{p=1}^{L}\sum_{q=1}^{L} \beta^*_{p,\mu}\beta^*_{q,\nu}\langle\mu|\nu\rangle \tag{2.6}$$

This is used to calculate the symmetric orthogonalising transformation-matrix, given by

$$X = S^{-\frac{1}{2}} = Us^{-\frac{1}{2}}U' \tag{2.7}$$

where U is the eigenvector, and s is the eigenvalue of S.

The core Hamiltonian is the sum of the kinetic and nuclear potential terms

$$H^{core} = T + V^{nuclear} \tag{2.8}$$

where

$$\begin{aligned} T_{\mu,\nu} &= \int dr \psi_\mu^*(r - R_A) \left(-\frac{1}{2}\nabla^2 \right) \psi_\nu(r - R_B) && (2.9) \\ &= \sum_{p=1}^{L} \sum_{q=1}^{L} \beta_{p,\mu}^* \beta_{q,\nu}^* \langle \mu | -\frac{1}{2}\nabla^2 | \nu \rangle && (2.10) \end{aligned}$$

and

$$\begin{aligned} V_{\mu,\nu}^{nuclear} &= \int dr \psi_\mu^*(r - R_A) \left(-\sum_a \frac{Z_a}{|r - R_a|} \right) \psi_\nu(r - R_B) && (2.11) \\ &= \sum_{p=1}^{L} \sum_{q=1}^{L} \beta_{p,\mu}^* \beta_{q,\nu}^* \langle \mu | \frac{Z}{r} | \nu \rangle && (2.12) \end{aligned}$$

From this, the Fock matrix is evaluated as:

$$F_{\mu,\nu} = H_{\mu,\nu}^{core} + 2 \cdot J_{\mu,\nu} - V_{\mu,\nu}^x \qquad (2.13)$$

where $J_{\mu,\nu}$ is the coulomb interaction matrix, and $V_{\mu,\nu}^x$ is the exchange matrix. These are given as:

$$J_{\mu,\nu} = \sum_{\lambda,\sigma} P_{\lambda,\sigma} \langle \mu, \nu | \lambda, \sigma \rangle \qquad (2.14)$$

$$V_{\mu,\nu}^x = \sum_{\lambda,\sigma} P_{\lambda,\sigma} \langle \mu, \sigma | \lambda, \nu \rangle \qquad (2.15)$$

This fock matrix is transformed to the orthogonalised basis set, and then diagonalised

$$\begin{aligned} F_o &= X' F X && (2.16) \\ F_o C_o &= C_o \epsilon && (2.17) \end{aligned}$$

The eigenvectors C_o are transformed back to give C, which is used to make a new density matrix.

$$C = X C_o \qquad (2.18)$$

$$P_{\mu,\nu} = 2 \sum_{a}^{\frac{N}{2}} C_{\mu,a} C_{\nu,a}^* \qquad (2.19)$$

2.1.4 Self-consistent field

A self consistent field (SCF) method is used to iteratively solve for the density matrix. The steps goes as follows:

1. Guess an initial density matrix
2. Calculate the Hamiltonian matrix, using the density matrix
3. Transform and diagonalise the Hamiltonian matrix
4. Transform the eigenvectors back
5. Derive a new density matrix

These steps are consistently performed, until a change in the derived density matrix is under certain threshold.

2.1.5 Density functional theory

The expression of Hamiltonian in terms of the Hartree-Fock matrix, however straight-forward, overestimates the energies; this under-binds the interactions[20]. The total energy derived from the Hartree-Fock method, thus, becomes an upper-bound for the actual ground-state total energy. To remedy this, a correlation potential is added which, along with the exchange term, aims to cancel any spurious self-interaction terms from the coulomb-repulsion matrix. This describes the solution to the Schrödinger wave equation as given in the Kohn-Sham formalism[19], cf. eq 2.1, which too is iteratively solved using SCF.

$$\left(T + V^{nuclear} + J^{ee} + V^{ex}\right)|\psi\rangle = E|\psi\rangle \tag{2.20}$$

Where T, $\mathrm{V}^{\mathrm{nuclear}}$, J^{ee}, and V^{ex} are the kinetic, core-potential (or external-potential), coulomb-repulsion (or two-electron), and exchange-correlation terms, respectively.

For the thesis, the adsorption energy calculations, using the ground-state total energies, are performed on CP2K[37] (version 4.1). It is a DFT atomistic simulation tool that relies on the GTO methods. This provides a minute control over output parameters — observable or non-physical; that helps in the training of machine-learning tools from different variables.

2.2 Data pre-processing and machine-learning techniques

This thesis employs several data pre-processing and machine learning techniques to draw out the inherent correlations in a dataset, within itself or with another property. This section discusses these techniques.

2.2.1 Mutual Information

The MI[38] signifies the amount of information a variable derives from another variable. MI between two discrete random spaces, X and Y, can be defined as:

$$I(X, Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) \tag{2.21}$$

Where p(x) and p (y) are the marginal probability distribution functions of X and Y, respectively, and p(x,y) is the joint probability function of X and Y. In our application, the X (Y) space is the discriptor (observable) space. This acts as a figure of merit, and quantifies the suitability of descriptor of interest.

We implement our MI[39] as described by Kraskov et. al.[40]. It works with high-dimensional feature and objective spaces, and makes the optimisation for hyper-parameters in descriptors, faster than comparing results from machine-learned models. The disadvantage though, of this implementation, is that: (a) comparing sets of features with unequal sizes is not straightforward, and (b) it depends on injective-ity of the descriptor and the target spaces — discussed further in chapter 5.

A MI, in our implementation, is not scaled; it is a function of size of dataset. Therefore, for 10000 data-points, or instances — used interchangeably — it ranges between 0 and 7.5. Therefore, the MI is calculated for random samples and for SOAP-lite descriptor as a reference. These values are in table 2.1.

2.2.2 Principal component analysis

An input-space matrix comprises multiple instance vectors stacked row-wise, and each instance vector contains the values of our features for that instance. However, these features can be correlated, which implies same information can be carried by fewer features. Here, a principal component analysis (PCA)

Table 2.1: MI values for reference

Dataset	**Observable**	**MI**
AuCu	Energy	1.52
	Charge	1.81
MoS_2	Energy	1.42
Random sample		0.85

transforms our input-space matrix into linearly-uncorrelated principal components. This transformation is performed such that the principal components are stacked with decreasing variance. Thus, the features can be reduced by truncating the principal components at a certain threshold in the variance.

PCA is computed by making a covariance matrix of the transpose of input space — into a matrix with each instance vector stacked column-wise. This covariance matrix is then diagonalised. Each eigenvector, of the diagonalisation, gives the coefficients corresponding to a linear combination for that principal component; the corresponding eigenvalue represents a figure of merit proportional to the variance in that principal component. Finally, the principal component matrix is derived by a dot product between the input-space matrix and the eigenvector matrix.

2.2.3 Kernel Ridge Regression

Kernel Ridge Regression (KRR) is a simple regression tool, that relies on a kernel (cf. eq 2.22) and data to behave as, or “learn”, polynomial (or gaussian) functions in space. Its ease of application makes it a prime tool for analysis on small datasets.

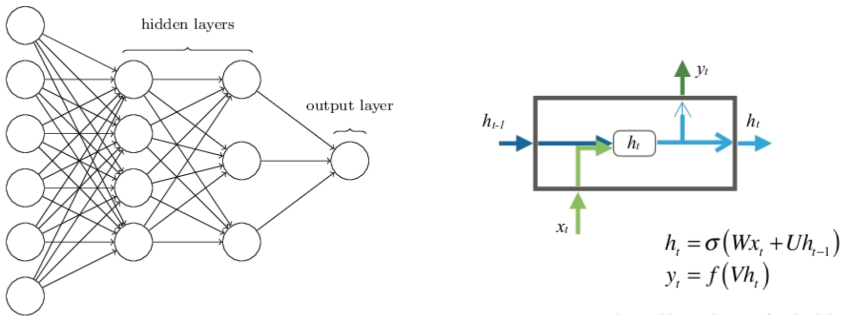
To start, we measure the distance between two instances, by using a kernel function d. This function is then used to calculate the kernel matrix, given as:

$$K_{ij} = d(X_i, X_j) - \alpha I_{ij} \tag{2.22}$$

for all the pairs of samples X_i and X_j in a training set. Here, I_{ij} is the identity matrix and α is a regularisation-parameter chosen between 0 and 1. After deriving the kernel matrix, the model is ready to predict newer instances. This is done in the similar fashion to when we trained.

$$D_{ij} = d(X'_i, X_j) \tag{2.23}$$

where X'_i is an instance of the new samples. Now, the prediction, Y', is given



(a) A typical multi-layer perceptron

(b) A simple RNN cell, being applied at t$^{\text{th}}$ instance

as

$$Y' = Y \cdot K^{-1} \cdot D^T \tag{2.24}$$

where Y is the vector of known outputs for the training set X. This method is applied for training all descriptors with constant feature sizes: LMBTR, BLEACH overlap matrix, and SOAP-lite. Since this method involves inverting a N × N matrix, where N is the number of instances, KRR can't be scaled easily for larger data-sets.

2.2.4 Multi-Layer Perceptrons

Multi-Layer Perceptrons (MLP) are the simplest of the constructs of neural networks. They consist of many hidden layers, comprising nodes, with their wights and biases, see fig 2.2(a). They can be trained by several optimisation techniques, like backpropagation, using BFGS, Adams, etc, or by an evolutionary algorithm. Once trained, they act as a high order polynomial function between the descriptor and objective space. The "depth" — number of hidden layers — of a MLP can be increased to make the network learn higher order polynomial relationships, by its massive non-linearly-activating nodes.

2.2.5 Recurrent neural network

Recurrent neural network (RNN) are artificial neural networks which operate on a sequence of data, definite or indefinite. They possess an internal state, see fig 2.2(b), which carries additional information of the past nodes, analogous to memory, to ones ahead. It makes multiple connections between nodes along a sequence; this forms a directed graph. It is primarily suited to find correlations in a stream of temporal data. Moreover, when applied to a function, it can also quantify the changing gradient over space.

In our application, a simple RNN is chosen, with one internal state. The input at every step is calculated as:

$$h_t = \sigma(Wx_t + Uh_{t-1}) \tag{2.25}$$

$$y_t = f(Vh_t) \tag{2.26}$$

Where, in the $t^t h$ instance, h_t is the internal state, x_t is the input, and y_t is the output. W, U, V and f, are the matrices and functions, whose variables are learned.

2.3 Descriptors

A descriptor is a representation of an atomic system. For a complete — or usable — representation, the descriptor adheres to a set of guidelines. In their paper, Huo et.al. [32] summarizes the properties that makes the descriptor desirable. These include:

- invariant to translation, rotation and atom permutation
- unique
- continuous
- general
- computationally cheap
- lossless (efficient)

Although, some application may require breaking of these properties — for example, predicting vector observables, such as force calculation, needs rotational variance — these guidelines are comprehensive and relevant to any descriptor design. The descriptors investigated in the thesis are based on SOAP and MBTR, with the aim to improve on them, for nano-catalyst applications. The following subsections, discusses SOAP and MBTR[41].

2.3.1 SOAP

SOAP is a local descriptor, that maps the local environment of atoms, around a point, very accurately — as much afforded by pre-selected angular-momentum terms. It affords prediction of a local observable, for example,

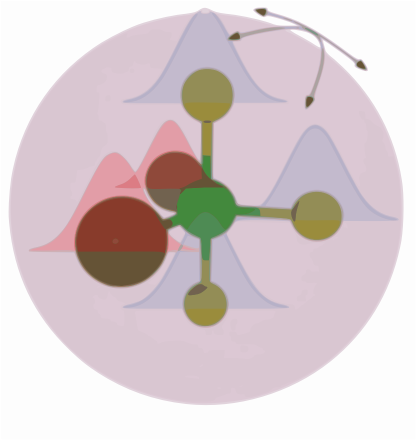


Figure 2.2: A schematic representation of the smooth overlap of atomic positions with the basis of centre atom[34]

charge, or adsorption energies. This makes it a suitable candidate for our applications.

