Quantum Machine Learning Applied to Chemical Reaction Space

Inauguraldissertation
zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von

Stefan Niklaus Heinen

2022

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät
auf Antrag von:

Prof. Dr. O. A. von Lilienfeld, Prof. Dr. M. Meuwly, Prof. Dr. J. Kästner

Basel, den 19. Oktober 2021

Dekan: Prof. Dr. Marcel Mayor

"I'm very pleased that the authors used the model to answer an actual scientific questions, something that is somewhat rare in the literature."

Reviewer 1

List of Publications

1. **Machine learning meets volcano plots: computational discovery of cross-coupling catalysts**[1]

   B. Meyer, B. Sawatlon, S. Heinen, O.A. von Lilienfeld, C. Corminboeuf; Chemical science, 9, (35), 7069-7077

2. **Machine learning the computational cost of quantum chemistry**[2]

   S. Heinen, M. Schwilk, G.F. von Rudorff, O.A. von Lilienfeld; Machine Learning: Science and Technology, 1, (2), 025002

3. **Thousands of reactants and transition states for competing E2 and S2 reactions**[3]

   G.F. von Rudorff, S. Heinen, M. Bragato, O.A. von Lilienfeld; Machine Learning: Science and Technology, 1, (4), 045026

4. **Toward the design of chemical reactions: Machine learning barriers of competing mechanisms in reactant space**[4]

   S. Heinen, G.F. von Rudorff, O.A. von Lilienfeld; The Journal of Chemical Physics, 155, (6), 064105

5. **Geometry Relaxation and Transition State Search throughout Chemical Compound Space with Quantum Machine Learning**

   S. Heinen, G.F. von Rudorff, O.A. von Lilienfeld, arXiv preprint arXiv:2205.02623

The first publication is not included in this thesis since it was a collaboration with the group of C. Corminboeuf from EPFL and I was only responsible for the machine learning part.

In the third publication (second author) I was responsible for the machine learning, parts of the data generation (providing input geometries, generating validation scripts, all the figures in the publication (except the work flow (Figure 5.2)), and I wrote the machine learning section, as well as running calculations on the local cluster.

The last paper is not yet published, but will be submitted soon.

Abstract

Stefan Niklaus HEINEN

Quantum Machine Learning Applied to Chemical Reaction Space

The scope of this thesis is the application of quantum machine learning (QML) methods to problems in quantum chemistry and chemical compound space, especially chemical reactions.

First, QML models were introduced to improve job scheduling of quantum chemistry tasks on small university and large super computing clusters. Using QML based wall time predictions to optimally distribute the workload on a cluster resulted in a significant reduction of the time to solution by up to 90% depending on the type of calculation studied: Ranging from single point calculations, over geometry optimizations, to transition state searches on a variety of levels of theory and basis sets.

The main focus of this thesis remains with the navigation through the chemical reaction space using QML models. To train and test these models large, consistent, and carefully evaluated data sets are required. While extensive data sets with experimental results are available, consistent quantum chemical data sets, especially for reactions, are rare in literature. Thus, a dataset for two competing text book reactions E2 and S_N2 was generated, reporting thousands of reactant complexes and transition states with different nucleophiles (- H^-, - F^-, - Cl^-, - Br^-), leaving groups (- F, - Cl, - Br), and functional groups (- H, - NO_2, - CN, - CH_3, - NH_2) on an ethane scaffold. The geometries were obtained on the MP2/6-311G(d) level of theory with subsequent DF-LCCSD/cc-pVTZ single point calculation.

However, limited by computational resources, the data set was incomplete. Therefore, reactant to barrier (R2B) machine learning models were introduced to support the data generation and complete the dateset by predicting ∼11'000 activation barriers solely using the reactant geometry as input. Using R2B predictions, design rules for chemical reaction channels were derived by constructing decision trees. Furthermore, Hammond's postulate was investigated, showing the limits for its application on reactants far away from the transition state, e.g. conformers.

Finally, the geometry relaxation and transition state search solely using machine learned energies and forces was investigated. Trained on 200 reactions, the QML model was able to find 300 transition states, reaching out of sample RMSD of 0.14Å and 0.4Å for reactant geometries and transition states, respectively. Although, relatively large RMSD for the geometries remain, the out of sample MAE of 26.06cm^{-1} for the transition state frequencies show a well described curvature of the transition state normal modes in agreement with the MP2 reference.

Acknowledgements

First, I want to thank Prof. Dr. O. A. von Lilienfeld who introduced me to science and toughed me these past five years how to become a scientist. Also, I'd like to thank Prof. Dr. M. Meuwly for being my second supervisor and the support I received during my PhD. I tank Prof. Dr. J. Kästner for being my external examiner, for grading my exam and being part of the "Prüfungskomitte" in my defense.

I'd like to thank Dr. M. Etter and Dr. G. F. von Rudorff for proof reading my thesis.

I thank Dr. A. Christensen for the help in my early years of my PhD and for introducing me to the field of machine learning and especially for writing the QML code, which made my entire work much easier! I'd like to thank Dr. G. F. von Rudorff again for all the support during the papers we published together.

I also want to thank the entire Chemspacelab (current and alumni) for the great five years I spent in this group, for the coffee breaks, the fun, the BBQ's and the Raclettes, especially the "harter Kern": Diana Thachieva, Krystel El Hage, Felix Faber, Marco Di Gennaro, and Sebastian Brickel. Also, all the people in the physical chemistry department which I didn't mentioned specifically, it was a great time in Basel.

I want to thank H.P. Lüthi for the help during my Master at ETHZ and also the direction he pointed my in academia. I also want to thank my friends in Wallis for a great time not only during "Fasntacht" (GM Caracas) but during the entire year, which was a welcoming distraction from the academic life.

Finally, I want to thank my parents for the unconditional support I received throughout every step in my life. If the reader would excuse my gaming analogy: If life would be a computer game, I played on the easiest difficulty thanks to my parents.

I'm truly thankful for the support I received from supervisors, friends, colleagues, and family.