It is rotationally, and permutationaly invariant. This is achieved by projecting the atoms onto spherical harmonics, centred at the point of interest. However, a projection of point sized neighboring atoms would need a wastefully high number of angular momentum terms. Thus, an overlap of smoothed out atomic positions, by gaussian smearing, are used. However, this also makes all the elements indistinguishable. Thus, SOAP is calculated for individual element-type; which are concatenated at the end.

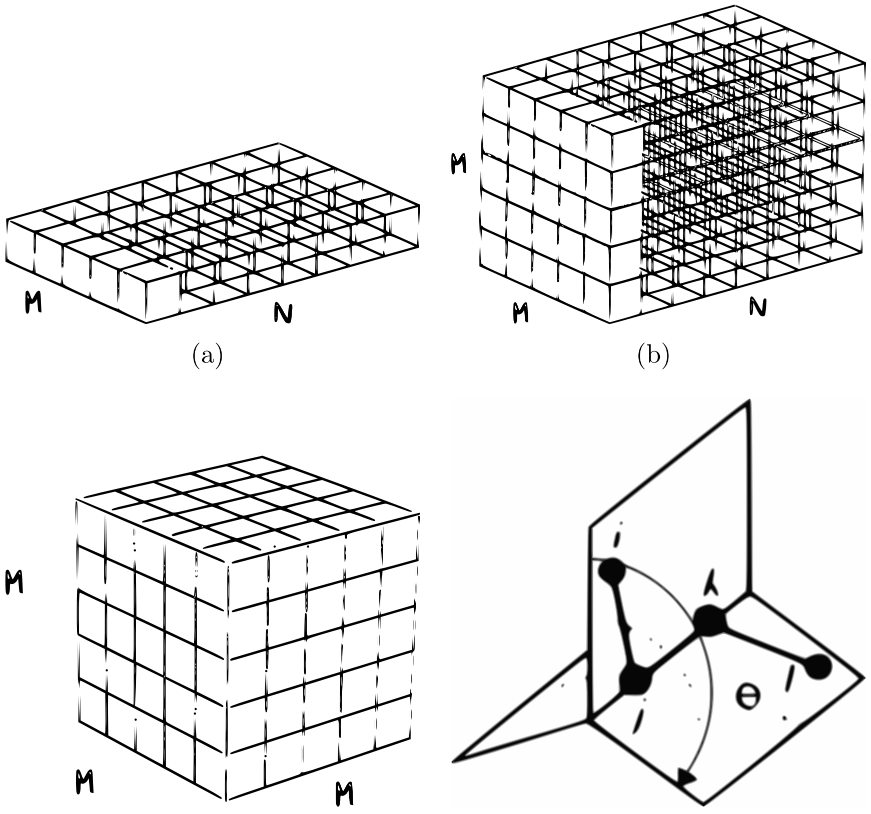
$$\rho^{\alpha}(r) = \sum_{i} e^{-(r-r_i^{\alpha})^2} \tag{2.27}$$

Where r$_i^{\alpha}$ is the position of ith atom of the α element. The obtained smeared atomic position, cf. atomic density, is decomposed using Laplace Spherical Harmonics — spherical harmonics in real space — and orthogonal basis set: $\Upsilon_{lm}(\theta, \psi)$ and $g_n(r)$. This maps them into coefficients of orthornormal basis functions used, see fig 4.1.

$$c^{\alpha}_{nlm} = \langle \rho^{\alpha} | g_n(r) \Upsilon_{lm} \rangle = \int_V g_n(r) \Upsilon_{lm}(\theta, \psi) \rho^{\alpha}(r, \theta, \psi) dV \tag{2.28}$$

The coefficients thus derived, are used to calculate a power spectra, and summed for all m's for rotational invariance.

$$P^{\alpha}_{nn'l} = \sum_{m} c_{nlm} c^{\alpha *}_{n'lm} \tag{2.29}$$



(a) (b)

(c) (d)

Figure 2.3: Schematic representations: (a) K_1 tensor; (b) K_2 tensor; (c) K_3 tensor (each box is a spectrum); (d) dihedral angle, when calculating K_4 tensor

2.3.1.1 SOAP-lite

SOAP-lite is the analytical implementation of SOAP. It is implemented upto $l = 9$, uses application-tailored basis functions, and is entirely written in C — with python wrapper. These three features combined, makes this implementation swift, and accurate[34].

2.3.2 MBTR

MBTR is a global descriptor for a molecule/crystal. It is translational, rotational, and permutation invariant; it forms tensors of combination of elements in pair, triplets, quadruplets, etc — called K_1, K_2, K_3, K_4, [. . .], respectively. The implementation used in thesis is from the Describe package[42], and hence, K is restricted to 3. All such combinations have a Gaussian-smeared exponentially-weighted histogram — spectra, henceforth— associated to it. Due to its global nature, it can predict global observables, like total energy, enthalpy change etc. The disadvantage of this approach is that it needs high number of features to be relevant for machine learning applications.

K_1 represents the spectrum of counts of element types, indexed by their atomic weights. Although, this puts an implicit bias, that atom similarity is a function of difference in atomic numbers. In end, it becomes a matrix of size $M \times N$, where M is the number of elements, and N is the number of bins, see fig 2.3(a).

K_2 represents the spectrum of inverse distances between pairs of element types. So, it becomes a matrix of size $M \times M \times N$, where M is the number of elements, and N is the number of bins, see fig 2.3(b).

K_3 represents the spectrum of angles between triplets of element types. So, it becomes a matrix of size $M \times M \times M \times N$, where M is the number of elements, and N is the number of bins, see fig 2.3(c).

K_4 represents the spectrum of dihedral angles (see fig 2.3(d)) between quadruplets of element types. So, it becomes a matrix of size $M \times M \times M \times M \times N$, where M is the number of elements, and N is the number of bins.

Weighting All the tensors, but K_1, are weighted. This diminishes the contribution from farther atoms, and hence, closer atoms have higher precedence. The Describe package implements exponential weighting by default.

Chapter 3

Methods

This chapter describes the methods, developed for this work, used to define newer descriptors and to evaluate machine-learning errors. The descriptors include: (a) SOAP based on charge density, numeric and analytic, (b) BLEACH based on overlap matrix, coulomb repulsion, fock matrix, and fock tensor, and (c) Local Many-Body Tensor Representation (LMBTR).

3.1 Descriptor design

3.1.1 SOAP based on charge density

SOAP is a powerful tool, as it accurately maps local surrounding with remarkable accuracy. However, it loses information by its treatment of neighboring atoms as gaussians of the same weight. Although it is remedied by separately calculating the power spectra for individual element types, this makes it lose its generality. More specifically, if a model is already trained with certain elements, an introduction of new element will render the model useless, and the training cycle would need to be restarted.

To overcome this, the idea is to replace the unit-amplitude gaussian with DFT-derived charge densities. From fig 3.1, the similarity of Gaussian distribution, and real charge density promotes the investigation in this direction. These charge densities are derived numerically and analytically.

- Numerical derivation
  Charge densities are interpolated from the ones directly printed by CP2K.

- Analytical derivation
  Charge densities are calculated analytically from density matrix, printed by CP2K, and GTO basis. This is carried out as follows:

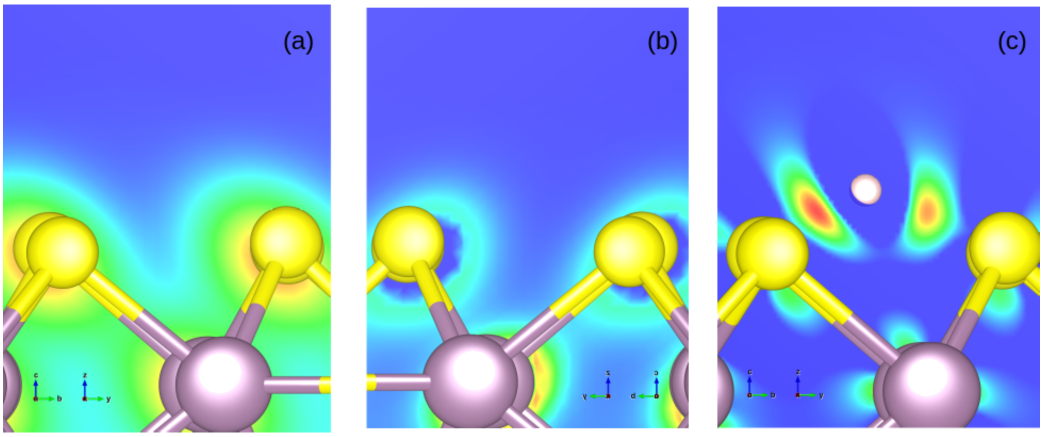


Figure 3.1: The images show similar environments with different calculations of charge distributions; (a) The gaussian smeared atoms of nano-cluster, around the hydrogen atom, (b) The charge distribution, calculated using DFT, around hydrogen, and (c) Charge distribution of hydrogen over nano-cluster subtracted by, charge distribution of hydrogen, sans the nano-cluster, and charge distribution of nano-cluster, sans the hydrogen. Purple balls are molybdenum , while yellow are sulfur, and grey is the hydrogen atom.

$$\rho(r) = Trace\big(\langle\chi_{r'}|\rho|\chi_r\rangle\big) \tag{3.1}$$

where $|\chi_r\rangle$ is the molecular orbitals (MO) vector at r, and ρ is the density matrix

Further, partial density of states (pDOS) for hydrogen (fig 3.2) indicate that, most hydrogen orbitals exist very near to the Fermi-level When compared to difference in pDOS for S and Mo, with and without the adsorbed hydrogen, it is evident that most of the meaningful interactions happen with these select MOs. Thus, another method is implemented, that scours the unwanted orbitals from the density matrix. More specifically, a new density matrix is created ignoring the molecular orbitals outside the energy window. This method is chosen over merely selecting d-bands, as proposed by Hammer et. al.[43], since, it accounts for hybridisation in the orbitals. The new charge density is calculated as:

$$\rho(r) = Trace\big(\langle\chi_{r'}|\rho_{selectMO}|\chi_r\rangle\big) \tag{3.2}$$

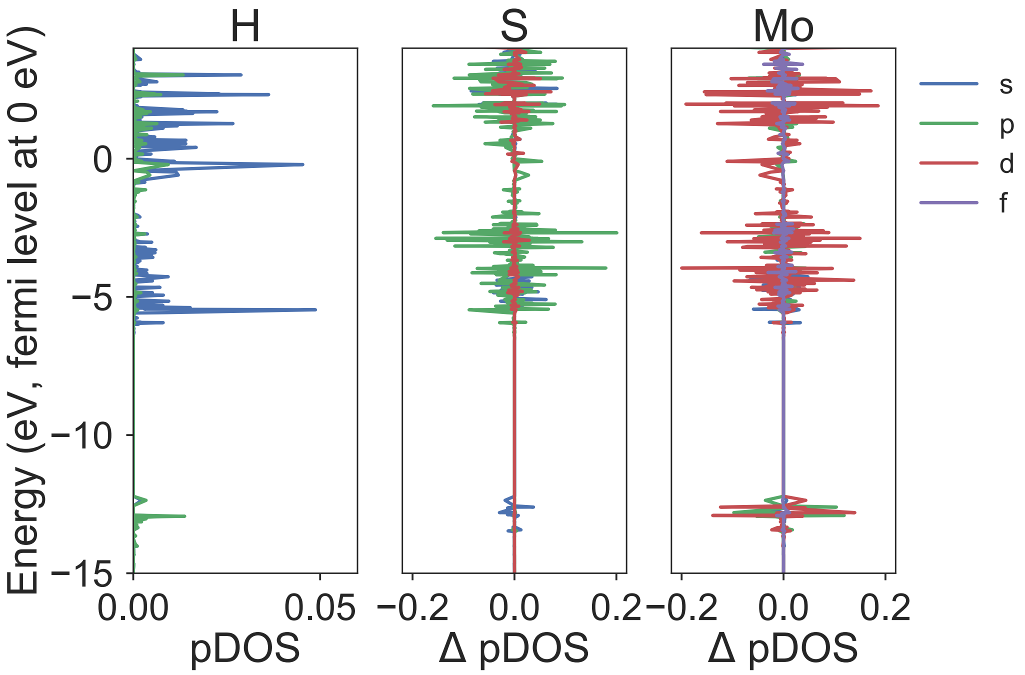


Figure 3.2: Example of pDOS for hydrogen (left), compared to difference in pDOS for S (mid) and Mo (right), with and without the adsorbed hydrogen

3.1.2 BLEACH

The idea of BLEACH arises from applying SOAP on charge densities. Calculating power-spectrum (eq 2.29) is very close to how orbital integrals work. This initiates the idea to instead do the entire calculation using analytical orbital integrals. For our analysis, we develop BLEACH using different aspects of electronic structure approaches, specific to Hartree-Fock methods — for ease in first adoption. Hence, overlap matrix, coulomb-repulsion matrix, and fock matrix, from gaussian type orbitals, are implemented.

- Overlap matrix
  Overlap matrix is the simplest form of interaction possible. It is calculated as (cf. 2.6):
  $$S_{ij} = \langle i|j\rangle \tag{3.3}$$
  where i is the basis of hydrogen and j is the basis of all neighbouring atoms.

- Coulomb repulsion, with density matrix with all or select MO
  Coulomb repulsion is the dot-product of density matrix and two-center

two-electron integral matrix (cf. 2.14), given as:

$$J_{ij} = \sum_{kl} \rho_{k,l} \langle i, j | k, l \rangle \tag{3.4}$$

Where i and j are hydrogen atom's orbitals, and k and l are the orbitals of atoms in nano-cluster — screened upto a cut-off. They are slightly more involved than overlap matrix, and their implementation is computationally heavy.

Similar to reasoning behind eq 3.2, the coulomb repulsion is calculated by density matrix made of select MO, given as:

$$J_{ij} = \sum_{kl} \rho_{selectMO_{k,l}} \langle i, j | k, l \rangle \tag{3.5}$$

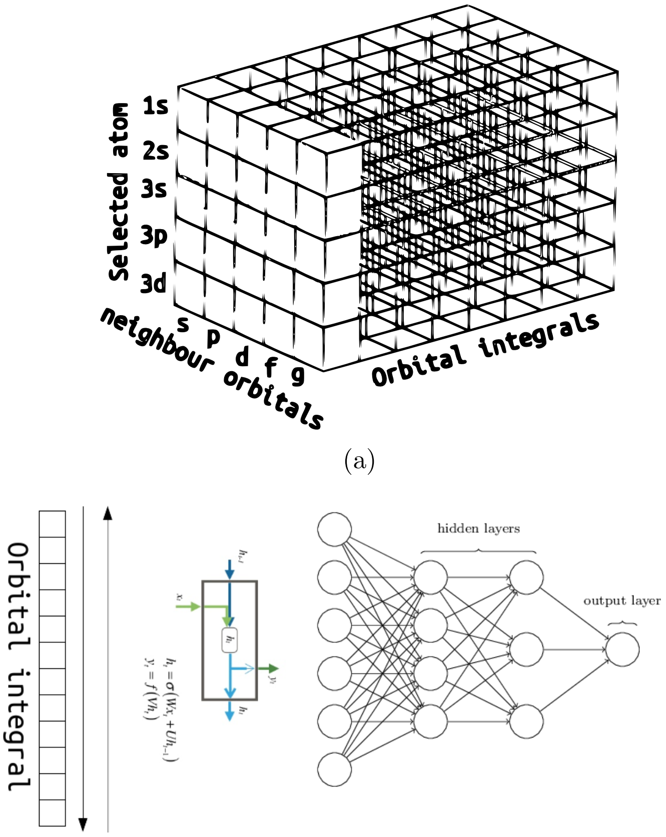
- Fock matrix
  To include further information in the model, BLEACH made from entire Fock matrix is investigated. This involved adding the kinetic and core-potential matrices to coulomb repulsion, given as (cf. 2.13):

$$F_{ij} = \langle i | \frac{-1}{2} \nabla^2 | j \rangle + \langle i | \sum_{c} \frac{Z_c}{r} | j \rangle + \sum_{ij} \rho_{k,l} \langle i, j | k, l \rangle \tag{3.6}$$

- Fock tensor
  In our case, the unknown is really the part of density matrix that corresponds to the hydrogen's orbital interactions with orbital of itself and other neighboring atoms. It indicates the amount of electrons present in each interaction. However, in the previous approaches, when those matrices are calculated, that part of density matrix is entirely approximated to 1. Thus, what we see is interaction of orbitals, filled or empty. And adding a priori weight to the summed elements seems complicated.

  The easiest way to solve the issue, in our opinion, is to not sum the interactions, and let neural-network learn from scratch. So, the idea becomes, that we split those interactions into corresponding orbitals, and lay them into a vector of orbital integrals — sorted, to naively add permutation invariance — see fig3.3(a).

  To each orbital-integral vector, a bi-directional RNN is applied. This forms a two-way directed graph, between subsequent orbital interaction. The outcome of which is fed into a MLP to predict the observable. The architecture of the model is presented in fig 3.3(b).



(a)

(b)

Figure 3.3: (a) A schematic representation of Fock tensor; (b)The operation of bi-directional RNN followed by MLP on BLEACH tensors

3.1.3 LMBTR

LMBTR is the local flavor of MBTR, where the element-type based tensors are evaluated from the perspective of an atom. This drops one dimension of each tensor, as element-types, of that dimension, are replaced by an atom of unique element-type. In other words, a tensor, $M \times [\ldots]_n$, of n+1 dimensions, becomes a tensor, $1 \times [\ldots]_n$, which is equivalent to a tensor, $[\ldots]_n$, of n dimensions. Thus, K_1 loses its significance, since it only indicates the atom's element-type. K_2 becomes $M \times N$, cf. fig 2.3(a), which now represents the spectrum of inverse distances between the atom and different types of elements. Similarly, K_3 becomes $M \times M \times N$, cf. fig 2.3(b). Also, a corresponding change in spectra is observed. This shift, to the atom perspective, makes the tensor-representation a local descriptor, and hence, suitable for adsorption energy prediction.

Apart from adsorption energy calculations, LMBTR shows bi-directional operability. Since LMBTR values are tangible in real space its plot, for the geometrically optimised hydrogen positions with energy in the region of interest, can point out the correlations in atom's spacial environment. This aids in scouting trends inherent in the data.

3.2 Error analysis

The performance of a machine-learning model can be tricky to comprehend in the absence of a reference for the errors in their training. An accuracy of 0.1 eV, for example, can be considered promising for our case, however, with a deviation of 0.2 eV in our observables, this accuracy is inadequate in comparison. Thus, in this thesis, average absolute deviation (AAD) (based on mean absolute error (MAE)) is referred to gauge the accuracy of our machine-learned model. AAD — in place of standard deviation, and variance — is chosen as it is robust, and our model accuracies are also calculated in MAE.

Error plot is another tool for easing our understanding of MAE. It gives an intuitive perspective on dataset, which can promote better methods at machine-learning. In the thesis, three such error plots are used, namely:

- Parity plot
  It is the scatter plot between each predicted and observed value. This gives a general idea of quality of fit, when bias in the data results in accurate prediction in the regions outside our interest.

- Histogram-compare plot
  It is the bar plot comparing histograms of predicted and observed values. This gives an indication of clusters in our distribution, and the accuracy of machine in identifying such clusters.

- Density-based accuracy plot
  It is the scatter plot between MAE of the instances and the fraction of the instances in each bin of the histogram of observed values. This representation gives accuracy based on density of dataset. Well learned models show an exponential decay (in upper-limit of error) towards lower MAE, while not so well learned models are spread across at higher MAE.

Chapter 4

Implementation and evaluation

This chapter discusses the implementation of the methods described previously. It begins by declaring the environment, in which the thesis is performed. It then continues to the descriptors: charge density-based SOAP, BLEACH , and LMBTR.

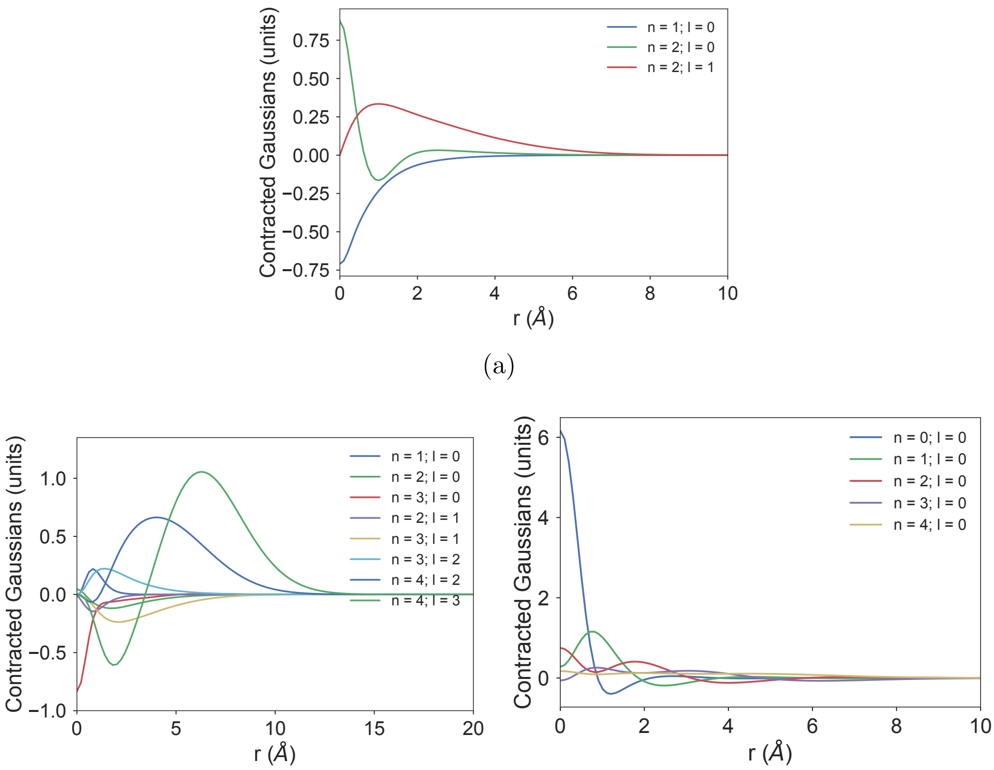
4.1 Environment

The works inscribed in this thesis, employs several tools, and programming architectures, which constitute the environment. This environment set-up is non-trivial, hence, for accurate and quick reproduction of the work, this section declares descriptor, machine-learning, and DFT simulation related environments.

4.1.1 Descriptor design

Code development for SOAP based on charge density (section 3.1.1) is done on C++ with armadillo[44]. All other codes, are developed in python, since, python is a high-level, interpreted language, that makes for an agile code development phase. Further, it is highly modular; python is made powerful by numerous packages that can be swiftly downloaded and installed from PyPi[45]. The work implements python packages such as: NumPy[46], SciPy[47], Pyscf[48], Matplotlib[49], Keras[50], Scikit-learn[51], Describe[42], and Seaborn[52].

Git[53] version control is also used, since, it streamlines, and un-complicates collaboration during the code development.