Contents

List of Publications **v**

Abstract **vii**

Acknowledgements **ix**

1 Introduction **1**
1.1 Motivation 1
1.2 Overview 1

2 Quantum Chemistry **5**
2.1 Introduction 5
2.2 The Schrödinger Equation 6
2.3 Hartree Fock 7
2.4 Electron Correlation 9
2.4.1 Post Hartree-Fock methods 9
2.4.2 Density functional theory 12

3 Quantum Machine Learning **17**
3.1 Introduction 17
3.2 Kernel Ridge Regression 19
3.2.1 Representations 20
3.2.2 Training & Testing 21
Representation & data set split 21
Cross Validation 23
Learning Curves 24
3.3 Data sets 25

4 Machine learning the computational cost of quantum chemistry **27**
4.1 Abstract 27
4.2 Introduction 28
4.3 Data 29
4.3.1 Quantum Data Sets 30
4.3.2 Toy System 30
4.3.3 Quantum Chemistry Tasks 31
4.3.4 Timings, Code, and Hardware 32
4.4 Methods 33
4.4.1 Quantum Machine Learning 33
4.4.2 Application: Optimal Scheduling 34
Job Array and Job Steps 34
Scheduling Simulator 35
4.5 Results and Discussion 36
4.5.1 Toy System 36

4.5.2 Quantum Machine Learning . . . 37
Single Point (SP) Wall Times . . . 37
Geometry Optimization (GO) Wall Times . . . 38
Transition State (TS) Wall Times . . . 40
Timings, Code, Hardware . . . 41
Single Point (SP) FLOPs . . . 41
4.5.3 Application: Optimal Scheduling . . . 41
Job Array and Job Steps . . . 41
Geometry Optimization Steps . . . 43
4.6 Conclusion . . . 44

5 Quantum Chemistry: E2 vs. S_N2 47
5.1 abstract . . . 47
5.2 Introduction . . . 48
5.3 Methods and Computational Details . . . 49
5.3.1 Machine Learning . . . 50
5.3.2 Reactants and Products . . . 50
5.3.3 Transition States . . . 53
5.4 Results . . . 55
5.4.1 Data . . . 55
5.4.2 Geometries . . . 55
5.4.3 Energies . . . 59
5.5 Conclusion . . . 62

6 Quantum Machine learning: E2 vs. S_N2 63
6.1 Abstract . . . 63
6.2 Introduction . . . 64
6.3 Methods . . . 65
6.3.1 Kernel Ridge Regression . . . 65
6.3.2 Representations . . . 66
6.3.3 Training & Testing: Learning curves . . . 66
6.3.4 Data & Scripts . . . 67
6.4 Results and discussion . . . 68
6.4.1 Learning Barriers . . . 68
6.4.2 New barrier estimates . . . 71
6.4.3 Design rule extraction . . . 73
6.4.4 Estimates of reactant and transition state geometries . . . 74
6.4.5 Hammond's postulate . . . 75
6.5 Conclusion . . . 76

7 Geometry Relaxation and Transition State Search 79
7.1 Abstract . . . 79
7.2 Introduction . . . 80
7.3 Methods . . . 82
7.4 Results and Discussion . . . 83

8 Conclusion 89

9 Appendix A 91
9.1 Atomic Units[22] . . . 91
9.2 Time independent Schrödinger equation . . . 91
9.3 Born Opennheimer approximation . . . 92

10 Appendix B **93**
10.1 Additional details on the data sets . . . 93
10.1.1 $\text{QMrxn}^{\text{GO}}_{\text{MP2}}$ . . . 93
10.1.2 $\text{QMrxn}^{\text{TS}}_{\text{MP2}}$ . . . 93
10.1.3 $\text{QM9}^{\text{GO}}_{\text{B3LYP}}$ . . . 94
10.1.4 $\text{QMspin}^{\text{SP}}_{\text{MRCI}}$ and $\text{QMspin}^{\text{GO}}_{\text{CASSCF}}$ . . . 94
10.2 First fit decreasing algorithm . . . 94
10.3 Learning curves . . . 95

11 Appendix C **97**
11.1 Learning Barriers . . . 97
11.2 Learning Geometries . . . 97

12 Appendix D **105**
12.1 Data Sets . . . 105
12.1.1 Constitutional Isomers . . . 105
12.1.2 Nucleophilic Substitution Reaction (S_N2) . . . 105
12.2 Model Training . . . 105
12.2.1 Constitutional Isomers . . . 105
12.2.2 Reactant Complexes . . . 106
12.2.3 Transition States . . . 106
12.3 Optimization . . . 106
12.3.1 Geometry Relaxation . . . 106
12.3.2 Transition State Search . . . 107
12.4 Validation . . . 107
12.4.1 Constitutional Isomers . . . 107
12.4.2 Reactant Complexes . . . 107
12.4.3 Transition States . . . 107
12.4.4 Success Rate . . . 107
12.4.5 RMSD's . . . 109
12.4.6 Frequencies . . . 109
12.4.7 Activation Energies . . . 109

Bibliography **111**

List of Figures

1.1 Development of computational resources and machine learning then and now. left: Merkel *et. al.*[5] in 1988 vs. right: in the year 2020. a) fictitious figure of a transition state in[5], b) Scheme of an activation barrier, c) Picture from Chapter 5 showing 2'466 transition state geometries projected into the xy plane, and d) summary of 7'500 differences in activation energies (Chapter 6) 2

1.2 Schematic overview of the Machine Learning work flow, starting from the left 1) High performance Computing and 2) Data Set Generation, towards the middle with Quantum Machine Learning, which takes geometries as input and returns Energies (top) and Energies and forces (bottom), to the right 3) Learning activation barriers from reactant geometries only and 4) Geometry optimization and transition state search using machine learned energies and forces.. 4

2.1 Flow chart showing the working principle of the self consistent filed method to solve the Roothaan-Haal equation. 9

2.2 Jacob's ladder showing different approximations for the exchange correlation functional. First step (LDA) using only the density $\rho(\mathbf{r})$, second step (GGA) using the density $\rho(\mathbf{r})$ and gradients $\nabla\rho(\mathbf{r})$, third step using the density $\rho(\mathbf{r})$, the gradients $\nabla\rho(\mathbf{r})$, and second derivatives $\nabla^2\rho(\mathbf{r})$ or kinetic energy density $\tau_{\alpha/\beta}$, the fourth step adds exact HF exchange, and the last steps adds virtual KS orbitals using a MP2 approach. 15

3.1 Illustration of a perceptron with input layers $\{x^1, ..., x^4\}$ (blue rectangles), neurons (gray circles), and output layer y (red circle). 17