(a)

(b)

(c)

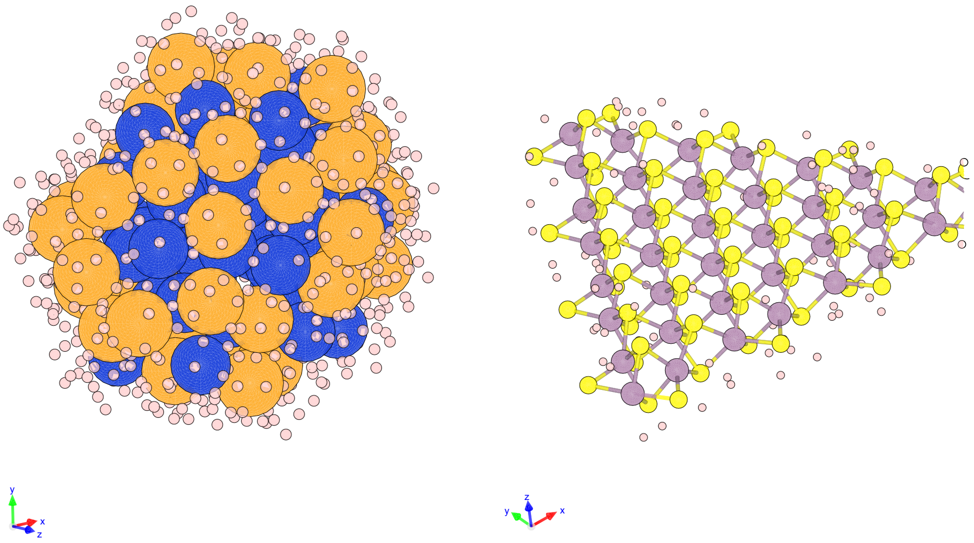
Figure 4.1: Radial part of GTO basis used: (a)DZVP-MOLOPT-SR-GTH H; (b) DZVP Mo; (c) SOAP-lite basis set.

4.1.2 Machine learning

Scikit-learn is used for simple machine learning architecture, like KRR. A high-level neural network API, keras — backended with tensorflow[54] — is used to set up MLP and RNN.

4.1.3 Basis set

Double zeta type functions (DZVP-MOLOPT-SR-GTH[55]) of H, and Mo are used for basis of H atom, in descriptor generation. Further, another basis set, developed for SOAP-lite, is also used, see fig 4.1. The SOAP-lite basis set introduces additional hyper-parameters, namely: number of radial basis (n), maximum angular momentum term ($l_{\max}$), and cutoff — to normalise the integration within.



(a) (b)

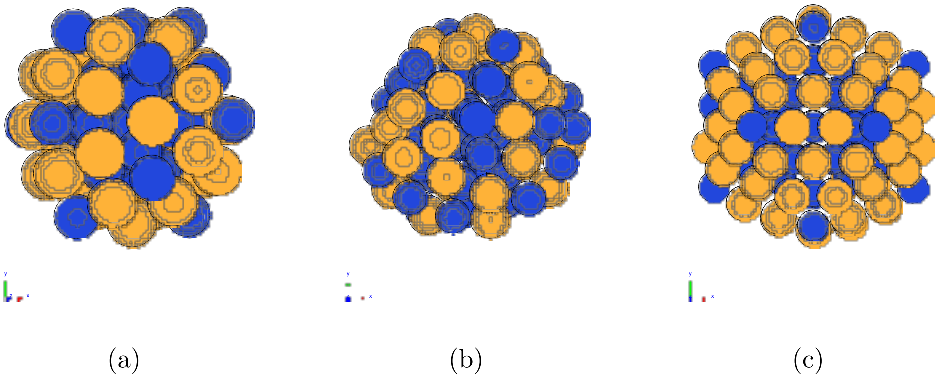
Figure 4.2: Example of database trained on: (a) AuCu nano-cluster with few (646) sparsely-chosen hydrogen adsorption site, and (b) One of the MoS_2 nano-cluster with random (110) hydrogen adsorption sites. Gold, blue, purple, and yellow circles represent Au, Cu, Mo, and S, respectively. And the smallest pink circles are hydrogen.

4.1.4 Database

All the implementation and evaluation of the descriptors are done on AuCu and MoS_2 nano-clusters. Each database has ten thousand calculations of adsorption energies, to aid the training of artificial networks. The AuCu database also has the magnitude of charge on hydrogen.

The AuCu nano-catalyst database comprises a single nano-cluster with ten-thousand random hydrogen positions. This makes for an easy database; due to vast number of hydrogen calculations, the entire feature space is readily covered, making predictions more reliable. On the other hand, MoS_2 nano-catalyst database has 91 different nano-clusters, with 110 hydrogen atoms per nano-cluster. This makes for a rather sparse space, and a challenging machine training. These observations are also evident from fig 4.2.

AAD of these databases are calculated, cf. table 4.1, to get a reliable reference of our MAE during training. It is evident, from the low AAD in



(a) (b) (c)

Figure 4.3: Example of (a) 55-atom, (b) 80-atom, and (c) 147-atom AuCu nano-cluster database to check bi-directional operability for LMBTR. Gold (blue) circles represent Au(Cu)

database, that the machine-learning model accuracy should be really high, for the model to be considered effective.

Table 4.1: Average absolute deviation in our database

Nano-cluster	**Observable**	**Average absolute deviation**
AuCu	Energy	0.3 eV
	Charge	0.077 e
MoS_2	Energy	0.52 eV

Also, to check the bi-directional operability of LMBTR, another database of geometrically optimised AuCu nano-cluster with singly adsorbed hydrogen is chosen, see fig 4.3. It comprises three different sizes of nano-clusters: 107 55-atom, 601 80-atom, and 187 147-atom nano-clusters, with varying concentrations of Au and Cu. Further, redundant hydrogen positions are removed from the database, by comparing SOAP-lite features. This makes the database, disperse, yet complete.

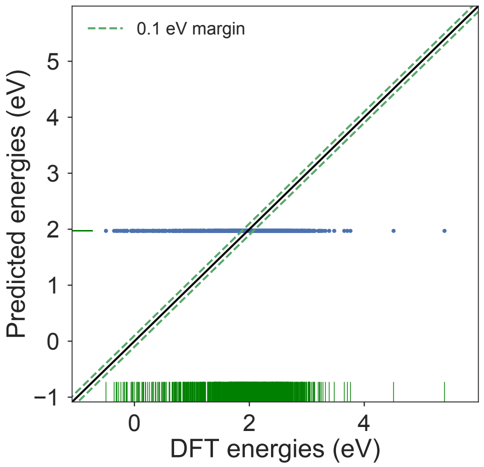


Figure 4.4: Parity plot when MAE is close to AAD of dataset.

4.1.5 DFT simulations

For Density functional simulations exchange-correlation function is approximated with PBE[56]. Gaussian type orbitals of Double Zeta type are used, with a core potential of type GTH-PBE. Van Der Waals potential of type DFTD3(J) are also added. The fineness of the grid is increased with requirement of finer numerical charge density file. SCF calculations are converged till energy accuracy of 1 e^{-6} eV. These simulations are further used to calculate the molecular orbitals, partial density of states, charge densities, and overlap matrix. All simulations for AuCu and MoS_2 database described in section 4.1.4 are static-point calculations. Furhter, to illustrate bi-directional operability of LMBTR, geometrically optimised structures are used. These geometric optimisations are performed using the Broyden-Fletcher-Goldfarb-Shanno(BFGS) algorithm.

4.2 SOAP based on charge density

As discussed in section 3.1.1, SOAP based on charge density is implemented two ways: numerically and analytically. These implementations are modeled after SOAP-lite as they exactly follow the derivation of coefficients, by integrating densities on grid given by Gaussian Quadrature rule, and calculation of power spectrum from these coefficients. However, they differ in the interface by which the densities are calculated on the grid points. These implementation are carried as followed.

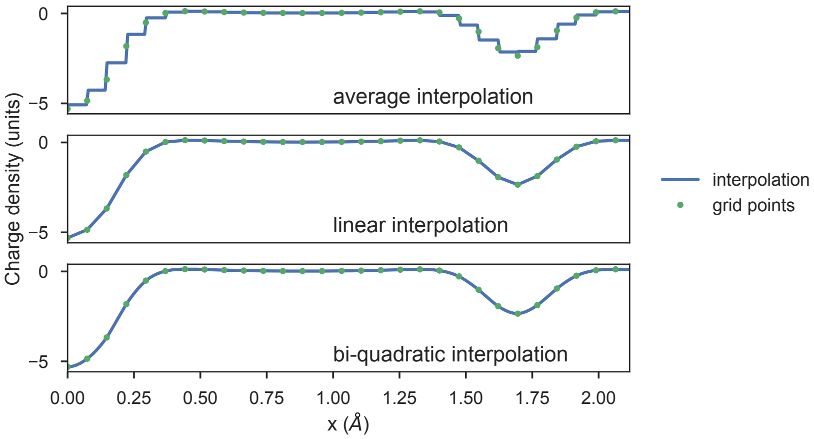
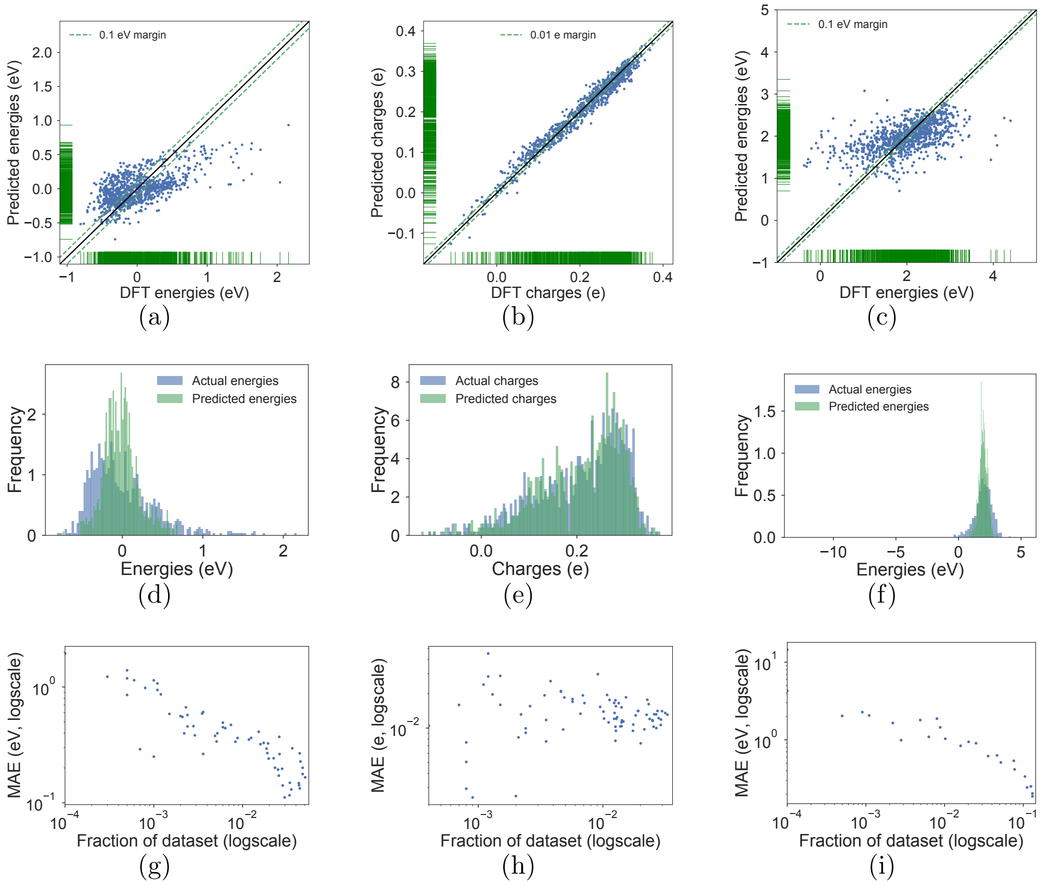


Figure 4.5: Average (top), linear (mid), and bi-quadratic (bottom) interpolations for fitting charge densities. The larger drop is a S atom, and other is a H atom (cf. fig 3.1)

4.2.1 Numerical

In this implementation, the charge densities values are read on a specific grid. Since, the grid queried for integration mismatches the specific grid, average, linear, and bi-quadratic interpolations are used. As seen in fig 4.5, the accuracy of fit increases from average to bi-quadratic, however, it also increases the computation time. The charge density on the specific grid is read from cube files, as printed in CP2K. In making the charge-density cube file, CP2K gives extreme control over voxel size; it is achieved by changing the energy cut-off for grid. In our case this energy cut-off is increased from 500 Ry. This corresponds to a voxel density of 384 voxels/$\mathring{A}^3$. This cut-off is capped at 1000 Ry — 8067 voxels/$\mathring{A}^3$ — since the sizes of individual cube file shoot up to 800 MB, which makes the implementation impractical. Further, a cut-off for the integration is optimised. This cut-off is increased from 5 Å until the change in descriptor value is within $1 \times e^{-4}$. Hence, a cut-off of 7 Å is chosen. With this implementation a maximum MI of 0.96 is seen. This implies no correlation, cf. table 2.1. It becomes more evident, when this is trained using KRR for MoS_2 dataset. The MAE stands at 0.52 eV, which is the AAD, cf. table 4.1, in our dataset.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.6: Plots of (a, b, and c) parity, (d, e, and f) histogram-compare, and (g, h, and i) density-based accuracy for learning energies (left column) and charges (middle column) on AuCu, and energies (right column) on MoS_2 nano-cluster database, using overlap matrix

4.2.2 Charge density using density matrix with all and select MO

This implementation follows the same principle as above: it returns charge density values quarried at certain grid points. However, here the charge density values are calculated analytically. Thus, the issues with interpolations are circumvented. The only hyper-parameter attached to this approach is cutoff — of the neighbour region. Similar to previous approach, cut-off is increased from 5 Å until the change in descriptor value is within $1 \times e^{-4}$, and hence a cut-off of 7 Å is chosen. The charge density calculations are performed using PYSCF. Since, PYSCF package is available in python, a python implementation of SOAP-lite is also carried out.

With this approach, the MI reaches 1.32 for MoS_2 dataset. This is found to be equal to that of SOAP without atom distinction. MAE in adsorption energy prediction comes at 0.51 eV, which is also the AAD, see table 4.1, in that dataset. The accuracy doesn't improve with using bigger basis set. Regardless of MAE a parity plot is made, for predicted v/s test observables, as shown in fig 4.4.

4.3 BLEACH

Since, BLEACH heavily relies on PYSCF, all codes are written in python. For convenience, a nano-cluster system is initiated in a BLEACH class, with objects, such as density matrix, or MO matrix, and methods that attach to PYSCF, functions — to calculate various integrals. It has the ability to take in basis sets in CP2K and SOAP-lite formats. A .create() method makes the descriptor of the system. MI is then calculated between the descriptor and the observable. Also, basic file parsers are written to parse CP2K's density-matrix, overlap-matrix, and MO-matrix files. In this section, different implementation of .create() method are discussed.

4.3.1 Overlap matrix

For the calculation of Overlap matrix, cutoff is the only hyper-parameter to optimise. It is similarly optimised as methods above. A cut-off of 7 Å is chosen, which guarantees most information of the neighbourhood, and keeps the implementation swift. The MI, reported in table 4.3, is higher than in previous attempt with charge density, and very close to SOAP-lite. Thus, it is trained on a KRR model.

The highest MAE, over different basis set, is achieved at 0.25 eV, 0.013 e, and 0.42 eV for AuCu energies and charges, and MoS_2 energies, respectively. These values matches our MI analysis, cf. table 2.1.

The plots in fig 4.6 reveal the underlying nuances in data. It is evident from parity plots for energy in fig 4.6 (left, and right columns) that the model fails at data with higher energy, due to lack of data-points. Further, it also is shown in the histogram that due to over abundance of data near mean, the predictions are centered there too. And, a disperse density-based accuracy plot reveals heavy clustering in energy values, that lead to very

Table 4.3: MI values for Overlap matrix implementation of BLEACH

Dataset		**MI**
AuCu	Energy	1.53
	Charge	1.94
MoS_2	Energy	1.40

few spots well represented, while others under-learnt. In the charge plots see fig 4.6 (middle column) the prediction is within the margins. This gives well overlapping histograms and hence, lower MAE.

4.3.2 Coulomb repulsion, with density matrix made from all and select MO

In this iteration, the function for overlap matrix is replaced with coulomb interaction, as given in section 3.1.2. Similar to other methods, a cut-off of 7 Å is chosen after optimisation. Different basis sets of H, Mo, and SOAP-lite are also used. Finally, for density matrix, all and select-MO are implemented. In the case of all MO the density matrix is read directly from CP2K; and in the case of select-MO the MO matrix is used. This matrix comprises individual eigenvectors stacked column-wise. The eigenvalues corresponding to these eigenvectors are their energies. These energies are shifted, so that Fermi-level — also given by CP2K — is at 0 eV The eigenvectors corresponding the energies, or eigenvalues, in our region of interest are kept. A dot product of transform — since MO values are real, conjugate-transform if complex — of select-MO and MO gives the required density matrix.

A maximum MI of 0.9 is achieved with all MOs and hydrogen basis. Since, this value is close to random sampling, cf. table 2.1, the method is abandoned for adding more information.

4.3.3 Fock matrix

To incorporate further interaction, Fock matrix is proposed in section 3.1.2. Its implementation involves adding kinetic and core potential integrals to the BLEACH integrals. Similar to other methods, a cut-off of 7 Å is chosen after optimisation.

A maximum MI of 1.2 is achieved, using hydrogen basis. It is a slight improvement from previous BLEACH implementations, however, this is still

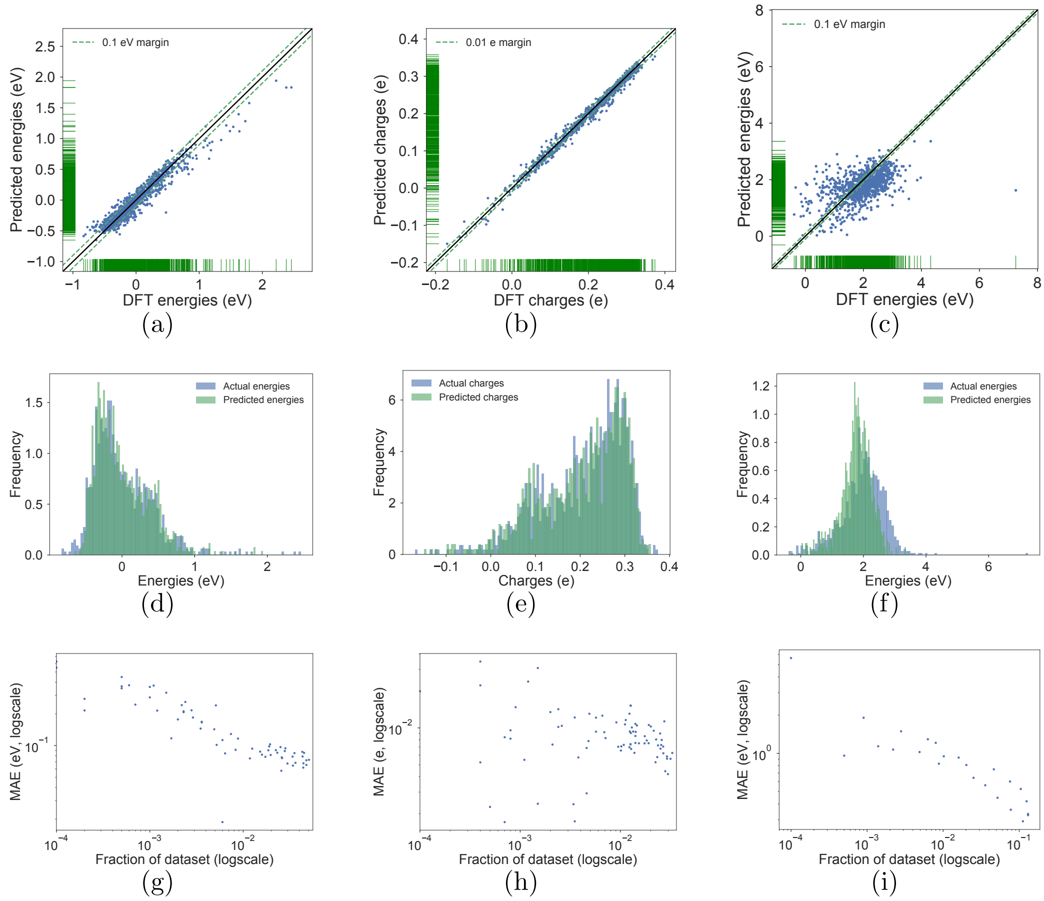


Figure 4.7: Plots of (a, b, and c) parity, (d, e, and f) histogram-compare, and (g, h, and i) density-based accuracy for learning energies (left column) and charges (middle column) on AuCu, and energies (right column) on MoS_2 nano-cluster database, using Fock tensor

not as promising a overlap matrix.

4.3.4 Fock tensor

A fock tensor is created, with the idea to have wighted sum for orbital interaction. Its implementation involves not summing orbital interactions in making BLEACH based on Fock matrix. This implies that the orbital integrals are kept separate into a tensor, cf. fig 3.3(a)(a). The integrals for one permutation of hydrogen's basis and other s, p, d, f in nano-cluster, are kept in one vector. To remove permutation invariance, these vectors are also sorted. The orbital integrals vectors vary in lengths, because the counts of orbital interactions are different. To effectively learn such input, a keras model, of a RNN-MLP architecture (cf. fig 3.3(a)(b)), is applied.

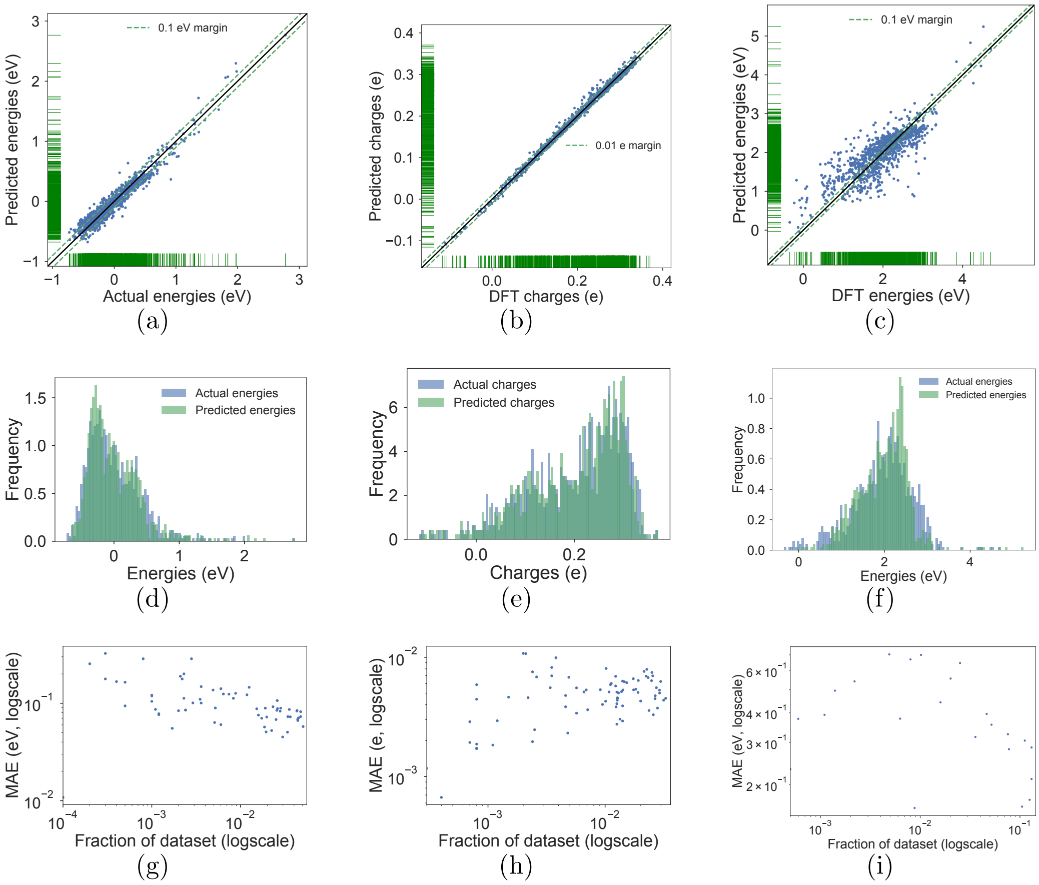
The implementation of MI does not work with arbitrary length matrices. Hence, to get a naive idea of MI, the fock tensor is padded with 0. The MI between padded fock tensor and observable is shown in table 4.5.