3.2 Scheme showing how the data set is split into training and test set. Training set is used to optimize hyperparameters σ and λ using k-fold cross validation where the training set is split again into training and validations. 22

3.3 Scheme of working principle of k-fold cross validation for one set of hyperparameters σ and λ. 23

3.4 Results of the 5-fold cross validation shown as a heat map with encoded MAE in the color map for FCHL19. 24

3.5 Fictitious learning curves showing basic principles of good/bad performing models, as well as basic principles of learning curves (left) and learning curves for the constitutional isomers of QM9 (right) for three representations. 25

4.1 Compute resource growth of 500 fastest public supercomputers.[8] Estimated use by chemistry and materials sciences corresponds to 35%, corresponding to 2017 usage on Swiss National Supercomputing Center.[74]. 30

4.2 Wall time distribution of all tasks using kernel density estimation. . . . 32

4.3 Wall to CPU time ratio (using kernel density estimation) for Molpro calculations to identify runs with high wall time overhead due to heavy I/O load on clusters. 35
4.4 2D non-linear toy systems consisting of the Rosenbrock ("Rosen") and Himmelblau ("Him") functions and minimum search with three optimizers (Nelder-Mead (NM), BFGS, and Newton-CG (N-CG)). a) Learning curves showing the prediction error of KRR for Rosen (solid lines) and Him (dashed lines) function using starting point (x, y) as representation input. b) Top row shows the function values for Rosen (left) and Him (right). Row two, three, and four show the number of optimization steps (encoded in the heat map) for 10200 starting points for NM, BFGS, and N-CG, respectively. c) Row two, three, and four show the relative prediction error of the ML model trained on the largest training set size $N = 3200$ for NM, BFGS, and N-CG, respectively. 36
4.5 Learning curves showing normalized test errors (cross validated MAE divided by median of test set) for seven tasks using BoB (solid) and FCHL (dashed) representations. The model was trained on wall times normalized w.r.t. number of electrons. Horizontal lines correspond to the performance estimating all calculations have mean run time (standard deviation divided by mean wall time of the task). 37
4.6 PCA plots of kernel elements for BoB (left) and FCHL (right) for data set $\mathbf{QMspin}^{\mathbf{SP}}_{\mathbf{MRCI}}$. The weights of the two first principal components for the molecules in the data sets are plotted against each other and corresponding wall times are encoded as a heat map. Insets show the first 100 eigenvalues on a log scale. 39
4.7 Learning curves showing normalized test errors (cross validated MAE divided by median of test set) for the first two geometry optimization steps on task $\mathbf{QMspin}^{\mathbf{GO}}_{\mathbf{CASSCF}}$ using BoB and FCHL as representations. The model was trained on CPU times divided by the number of electrons. Horizontal lines correspond to the performance estimating all calculations have mean run time (standard deviation divided by the mean wall time of the data set). 40
4.8 Learning curves showing normalized prediction errors (cross validated MAE divided by median of test set) for FLOP count and wall times on task $\mathbf{QM9}^{\mathbf{SP}}_{\mathbf{CC/DZ}}$ using BoB and FCHL representations. 42
4.9 Scheduling efficiencies for the seven different tasks (columns) assuming a certain per-job wall time limit specified in column title. Infrastructure assumptions correspond to either a large (solid lines, L) compute center or a small (dashed lines, S) university compute center. Top row reports CPU time overhead reduction when using the QML based (blue) rather than the conventional (green, orange) packing. Results are given relative to the total CPU time needed for the calculations of each data set for established methods (job array and jobs steps, see text) and our suggested method (QML). Bottom row shows actual vs. predicted times (using FCHL as representation) for all calculations in each data set using maximum training set size. 43

4.10 CPU time overhead and wall time for geometry optimizations compared between the simple approach and the QML approach. See text for details of the strategies. CPU time overhead given in percent relative to the bare minimum of CPU time needed. Wall time given relative to the wall time resulting from using the QML approach. All geometry optimizations come from task $\mathbf{QM9^{GO}_{CASSCF}}$. 44

5.1 Energy diagram of the competing E2 and S_N2 reactions, exemplifying kinetic vs. thermodynamic control, respectively. Reactant conformers ($R_{S,E}$) are shared between the reactions, while transition states ($TS_{S/E}$), product conformers ($P_{S/E}$), reactant complexes ($R'_{S/E}$) ,and product complexes ($P'_{S/E}$) are specific to each reaction. For each reaction, the energy difference between transition state and reactant complex is the activation energy $E_a^{S/E}$. 51

5.2 Validation procedure for reactant geometries starting from a candidate reactant structure to the decisions whether to accept or discard that candidate geometry. 52

5.3 Illustration of validation procedures for generating E2 (top) and S_N2 (bottom) geometries. Normal mode requirements for transition states (left column) show concerted motions which are characteristic for the reaction in question (red arrows point towards product, blue arrows towards reactant). Bond cleavages tested for reactant complexes and product complexes are shown in the mid and right column, respectively. Blue perpendicular lines correspond to removal of Y^-, YH and X^- for E2 and X^- or Y^- for S_N2 leading to infinite separation as shown in Figure 5.1). Bond cleavage indicated by red perpendicular lines corresponds to product formation. 54

5.4 Distribution of substituents R*k*, leaving groups X, and nucleophiles Y for all activation energies in the data set. Top: distribution for E2, bottom: distribution for S_N2. 56

5.5 Overview scatter plot of atomic positions of scaffold carbon atoms and nucleophiles and leaving groups for all transition states for E2 (top row) and S_N2 (bottom). Transition state geometries have been translated such that C1 (top left) or C2 (top right and bottom) are in the origin. Additionally, they have been rotated such that all atoms shown with the exception of the hydrogen sites in the top left panel are planar. Coordinates of other atoms have been projected into the figure plane. 57

5.6 a) MP2 energies of constrained geometries. b) ΔE_a of constrained to unconstrained geometries as obtained from quantum chemistry calculations (training data for the ML models). Inset: Learning curves for the Δ-ML models (constrained to unconstrained energies) illustrating test errors (MAE in kcal/mol) vs training set size (N). c) ML shifted MP2 energies. d) LCCSD energies on unconstrained MP2 geometries. e) ΔE_a of MP2 to LCCSD energies as obtained from quantum chemistry calculations (training data for the ML models). Inset: Learning curves for the Δ-ML models (MP2 to LCCSD energies) illustrating test errors (MAE in kcal/mol) vs training set size (N). f) ML shifted LCCSD energies. 59

6.1 **Scheme for competing reactions E2 vs. S_N2.** Top row: Transition states E2 (**4**) and S_N2 (**5**). Middle row: Reactant and nucleophile at infinite separation (**1**). In gas phase the energy of the transition state often lies lower than the energy of the reactants at infinite separation[218]. Bottom row: Product geometries at infinite separation (**6** and **7**) and reactant complexes (**2** and **3**). Properties of interest for this work are activation energies E_a^E and E_a^S, reactants, reactant complexes, and transition states. Table shows substituents R, leaving groups X, and nucleophiles Y. 64

6.2 **Learning curves** Activation energy prediction errors (out-of-sample) as a function of training set size N for activation barriers (E_a) of E2 (left) and S_N2 (right) using reactant geometries as inputs only. Results are shown for four representations (BoB, SLATM, FCHL19, one-hot) used within KRR models. Training data reference level of theory corresponds to DF-LCCSD/cc-pVTZ//MP2/6-311G(d), and estimated MP2 error is denoted as a green dashed horizontal line. Insets: Reference vs. estimated activation barriers using one-hot-based predictions and R^2 values being 0.89 and 0.94 for E2 (left) and S_N2 (right), respectively. 70

6.3 **Kernel PCA of the training set.** Kernel PCA of one hot encoding colored by the energy difference of activation energies of the two reactions $\Delta E_a = E_a^E - E_a^S$. Inset: Eigenvalues of the kernel PCA. Clusters represent most frequent combinations of leaving groups X (green) and nucleophiles Y (black). 71

6.4 **Learning curves across clusters** test error (MAE) vs. training set size (N) excluding NO_2 from training at position R2 (spheres) and at positions R2 and R3 (diamonds) for both reactions E2 (left) and S_N2 (right). The test set only contains compounds with NO_2 at position R2 (spheres) or at positions R2 and R3 (diamonds). 71

6.5 **Completion of data set using predictions of R2B models** Differences in activation energies ($\Delta E_a = E_a^E - E_a^S$) for all 7,500 reactions (calculated and predicted). Every square stands for a combination of R1-4, X, and Y shown in Figure 1. Positive values denote compounds that undergo a S_N2 reaction and negative values lead towards an E2 reaction. 72

6.6 **Histogram of energy distribution of ΔE_a.** Differences in activation energies ($\Delta E_a = E_a^E - E_a^S$) of 529 overlapping training instances (blue), 11k predictions (orange) and all 7'500 reactions (green). Molecules of the three highest, respectively lowest barrier differences are shown as molecules. 73

6.7 **Decision tree using extracted rules and design guidelines.** Decision tree using the R2B estimated activation barriers to predict changes in barrier heights by starting at all reactions (first energy level on the left) and subsequently apply changes by substituting functional groups, leaving groups and nucleophiles with E2 as an example. Blue dotted lines refer to an accepted change meaning only compounds containing this substituents at the position are considered. Orange dotted lines refer to substitution declined, meaning all compounds except the decision are kept. Vertical lines on the right of energy levels denote the minimum first (lower limit), and the third (upper limit) quartile of a box plot over the energy range. Numbers above energy levels correspond to the number of compounds left after the decision. Lewis structures resemble the decision in question. 74

6.8 **Model evaluation of geometrical properties using learning curves.** Test errors (MAE) of distances $d_{\mathrm{x,y}}$, angles α and β and dihedrals θ for both reactions E2 (a) and S_N2 (b). Horizontal lines correspond to the null model which uses the mean value of the training set for predictions. Compounds (**2** and **3**) illustrate the learned properties of the E2 reaction (**2**) and the S_N2 reaction (**3**) for reactant complexes and transition states. 75

6.9 **Applicability of Hammond's postulate.**Frequency heat map of activation energies projected onto structural differences in distances (d_x and d_y) between reactant complex conformers and transition states for both reactions E2 (first two plots) and S_N2 (last two plots). For S_N2 a good linear correlation (R^2 are 0.8 and 0.65 for d_x and d_y, respectively) in agreement with Hammond's postulate can be observed, while for E2 only d_x shows good correlation ($R^2 = 0.86$) whereas d_y lacks correlation ($R^2 = 0.5$). 76

7.1 Schematic potential energy surfaces in chemical compound space. Arrows show the working principle of OQML based iterative structural optimization: Training on reaction profiles of different chemical systems (purple), the OQML model is able to interpolate forces and energies throughout chemical compound space enabling the relaxation of the reactant and the search of the transition state (orange). Input geometries (squares) are easily obtained, e.g. from universal force field predictions. 81

7.2 Learning curves for energies (top left) and atomic forces (bottom left). RMSD performance curves of geometry vs. training set size N (middle). Success rates of the optimization and transition state searches (right). Colours correspond to results based on three distinct datasets: constitutional isomers (QM9), reactant complexes (QMrxn20), and transition states (QMrxn20). Dashed green and dotted orange horizontal lines correspond to the success rate of the reference calculations for the TS searches and the geometry optimization, respectively 84

7.3 Example normal mode scan showing energy changes as a function of distortion along TS modes for the transition state of the S_N2 reaction of [H(CN)C-C(CH_3)(NH2)] with Cl and F as leaving group and nucleophile, respectively. Geometry of an MP2/6-311G(d) converged and validated TS was used and distorted along its normal modes. Subsequently, single point calculation (MP2), as well as ML predictions were plotted for the first 23 normal modes and their displacements. The x-axis describes the index of the distorted geometry and the y-axis describes the energy relative to the MP2 equilibrium geometry. Both energies, MP2 (blue) and ML (orange) are scaled by the equilibrium geometry (index 10). 86

10.1 Learning curves showing normalized test errors (cross validated MAE divided by median of test set) using BoB and FCHL as representations. The model was trained on CPU times. Horizontal lines correspond to the performance assuming all calculations have mean run time (standard deviation divided by the mean wall time of the data set. 95

10.2 Learning curves showing normalized test errors (cross validated MAE divided by median of test set) using BoB and FCHL as representations. The model was trained on normalized (by number of electrons) CPU times. Horizontal lines correspond to the performance assuming all calculations have mean run time (standard deviation divided by the mean wall time of the data set. 95