Multiple trainings with different sizes of RNN-MLP architecture are performed, while considering H, Mo, and SOAP-lite basis sets. The MAE in this process, is minimum at 0.088 eV, 0.013 e, and 0.42 eV for AuCu energies and charges, and MoS_2 energies, respectively. This is consistent with the MI. The architecture, that achieved least MAE, comprises a bi-directional RNN with 64 filters per orbital-integral vector, followed by three neural networks of 65×64 (bias included) dimension.

The plots in fig 4.7 also corroborate the findings. The parity charts show a fit well within margin for charge prediction for AuCu. However, this is not seen in other two. Further, the histogram-compare shows that in case of charge prediction the LMBTR learns the two different clusters — the two maximums — in our database. Histogram compare reflects MAE as it shows that the energy prediction in MoS_2 database completely mispredicts higher energy case, even when ample data is provided. Instead, the model predictions cluster more towards the mean. The density-based accuracy show that, while energy predictions can improve with further data, i.e. high variance, the charge prediction will not, i.e. low variance.

Table 4.5: MI values for Fock tensor implementation of BLEACH

Dataset		**MI**
AuCu	Energy	1.83
	Charge	2.36
MoS_2	Energy	1.40



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.8: Plots of (a, b, and c) parity, (d, e, and f) histogram-compare, and (g, h, and i) density-based accuracy for learning energies (left column) and charges (middle column) on AuCu, and energies (right column) on MoS_2 nano-cluster database, using LMBTR

4.4 LMBTR

In our implementation LMBTR, K_2 and K_3 are considered. Both of these K's have count (of number of grid points), decay factor, and sigma as hyper-parameters. These hyper-parameters are optimised with a Nelder-Mead algorithm[57] that maximises MI. This gives the values in table 4.7. Using these values, KRR is trained on LMBTR values.

The MAE, with this optimisation, is found at 0.05 eV, 1.7 me, and 0.42 eV for AuCu energies and charges, and MoS_2 energies, respectively. This is also consistent with our MI values, cf. table 4.8. These values are also close to MAE for MBTR for same clusters[34].

Table 4.7: Optimised hyper-parameter values for LMBTR

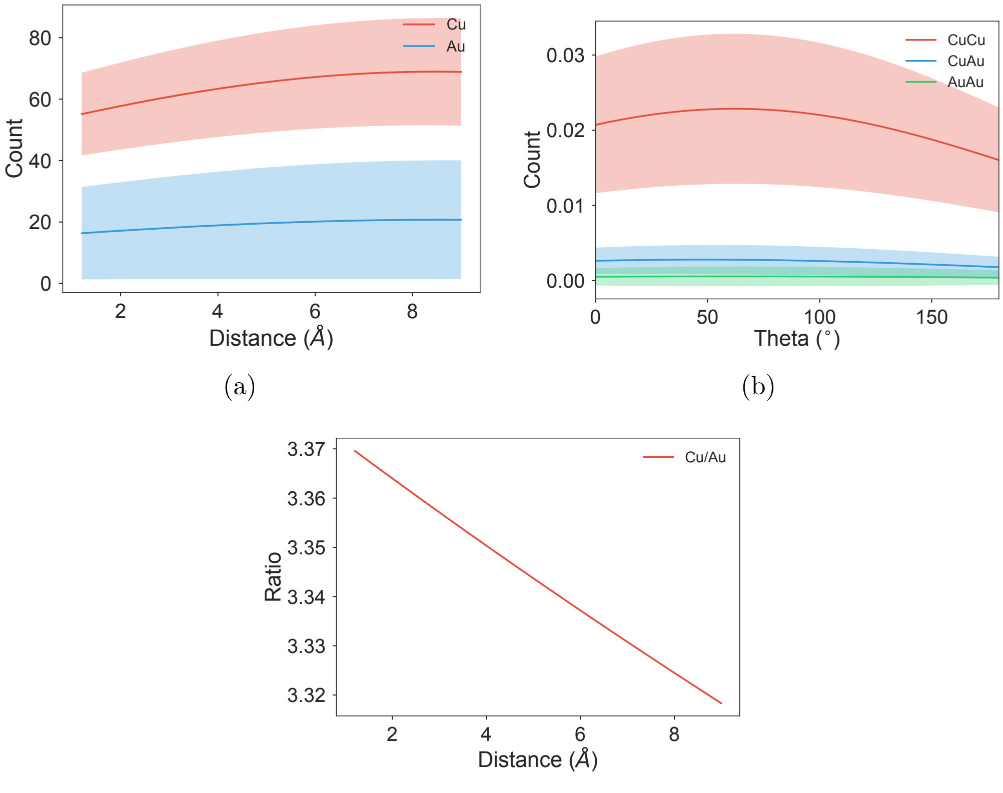
Dataset	Hyper-parameter		Value
AuCu	K_2	decay factor	2.5
		σ	0.1
		count	150
	K_3	decay factor	1
		σ	1
		count	40
MoS_2	K_2	decay factor	1.6
		σ	1.1
		count	150
	K_3	decay factor	0.5
		σ	0.25
		count	40

The plots for MAE, in AuCu database are in fig 4.8. The parity plots are linear, with some slight deviation at higher energies. The model learns the two clusters of hydrogen position, as seen in histogram-compare plots. However, in the MoS_2 case, the points are more scattered. A trend similar to previous cases is seen, where the model mis-predicts at higher energies.

Besides training, LMBTR can also trace real space, from observable space. This bi-directional operabilty is exploited to find correlations in atom positions in real space with adsorption energies. For this thesis, AuCu-895 nano-cluster, with geometrically relaxed hydrogen is used. The geometry

Table 4.8: MI values for LMBTR

Dataset		MI
AuCu	Energy	2.12
	Charge	2.93
MoS_2	Energy	1.96



(a)

(b)

(c)

Figure 4.9: Plots of (a) distances, (b) cosine of angles, with standard deviation regions, and (c) Au/Cu ratio for bi-directional operability in energy between -0.1, 0.07 on AuCu-895 nano-cluster database, using LMBTR

relaxation for hydrogen position represents real cases more accurately. Also, when LMBTR values are plotted for certain energies, the noise from unrelaxed geometries — that skew the dataset, since they are least probable in real scenarios— is avoided. This plot of LMBTR for AuCu-895 nano-cluster, in fig 4.9, clearly imply that Cu should be higher concentration on surface, for a catalytic site, which catalysis one hydrogen atom; although, a lower density of Au adds to catalysis. Also, ratio of Cu to Au doesn't change through the nano-cluster. It is important to note these graphs are from the perspective of hydrogen adsorption site, that fit our criteria. So all ratios come from a point outside the nano-cluster.

This bi-directional operabilty is further exemplified when PCA of LMBTR is performed. It is found that a majority of information of LMBTR is conveyed in four principal components, in the AuCu dataset. This is further

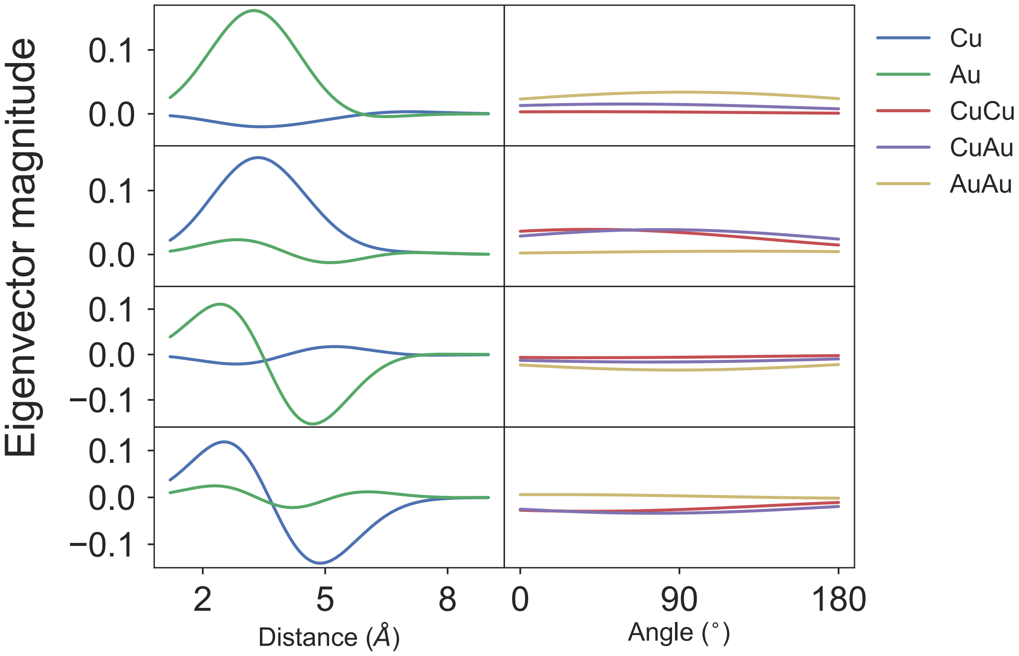


Figure 4.10: Plot of eigenvector values (or weights) for distances (left column) and angles (right column) corresponding to the four principal components. These principal components are stacked row-vise with decreasing eigenvalues.

confirmed by KRR machine-learning. The KRR model gives equal MAE for the four principal components as with the four hundred and twenty LMBTR features. The weights of linear combinations, that make these components, reveal further correlations. In fig 4.10, these weights are plotted for the four principal components and following observations are made.

- First principal component (cf. fig 4.10(first row))
  This component has higher coefficients for Au. This dependence on Au distance is maximum between 2–5 Å. There is a slight negative dependence on distances in Cu distribution, which is maximum between 3–4 Å. The dependence on angles is weak, although, a spread of weights on Au-Au angles is visible.

- Second principal component (cf. fig 4.10(second row))
  Similar to the first component, the second component's dependence on distance is maximum between 2–5 Å, although for Cu distribution. Also, a weak dependence on Cu-Cu and and Cu-Au angles is seen, more toward acute angles. Further, a weak dependence on distances of Au

distribution is seen. This dependence is positive with Au around 3 Å and negative around 5 Å.

- Third principal component (cf. fig 4.10(third row))
  The third component also has higher coefficients for Au. Although, the dependence on atoms around 2–3 Å is positive, while atoms around 4–5 Å is negative. An opposite dependence of Cu atoms is seen, around 3 Å, and 5Å. A weak dependence of Au-Au angles is also seen.