10.3 Learning curves showing normalized test errors (cross validated MAE divided by median of test set) using BoB and FCHL as representations. The model was trained on wall times. Horizontal lines correspond to the performance assuming all calculations have mean run time (standard deviation divided by the mean wall time of the data set. 96

10.4 Learning curves showing normalized test errors (cross validated MAE divided by median of test set) using BoB and FCHL as representations. The model was trained on normalized (number of occupied orbitals to the power of 2 times number of basis functions to the power of 4) wall times. Horizontal lines correspond to the performance assuming all calculations have mean run time (standard deviation divided by the mean wall time of the data set. 96

11.1 Hyperparameter optimization of R2B models: a) BoB, b) SLATM, c) FCHL19, and d) one-hot encoding for E2 (left) and S_N2 (right) reactions. 98

11.2 **Learning curves** Activation energy prediction errors (out-of-sample) as a function of training set size N for activation barriers (E_a) of E2 a) and S_N2 b) using reactant complex geometries (dashed lines) and reactants (solid lines) as inputs only. Results are shown for four representations (BoB, SLATM, FCHL19, one-hot) used within KRR models. Training data reference level of theory corresponds to DF-LCCSD/cc-pVTZ//MP2/6-311G(d), and estimated MP2 error is denoted as a green dashed horizontal line. 99

11.3 **Rogers-Tanimoto coefficients** Similarity check for the one hot encoding between the training and test molecules using the Rogers-Tanimoto coefficient. Two binary representations are equal and inverse if their coefficient is 0 and 1, respectively. 101

12.1 Example ethane scaffold of an S_N2 reactions (left) with leaving groups X, nucleophiles Y, and functional groups R1-4 (right). 106

12.2 Validation for transition states a) Angle of reaction center, b) minimal distance of 0.9Å between the atoms, c) movement of atoms in reaction center using the first normal mode, and d) movement of all other atoms and normal modes. 108

List of Tables

4.1 Seven tasks used in this work generated from three data sets (QM9, QMspin, QMrxn), using three use cases (SP, GO, TS) on different levels of theory and basis sets. 32
4.2 QML results (normalized prediction errors) for seven task and both representations (BoB and FCHL) for largest training set size (N_{max}). . . 37

5.1 Chemical space for our reaction database: substituents R, leaving groups X and the nucleophiles Y^-. Molecular skeleton is ethane, see also Figure 5.1. The letters refer to the labels in our data set files. 50

7.1 Summary of results for 300 out of sample test cases for geometry optimizations (left) and transition state searches (right) for OQML models and three DFT methods for comparison with MP2/6-311G(d) as reference. The table shows the difference in geometry (RMSD), in frequency (ν), and in activation energy (E_a) for each method. N corresponds to the training set size of the ML models (MP2/6-311G(d)) and to the data set size for parametrization of the DFT functionals. Geometry optimizations using the LBFGS algorithm from the ASE package were truncated after 50 iterations and the default thresh hold (*fmax* = 0.05 eV/Å) was used. The limit for the transition state search was the default iteration limit of 100 steps. Frequency values in parenthesis for the transition states are the errors of the first (imaginary frequency) 85

10.1 Data sets of calculations used in this work: Software used for calculations, number of cores used per calculation, and CPU types the calculation ran on as well as details of the ML hyperparameter. 93

11.1 Results from R2B models used to generate learning curves for the E2 reaction. 99
11.2 Results from R2B models used to generate learning curves for the S_N2 reaction. 100
11.3 Results from R2B models used to generate learning curves for the S_N2 reaction. Learning of the reactant complex and transition state geometries. 102
11.4 Results from R2B models used to generate learning curves for the E2 reaction. Learning of reactant complexes. 103
11.5 Results from R2B models used to generate learning curves for the E2 reaction. Learning of transition state geometries. 103

12.1 Success rates for geometry optimizations of ground states (GS) and transition state searches (TS) for the OQML model and the three DFT methods. 109

List of Abbreviations

QM	**Q**uantum **M**echanics
SDE	**S**chrö**d**inger **E**quation
SCF	**S**elf **C**onsistent **F**ield
HF	**H**artree **F**ock
MP2	**M**øller **P**lesset
CI	**C**onfiguration **I**nteraction
CC	**C**oupled **C**luster
DFT	**D**ensity **F**unctional **T**heory
SP	**S**ingle **P**oint
GO	**G**eometry **O**ptimization
TS	**T**ransition **S**tate
GS	**G**round **S**tate
FF	**F**orce **F**ield
ML	**M**achine **L**earning
QML	**Q**uantum **M**achine **L**earning
OQML	**O**perator **Q**uantum **M**achine **L**earning
KRR	**K**ernel **R**idge **R**egression
ANN	**A**rtificial **N**eural **N**etwork
CV	**C**ross **V**alidation
LC	**L**earning **C**urve
PCA	**P**rincipal **C**omponent **A**nalysis
MAE	**M**ean **A**bsolute **E**rror
RMSD	**R**oot **M**ean **S**quare **D**eviation
CM	**C**oulomb **M**atrix
BoB	**B**ag **o**f **B**onds

SLATM	**S**pectrum of **L**ondon and **A**xilrod **T**eller **M**uto
CPU	**C**entral **P**rocessing **U**nit
HPC	**H**igh **P**erformance **C**omputing
FLOPS	**F**loating **P**oint **O**perations per **S**econd

Dedicated to my parents whose unconditional support made my
path towards becoming a scientist possible.

Chapter 1

Introduction

1.1 Motivation

Computational chemistry, in addition to experimental work, allows for fast and deterministic results delivering insights in chemical reactions. As an example, Merkel *et al.* in 1988[5] studied the S_N2 reaction for a simple system ($H^- + CH_3F \rightarrow F^- + CH_4$) on MP2/6-311G** level of theory, giving insights in transition state geometries as well as rate constants.

With expanding compute resources (Moore's law[6]), more and larger systems could be studied. Even though, quantum chemical calculations allow for fast insights in chemistry, the more accurate methods are used, the more expensive the calculations will get. Hence, there is always a trade-off between accuracy and speed. Here, machine learning offers a solution, which keeps the accuracy of high level quantum chemical calculation but delivers the results (predictions) in milliseconds. Although, chemical space is immense[7], there are many redundancies in molecular structures. This fact is used by machine learning models, which infer data between molecules close in representation space, allowing to circumvent expensive quantum chemical calculations for an entire data set and deliver fast and accurate predictions throughout chemical compound space given a training set.

In 1988, the computational resources were 10^{-3} peta FLOPS (top 500 super computers[8]), which limited the study of reactions to a few dozens at most, as shown by Merkel *et. al.*[5]. Today these resources grew to 10^{3} peta FLOPS and allow for the generation of large, consistent data sets, studying the same reaction but adding multiple functional groups, different leaving, as well as attacking groups, and therefore expanding the chemical space on a similar level (MP2/6-311G(d)). Figure 1.1 sets the computational resources from the 1990's and the 2020's into contrast. Going from the analysis of few transition states (a), to thousands of transition state geometries (c), and from a few rate constants (b) towards 15'000 activation barriers (d) allowing for a much broader and more systematic insight into the chemical reaction space.

1.2 Overview

Up to now, the main focus in machine learning was to develop reaction predictors, artificial neural networks (ANN) using experimental but energy free data to predict the outcome of a reaction, or the optimal reaction path for a given product[9–15]. Only a few quantum reaction data sets are available[16,17], therefore the focus of this thesis was to generate a quantum chemical data set (Chapter 5), the use of machine

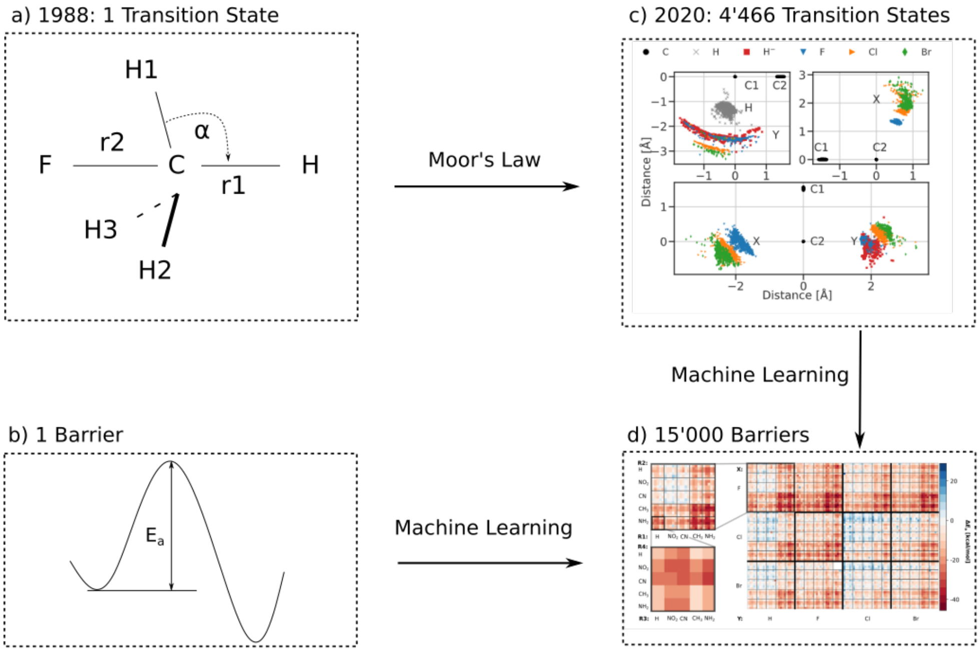


FIGURE 1.1: Development of computational resources and machine learning then and now. left: Merkel *et. al.*[5] in 1988 vs. right: in the year 2020. a) fictitious figure of a transition state in[5], b) Scheme of an activation barrier, c) Picture from Chapter 5 showing 2'466 transition state geometries projected into the xy plane, and d) summary of 7'500 differences in activation energies (Chapter 6)

learning models to support the data generation (Chapter 6), as well as the analysis of quantum chemical data sets, using machine learning techniques (Chapter 6 & 7).

This thesis walks the reader through the entire process of quantum machine learning. The workflow for quantum machine learning is shown in Figure 1.2 with the insets being representative equations of Chapters 2-4 and the introduction figures of Chapters 5-7.

This thesis starts with a brief introduction to quantum chemistry (**Chapter 2**), followed by an introduction to machine learning, in particular kernel ridge regression (KRR) in **Chapter 3**.

Having the tools for quantum chemical calculations and machine learning, **Chapters 4** and **5** deal with the generation of the data set. **Chapter 4** is devoted to the optimization of scheduling in high performance compute (HPC) centers for different quantum chemical methods on different levels of theory and basis sets. The generation of the E2 vs. S_N2 data set using quantum chemical calculations yielding thousands of reactant complexes and transition states is described in **Chapter 5**. Also, Δ-machine learning models[18] were applied to obtain reaction barriers on coupled cluster level of theory. Both reactions, E2 and S_N2, have a common reactant, an ethane scaffold which was substituted by various functional groups, leaving groups, and nucleophiles.

Chapter 6 introduces the reactant to barrier (R2B) machine learning model, which solely depends on the reactant geometry and after training, predicts activation barriers. Using the R2B model, the data set from Chapter 5 was completed with ∼11'000 activation barrier predictions and subsequently analysed. Using this data, supporting experimental reaction design was possible by constructing decision trees. Also, learning key geometrical parameters, Hammond's postulate was studied and its limitations for the E2 reaction were shown.

Chapter 7 addresses the geometry optimization and transition state search using the operator quantum machine learning approach[19,20]. Using the atomic simulation environment (ASE) LBFGS optimizer or gaussian09 QST2 optimizer with predicted forces and energies, allowed fast and accurate screening of the chemical reaction space of the constitutional isomers of QM9 and the S_N2 reaction, for geometry optimizations or transition state searches, respectively.

The supplementary information of chapter 4, 6, and 7 can be found in the Appendices B, C, and D, respectively.