- Fourth principal component (cf. fig 4.10(fourth row))
  Similar to third component, this component has a higher coefficient for distances in Cu distribution. A positive dependence around 2–3 Å and negative around 4–5 Å is observed. However, a slight negative dependence of acute angles of Cu-Cu and Cu-Au is seen. Moreover, a positive dependence of Au atoms, around 2 and 6 Å, is also seen.

These dependences, in making the four principal components, imply an interdependence of the spectrum for Cu and Au. This brings out the inherent correlations in our AuCu dataset.

Chapter 5

Discussion

After a thorough development of methods and their implementation, this chapter discusses few meaningful findings, with attached outlook.

5.1 SOAP based on charge density

The implementation based on interpolations didn't show any "learning", even for very-fine grids. This casts serious doubts on interpolation based approaches in this scenario.

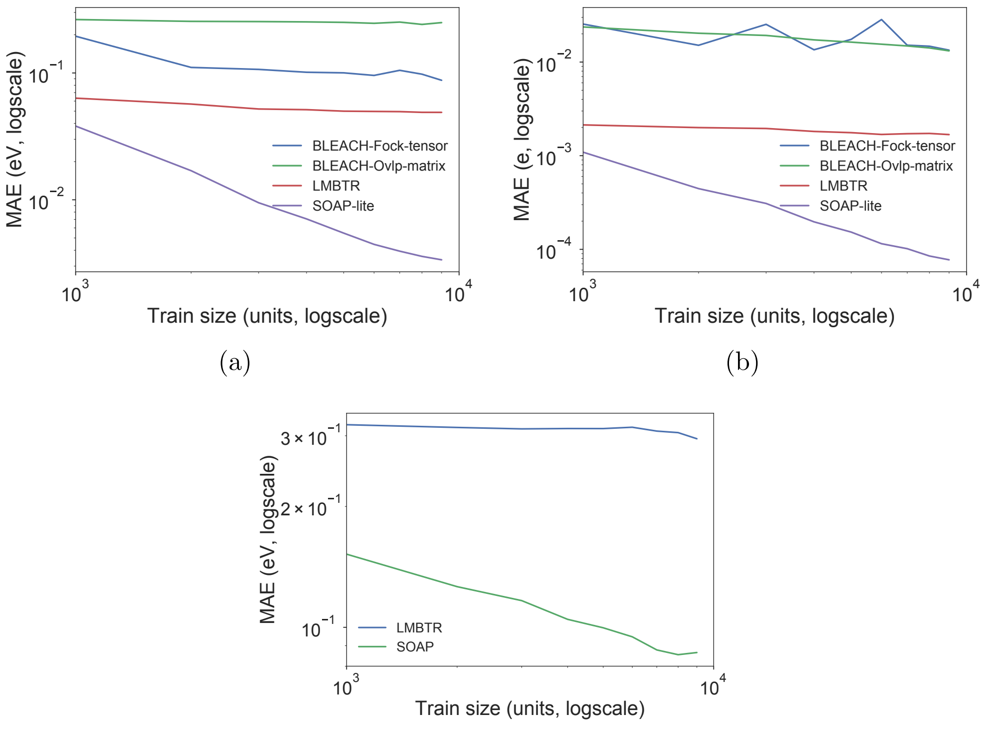
Moreover, the charge density calculation derived from density matrix is only as good as the implementation of SOAP which treats all elements the same. This implies, the charge density approach cannot, in itself, differentiate between atoms of different elements.

5.2 BLEACH

The overlap matrix not only performs better than coulomb repulsion, it is also much faster to calculate. Calculating 2-center 2-electron matrix is computationally heavy, and isn't implemented for parallel execution in PYSCF. Although, the overlap matrix method gave an accuracy of 0.25 eV on AuCu single cluster, it fails on MoS_2 multi-cluster. This is also seen with BLEACH Fock tensor approach.

In the BLEACH Fock tensor approach RNN, with padded features, is chosen over direct KRR approach because the accuracy of predictions are the same, and the RNN can make simple directed graphs and overfit the dataset. This implies a better learning is possible with bigger dataset.

Also, the select-MO derived density-matrix application doesn't improve on accuracy. This indicates that density matrix — and methods that rely on



(a) (b)

(c)

Figure 5.1: Learning curves for nano-cluster database of AuCu: (a) energy and (b) charge, and MoS_2: (c) energy

it — already incorporates sensitivity to orbitals at required levels.

5.3 LMBTR

LMBTR shows promise for it is simple and functional. It achieves a MAE below 0.1 eV for energy prediction in AuCu database; which is equal to that of MBTR dataset[34]. This implies LMBTR doesn't lose information in the shift from global to local scope. It reduces the number of features, from twelve thousand two hundred and sixty to four hundred and twenty, drastically with same hyper-parameters (cf. table 4.7). LMBTR use is, thus, reasonable since we employ local observable and it eases machine learning.

Further, the bi-directional operability of LMBTR is of a major significance, when compared to other descriptors. This reveals insights that can be extended into real space. In the AuCu-895 dataset, it is found that high

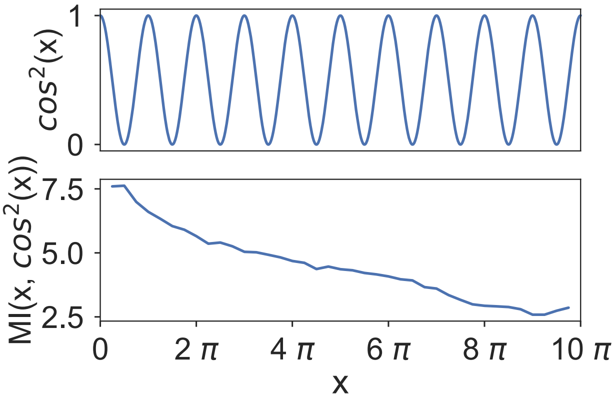


Figure 5.2: A comparison of function $\cos^2$(x) (top) with MI with increasing range of x (bottom)

concentrations of Cu is important, while a small concentration of Au is necessary too. An approximate ratio of Cu to Au of 3.3 is found appropriate for single hydrogen adsorption energy between -0.1–0 eV. Whereas, in the case of AuCu dataset, it is found that relevant information can be expressed as linear combination of four principal components. These linear combinations further reveal the principal correlations in variations of LMBTR features. It is interesting to note that these principal components transform the data into vectors that primarily rely on distributions of a certain element; and that the effect of angles is less pronounced.

5.4 Learning curves

Learning curves are made for descriptors discussed, with optimised hyperparameters, and compared with SOAP-lite. This includes: BLEACH Fock, BLEACH overlap matrix, and LMBTR for energy and charge prediction on AuCu and LMBTR on MoS_2.

It is evident that none of the descriptors proposed performs better than SOAP-lite. Although, LMBTR's accuracy comes within 0.1 eV for AuCu, it doesn't perform well for MoS_2. Moreover, BLEACH fock matrix performs better than overlap matrix on AuCu, but similar on MoS_2, as also indicated by MI values.

Table 5.1: The table summarizes MI values calculated throughout the thesis. The MI values which do not correlate with learning errors are in bold.

Descriptor	**Dataset**		**MI**	**MAE**
SOAP-lite	AuCu	Energy	**1.52**	3 meV
		Charge	**1.81**	0.7 μ e
	MoS_2	Energy	**1.42**	0.1 eV
BLEACH Overlap matrix	AuCu	Energy	1.53	0.25 eV
		Charge	1.94	13 me
	MoS_2	Energy	1.40	0.42 eV
BLEACH Fock tensor	AuCu	Energy	1.83	87 meV
		Charge	2.36	7.7 me
	MoS_2	Energy	1.40	0.48 eV
LMBTR	AuCu	Energy	2.12	48 meV
		Charge	2.93	1.7 me
	MoS_2	Energy	1.96	0.3 eV

5.5 Error analysis

For evaluating a dataset, AAD is a vital tool to gauge the accuracy of machine-learning models. This is also confirmed by the data visualisation in chapter 4. Also, the parity plot, histogram-compare plots, and density-based accuracy plot gives a better perspective on out dataset and its learning.

To compare the MI, a summary of tables 2.1, 4.3, 4.5, and 4.8 is put in table 5.2. A correlation between MI values, and MAE for LMBTR, BLEACH Fock, and BLEACH overlap matrix is found. However, this correlation fails for SOAP-lite; here, the MI indicates lower information where the training accuracies are higher — as compared to other descriptors. This is explained by the nature of implementation as described by Krakov et. al.[40].

It is found that the implementation of MI is also dependent on injective-ity of the descriptor and the target spaces. It implies that the implementation projects MI reliably only for injective functions; however, its reliability decreases with decreasing injective-ity. To show this, MI, between x and $\cos^2$(x), is plotted against range of x, in fig 5.2. With an increase in range multiple x start to give same value of $\cos^2$(x), due to its periodicity. This signifies a decrease in injective-ity, while the information is same. It is seen that MI value stays at maximum value till $\pi/2$ — i.e. until $\cos^2$(x) is injective; then it starts decreasing with increasing range of x. Due to this dependence of our implementation of MI, optimisations for LMBTR were redone while minimising MAE. The hyper-parameters obtained with this approach are found same as the ones determined while maximising MI. Similar tests are done for SOAP based on charge density, and BLEACH . The SOAP based on charge density showed no learning, with MAE equal to AAD. This is concurrent with our MI values, however, a direct correlation cannot be justified as MAE is at the threshold of AAD — which is as good as learning from random variables. Although, the implementations of BLEACH showed the concurrency. Thus, the correlation of MI with MAE for LMBTR, BLEACH is confirmed.

A significant implication of dependence of our implementation's MI over injective-ity is observed when compared to the error analysis. As errors in BLEACH and LMBTR are correlated to MI, these functions can be considered to be of an injective kind — not necessarily linear; whereas, SOAP-lite is not. While this explains the high accuracies in predicting the energies with low MI values for SOAP-lite, it also indicates the complexity in SOAP function that allows such accuracies.

Chapter 6

Conclusions

Earth-abundant catalyst design is a challenging task. It involves deep understanding of catalytic process. To accomplish this, theoretical techniques can be employed; but these are computationally heavy and time consuming. Here, machine-learning based on large databases is useful. This thesis designs and describes descriptors based on MBTR, and SOAP— vital for machine-learning applications.

In the case of MBTR, LMBTR is tested, which is proven to be an efficient, fast, and simple descriptor for predicting local observables. It doesn't lose any information over MBTR, and achieves equally accurate prediction.

While, in the case of SOAP, SOAP with charge density and BLEACH is tested. It is found that evaluation of SOAP with charge density, by numerical integrals over interpolation fails and by analytic integrals cannot distinguish between elements. Further, in implementation of BLEACH , the overlap matrix based approach does succeed but MAE isn't satisfactory. This is improved in Fock tensor approach, where a massive RNN-MLP neural network is implemented. It gives satisfactory result for simpler system (AuCu single cluster), however, fails for a more complex one (MoS_2 multi-cluster).

For error analysis, parity, histogram-compare, and density-based accuracy plots give vital understanding of the model prediction. This supplements the understanding by MAE. Further, AAD proves to be a robust quantity to gauge the inherent deviation in the dataset; which gives an appropriate scale for MAE.

The accuracy in predictions for newer methods didn't precede SOAP-lite. However, the relevance of LMBTR and BLEACH still stands. The backward operability of LMBTR is crucial in visualising the inherent patterns and correlations in the dataset; while BLEACH provides an element-agnostic approach to descriptor methods, which can readily be extended to diverse systems.

Chapter 7

Outlook

This thesis introduced new descriptors, which are modeled on SOAP and MBTR. These descriptors are rigorously optimised and compared for better model accuracies, and their possible application are discussed. These extensive insights reveal that further improvements in the same direction are possible.