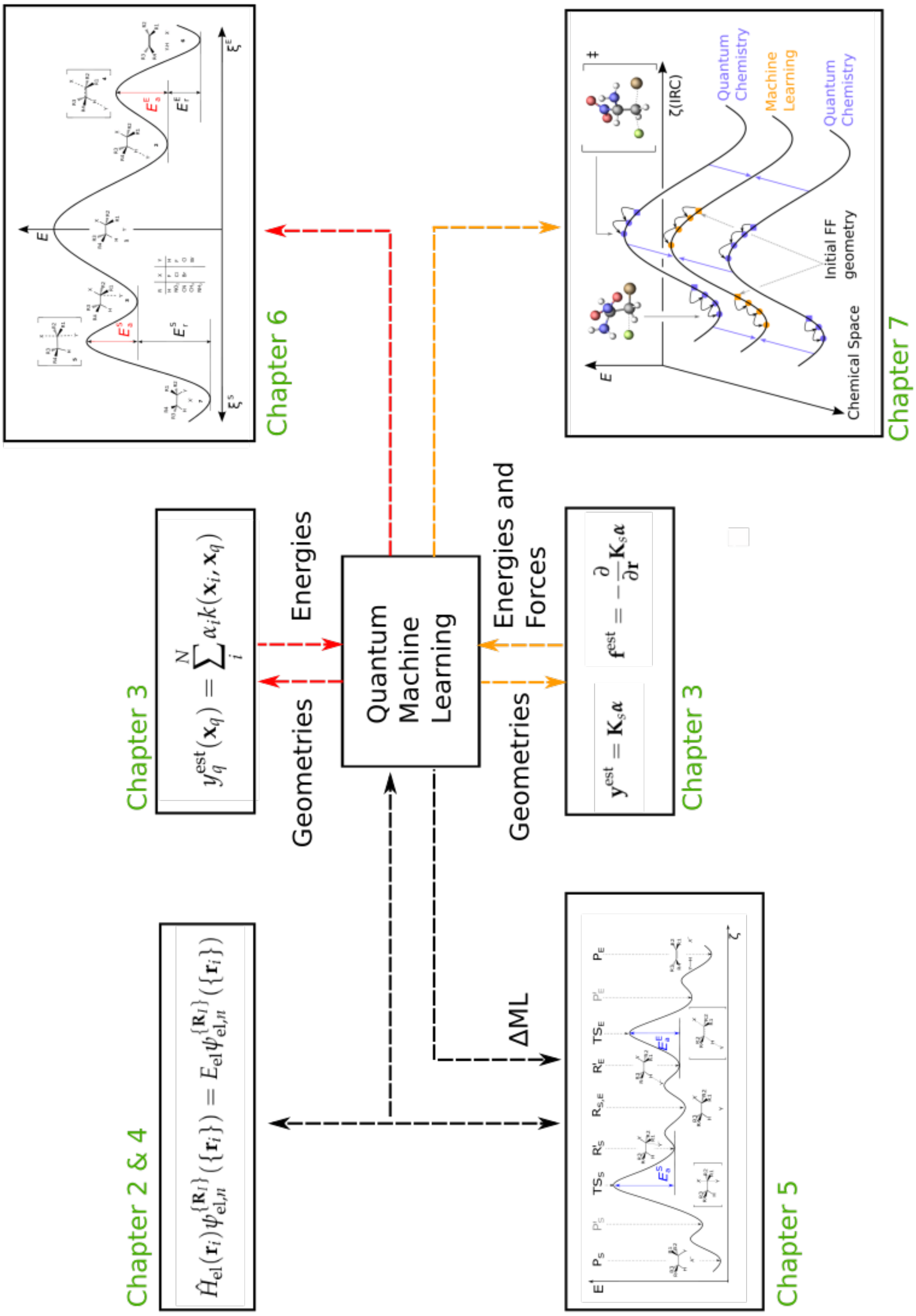

Chapter 2 & 4
$\hat{H}_{\text{el}}(\mathbf{r}_i)\psi^{\{\mathbf{R}_I\}}_{\text{el},n}(\{\mathbf{r}_i\}) = E_{\text{el}}\psi^{\{\mathbf{R}_I\}}_{\text{el},n}(\{\mathbf{r}_i\})$
Chapter 3
$y^{\text{est}}_q(\mathbf{x}_q) = \sum_i^N \alpha_i k(\mathbf{x}_i, \mathbf{x}_q)$
Geometries
Energies
Quantum Machine Learning
Chapter 6
Geometries
Energies and Forces
ΔML
Chapter 3
$\mathbf{y}^{\text{est}} = \mathbf{K}_s\boldsymbol{\alpha}$
$\mathbf{f}^{\text{est}} = -\frac{\partial}{\partial \mathbf{r}}\mathbf{K}_s\boldsymbol{\alpha}$
Chapter 5
Chapter 7
E
ξ
Chemical Space
Initial FF geometry
ζ(IRC)
Quantum Chemistry
Machine Learning
Quantum Chemistry
‡

FIGURE 1.2: Schematic overview of the Machine Learning work flow, starting from the left 1) High performance Computing and 2) Data Set Generation, towards the middle with Quantum Machine Learning, which takes geometries as input and returns Energies (top) and Energies and forces (bottom), to the right 3) Learning activation barriers from reactant geometries only and 4) Geometry optimization and transition state search using machine learned energies and forces..

Chapter 2

Quantum Chemistry

This chapter relies on the books of Levin[21], Szabo and Ostlund[22], and Steinhauser[23].

2.1 Introduction

The development of quantum mechanics began in the 1900 when the lower frequencies of the black body radiation were investigated and became experimentally available. This lead to the failing of classical mechanics, in particular Wien's formula[24], to describe the low frequency area in black body radiation:

$$I = \frac{a\nu^3}{e^{b\nu/T}} \tag{2.1}$$

where a an b are empirical constants and ν is the energy frequency. In the low frequency area the classical description of continuous energy levels failed. Max Planck proposed a new formula which also describes the low frequency area accurately:

$$I = \frac{a\nu^3}{e^{b\nu/T} - 1} \tag{2.2}$$

This lead Planck to believe that electromagnetic radiation can only be absorbed or emitted in discrete packets, meaning it is restricted to whole numbers. These whole numbers are multiples of $h\nu$ with $h \approx 6.626 \times 10^{-34}\,\mathrm{J} \cdot \mathrm{Hz}^{-1}$ being Planck's constant which he found by fitting the experimental black body curve. Thus the energy of each packet is **quantized**.