It is evident that LMBTR can be intuitively modeled to learn from vastly different perspectives. It can be developed, for example, from surface-atom centric, nano-cluster core centric, or other novel perspective. In our study the LMBTR is developed from the perspective to hydrogen adsorption site. If we instead choose the perspective of nano-cluster, trends like hydrogen coverage, or stable atom-ratios can be pointed out. A modification in LMBTR can allow it do the same. However, such methods will require a larger database.

Further, the similarities in SOAP integration and BLEACH overlap matrix integrations are evident. The equation 2.28 can be understood as an integration of product of two spherical harmonics with different centers. The first spherical harmonic clearly is the hydrogen atom, with radial density and angular part. While the second spherical harmonics is the density of atoms, expressed as a gaussian. This harmonic is a s-orbital with the gaussian as radial part, and Y_{00} — which is a constant — as angular part. Hence, the SOAP integral is the overlap matrix, cf. eq 2.6. However, in our work, the overlap matrix does not give significant MAE. This is attributed to the manner in which the overlap matrix is utilised. While in BLEACH , it is simply summed for all atoms, irrespective of element-type, the SOAP-lite circumvents this by storing the product into a tensor of size n×n×l — which is later flattened, and redundant values removed — for individual element. Therefore, an implementation of BLEACH overlap matrix, which keeps coefficient separate, while not separating element type — as it breaks the purpose of BLEACH — can further improve accuracy.

Bibliography

[1] Bård Lindström and Lars J Pettersson. A brief history of catalysis. *Cattech*, 7(4):130–138, 2003.

[2] Max Appl. The haber-bosch process and the development of chemical engineering. *A Century of Chemical Engineering (ed. W. Furter), Plenum Publishing Corporation*, pages 20–51, 1982.

[3] Christos Comninellis. Electrocatalysis in the electrochemical conversion/combustion of organic pollutants for waste water treatment. *Electrochimica Acta*, 39(11-12):1857–1862, 1994.

[4] Gunter Alfke, Walther W Irion, and Otto S Neuwirth. Oil refining. *Ullmann's Encyclopedia of Industrial Chemistry*, 2007.

[5] Ch Elschenbroich. Organometallics wiley, 2006.

[6] Gerald Frenkel, David Lincoln Nelson, Brook Chase Soltvedt, and Albert Lester Lehninger. *Test Bank for Nelson and Cox, Lehninger Principles of Biochemistry.* Worth Publishers, 2000.

[7] A Lasia. Hydrogen evolution reaction. *Handbook of fuel cells*, 2, 2010.

[8] Yanguang Li, Hailiang Wang, Liming Xie, Yongye Liang, Guosong Hong, and Hongjie Dai. MoS_2 nanoparticles grown on graphene: an advanced catalyst for the hydrogen evolution reaction. *Journal of the American Chemical Society*, 133(19):7296–7299, 2011.

[9] Damien Voiry, Maryam Salehi, Rafael Silva, Takeshi Fujita, Mingwei Chen, Tewodros Asefa, Vivek B Shenoy, Goki Eda, and Manish Chhowalla. Conducting MoS_2 nanosheets as catalysts for hydrogen evolution reaction. *Nano letters*, 13(12):6222–6227, 2013.

[10] Eric J Popczun, James R McKone, Carlos G Read, Adam J Biacchi, Alex M Wiltrout, Nathan S Lewis, and Raymond E Schaak. Nanostructured nickel phosphide as an electrocatalyst for the hydrogen evolution

reaction. *Journal of the American Chemical Society*, 135(25):9267–9270, 2013.

[11] Yanmei Shi and Bin Zhang. Recent advances in transition metal phosphide nanomaterials: synthesis and applications in hydrogen evolution reaction. *Chemical Society Reviews*, 45(6):1529–1541, 2016.

[12] Rasmus Kronberg, Mikko Hakala, Nico Holmberg, and Kari Laasonen. Hydrogen adsorption on MoS_2-surfaces: a dft study on preferential sites and the effect of sulfur and hydrogen coverage. *Physical Chemistry Chemical Physics*, 19(24):16231–16241, 2017.

[13] Hong Li, Charlie Tsai, Ai Leen Koh, Lili Cai, Alex W Contryman, Alex H Fragapane, Jiheng Zhao, Hyun Soo Han, Hari C Manoharan, Frank Abild-Pedersen, et al. Activating and optimizing MoS_2 basal planes for hydrogen evolution through the formation of strained sulphur vacancies. *Nature materials*, 15(1):48, 2016.

[14] Jakob Kibsgaard, Charlie Tsai, Karen Chan, Jesse D Benck, Jens K Nørskov, Frank Abild-Pedersen, and Thomas F Jaramillo. Designing an improved transition metal phosphide catalyst for hydrogen evolution using experimental and theoretical trends. *Energy & Environmental Science*, 8(10):3022–3029, 2015.

[15] Di-Yan Wang, Ming Gong, Hung-Lung Chou, Chun-Jern Pan, Hsin-An Chen, Yingpeng Wu, Meng-Chang Lin, Mingyun Guan, Jiang Yang, Chun-Wei Chen, et al. Highly active and stable hybrid catalyst of cobalt-doped FeS_2 nanosheets–carbon nanotubes for hydrogen evolution reaction. *Journal of the American Chemical Society*, 137(4):1587–1592, 2015.

[16] Yao Zheng, Yan Jiao, Mietek Jaroniec, and Shi Zhang Qiao. Advancing the electrochemistry of the hydrogen-evolution reaction through combining experiment and theory. *Angewandte Chemie International Edition*, 54(1):52–65, 2015.

[17] H2020-NMP-2015-two-stage. Towards replacement of critical catalyst materials by improved nanoparticle control and rational design. webpage, 2016. `http://www.critcat.eu`.

[18] Jeff Greeley, Thomas F Jaramillo, Jacob Bonde, IB Chorkendorff, and Jens K Nørskov. Computational high-throughput screening of electrocatalytic materials for hydrogen evolution. *Nature materials*, 5(11):909, 2006.

[19] Walter Kohn, Axel D. Becke, and Robert G. Parr. Density Functional Theory of electronic structure. *The Journal of Physical Chemistry*, 100(31):12974–12980, 1996.

[20] Carlos Fiolhais, Fernando Nogueira, and Miguel AL Marques. *A primer in density functional theory*, volume 620. Springer Science & Business Media, 2003.

[21] Anubhav Jain, Geoffroy Hautier, Charles J Moore, Shyue Ping Ong, Christopher C Fischer, Tim Mueller, Kristin A Persson, and Gerbrand Ceder. A high-throughput infrastructure for density functional theory calculations. *Computational Materials Science*, 50(8):2295–2310, 2011.

[22] Giovanni Pizzi, Andrea Cepellotti, Riccardo Sabatini, Nicola Marzari, and Boris Kozinsky. Aiida: automated interactive infrastructure and database for computational science. *Computational Materials Science*, 111:218–230, 2016.

[23] M Scheffler and C Draxl. Computer Center of the Max-Planck Society, Garching, The NoMaD Repository, 2014.

[24] Luca M Ghiringhelli, Jan Vybiral, Sergey V Levchenko, Claudia Draxl, and Matthias Scheffler. Big data of materials science: critical role of the descriptor. *Physical review letters*, 114(10):105503, 2015.

[25] Dimitris K. Agrafiotis, Walter Cedeño, and Victor S. Lobanov. On the use of neural network ensembles in qsar and qspr. *Journal of Chemical Information and Computer Sciences*, 42(4):903–911, 2002. PMID: 12132892.

[26] Jörg Behler. Perspective: Machine learning potentials for atomistic simulations. *The Journal of chemical physics*, 145(17):170901, 2016.

[27] Bin Jiang and Hua Guo. Permutation invariant polynomial neural network approach to fitting potential energy surfaces. *The Journal of chemical physics*, 139(5):054112, 2013.

[28] Tom M Mitchell et al. Machine learning. wcb, 1997.

[29] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[30] Katja Hansen, Franziska Biegler, Raghunathan Ramakrishnan, Wiktor Pronobis, O Anatole Von Lilienfeld, Klaus-Robert Müller, and Alexandre Tkatchenko. Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space. *The journal of physical chemistry letters*, 6(12):2326–2331, 2015.

[31] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5):058301, 2012.

[32] Haoyan Huo and Matthias Rupp. Unified representation for machine learning of molecules and crystals. *arXiv preprint arXiv:1704.06439*, 2017.

[33] Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18):184115, 2013.

[34] Marc Jäger, Eiakihonroeda Morooka, Filippo Canova, Lauri Himanen, and Adam Foster. Machine learning hydrogen adsorption on nanocluster through structural descriptors. *NPJ Computational Material*, 2018. Under review.

[35] Attila Szabo and Neil S Ostlund. *Modern quantum chemistry: introduction to advanced electronic structure theory.* Courier Corporation, 2012.

[36] M Born and R Oppenheimer. Quantum mechanics of molecular systems. *Ann. d. Phys*, 84:457, 1927.

[37] JÃ¼rg Hutter, Marcella Iannuzzi, Florian Schiffmann, and Joost VandeVondele. <span style=”font-variant:small-caps;”>cp2k:</span> atomistic simulations of condensed matter systems: <span style=”font-variant:small-caps;”>cp</span> 2 <span style=”font-variant:small-caps;”>k</span> Simulation Software. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 4(1):15–25, January 2014.

[38] Aleksei F Deon and Yulian A Menyaev. The complete set simulation of stochastic sequences without repeated and skipped elements. *J. UCS*, 22(8):1023–1047, 2016.

[39] SINGroup. Mutual Information codes. git repository, 2017. `https://github.com/fullmetalfelix/ML-CSC-tutorial`.

[40] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[41] Filippo Federici. Lectures on Machine learning in material simulation using descriptors. git repository, 2018. `https://github.com/fullmetalfelix/ML-CSC-tutorial`.

[42] SIN Group, Aalto. DESCRIBE is a python package for creating machine learning descriptors for atomistic systems. git repository, 2018. `https://github.com/SINGROUP/describe`.

[43] BJKN Hammer and JK Nørskov. Electronic factors determining the reactivity of metal surfaces. *Surface Science*, 343(3):211–220, 1995.

[44] Conrad Sanderson and Ryan Curtin. Armadillo: a template-based c++ library for linear algebra. *Journal of Open Source Software*, 2016.

[45] Python Software Foundation. The Python Package Index (PyPI) is a repository of software for the Python programming language. webpage, 2018. `https://pypi.org/`.

[46] NumFocus. Package for scientific computing with Python. webpage, 2018. `http://www.numpy.org/`.

[47] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed ¡today¿].

[48] Qiming Sun, Timothy C. Berkelbach, Nick S. Blunt, George H. Booth, Sheng Guo, Zhendong Li, Junzi Liu, James D. McClain, Elvira R. Sayfutyarova, Sandeep Sharma, Sebastian Wouters, and Garnet Chan. Pyscf: the python-based simulations of chemistry framework, 2017.

[49] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[50] François Chollet et al. Keras. `https://keras.io`, 2015.

[51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[52] Michael Waskom, Olga Botvinnik, Paul Hobson, John B. Cole, Yaroslav Halchenko, Stephan Hoyer, Alistair Miles, Tom Augspurger, Tal Yarkoni, Tobias Megies, Luis Pedro Coelho, Daniel Wehner, cynddl, Erik Ziegler, diego0020, Yury V. Zaytsev, Travis Hoppe, Skipper Seabold, Phillip Cloud, Miikka Koskinen, Kyle Meyer, Adel Qalieh, and Dan Allan. seaborn: v0.5.0 (november 2014), November 2014.

[53] Linus Torvalds. Version control system, 2005. `git-scm.com`.

[54] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[55] Joost VandeVondele and Juerg Hutter. Gaussian basis sets for accurate calculations on molecular systems in gas and condensed phases. *The Journal of chemical physics*, 127(11):114105, 2007.

[56] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, Oct 1996.

[57] Michael JD Powell. On search directions for minimization algorithms. *Mathematical programming*, 4(1):193–201, 1973.