In 1905 Einstein proposed a solution to the problem of the photo electric effect in his "quantum paper"[25] by assuming the light being composed of particles (photons) with each photon having the energy:

$$E_{\text{photon}} = h\nu \tag{2.3}$$

This is in agreement with the observation that the kinetic energy of an emitted electron is independent of light's intensity but increases as the light's frequency increases. This behaviour was contradictory to what was expected from a wave like character in classical mechanics. The correct description of the photo electric effect later lead to the nobel prize in physics for Einstein.

These two applications of the quantization of the energy lead, in the early 1920s, to many contributions in the field of *quantum mechanics*, such as the work by Born "Zur Quantenmechanik"[26,27], by DeBroglie "Welle-Teilchen Dualismus"[28], in 1927

Heisenberg's uncertainty principle[29], and many more[30]. In 1926 Schrödinger introduced the wave mechanics and the famous Schrödinger equation (SDE)[31].

This chapter gives an introduction to the SDE and covers the different methods used in quantum chemistry to solve the SDE starting from the Hartree-Fock approach, over the post Hartree-Fock methods adding electron correlation (MP2, CC, CI), as well as the density functional theory (DFT).

2.2 The Schrödinger Equation

The Schrödinger equation is possibly the most important equation in quantum chemistry:

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},\mathbf{R},t) = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial \mathbf{r}^2} + -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial \mathbf{R}^2} + V(\mathbf{r},\mathbf{R},t)\right] = \hat{H}\Psi(\mathbf{r},t) \tag{2.4}$$

where $\Psi(\mathbf{r},\mathbf{R},t)$ is the wave function which depends on the position of the electrons $\mathbf{r}$, the positions of the nuclei $\mathbf{R}$, the time t and $\hat{H}$ is the Hamilton operator. In general, the time independent SDE is used in quantum chemistry. The wave function can be written as a product of two functions, one depending on the coordinates and the other depending on the time using following product Ansatz:

$$\Psi(\mathbf{r},\mathbf{R},t) = \psi(\mathbf{r},\mathbf{R}) \cdot f(t) \tag{2.5}$$

which leads to following expression of the time independent SDE (the derivation can be found in Appendix A):

$$\hat{H}\psi(\mathbf{r},\mathbf{R}) = E\psi(\mathbf{r},\mathbf{R}) \tag{2.6}$$

where $\hat{H}$ is the Hamilton operator:

$$\hat{H} = \underbrace{\sum_{i=1}^{n}\frac{1}{2}\nabla_i^2}_{\hat{T}_e} - \underbrace{\sum_{I=1}^{M}\frac{1}{2M_I}\nabla_I^2}_{\hat{T}_N} + \underbrace{\sum_{i<j}^{n}\frac{1}{\mathrm{r}_{ij}}}_{\hat{V}_{ee}} + \underbrace{\sum_{I<J}^{M}\frac{Z_I Z_J}{\mathrm{R}_{IJ}}}_{\hat{V}_{NN}} - \underbrace{\sum_{i,I}^{n,M}\frac{Z_I}{\mathrm{r}_{iI}}}_{\hat{V}_e N} \tag{2.7}$$

Here atomic units were used (derivation can be found in Appendix A), where i and I represent the electrons and nuclei, respectively. The number of electrons and nuclei are denoted as n and M. The first two terms are the kinetic operators of the electrons and the nuclei $\hat{T}_e$ and $\hat{T}_N$, respectively. The last three terms are the potential energy operators between the electrons $\hat{V}_{ee}$, the nuclei $\hat{V}_{NN}$, and between electrons and nuclei $\hat{V}_{eN}$. The Hamilton operator can thus be written in a more compact form:

$$\hat{H} = \hat{T}_e(\mathbf{r}) + \hat{T}_N(\mathbf{R}) + \hat{V}_{ee}(\mathbf{r}) + \hat{V}_{NN}(\mathbf{R}) + \hat{V}_{eN}(\mathbf{r},\mathbf{R}) \tag{2.8}$$

Since the nuclei are three orders of magnitude heavier than the electrons, the movement between these particles can be separated with the Born Oppenheimer approximation using the wave function product Ansatz:

$$\Psi(\{\mathbf{r}_i\},\{\mathbf{R}_I\}) \approx \chi_n(\{\mathbf{R}_I\}) \cdot \psi_{\mathrm{el},n}^{\{\mathbf{R}_I\}}(\{\mathbf{r}_i\}) \tag{2.9}$$

The derivation can be found in Appendix A. This leads to the electronic wave function, which is commonly used in quantum chemistry to calculate the energy of a system of a given geometry:

$$\hat{H}_{\mathrm{el}}(\mathbf{r}_i)\psi_{\mathrm{el},n}^{\{\mathbf{R}_I\}}(\{\mathbf{r}_i\}) = E_{\mathrm{el}}\psi_{\mathrm{el},n}^{\{\mathbf{R}_I\}}(\{\mathbf{r}_i\}) \tag{2.10}$$

with the electronic hamilton operator:

$$\hat{H}_{\text{el}} = \hat{T}_e(\mathbf{r}) + \hat{V}_{ee}(\mathbf{r}) + \hat{V}_{ext}(\text{r}) \tag{2.11}$$

The external potential ($\hat{V}_{ext}$) is the interaction between the electrons and the nuclei whose positions $\mathbf{R}_I$ enter parametrically in the equation. For every position of the nuclei, the electronic energy can be calculated resulting in the potential energy surface of a molecule $E_{\text{el}}(\mathbf{R})$. However, the SDE can only be solved analytically for the Hydrogen atom. For polyelectronic atoms or molecules the solution of the high dimensional SDE is only available using numerical methods. In the following sections the most common methods are introduced.

2.3 Hartree Fock

The hamilton operator (equation 2.8) can be divided into one electron operators $\hat{h}(i)$ and two electron operators $\hat{v}(i,j)$, with i and j being the electron indices.

$$\hat{h}(i) = -\frac{1}{2}\nabla_i^2 - \sum_{I=1}^{M} \frac{Z_I}{\mathbf{r}_{iI}} \quad \text{and} \quad \hat{v}(i,j) = \frac{1}{\mathbf{r}_{ij}} \tag{2.12}$$

using these two simplifications we can write the hamilton operator from equation 2.8 as follows:

$$\hat{H}_{\text{el}} = \sum_{i=1}^{n} \hat{h}(i) + \sum_{i=1}^{n}\sum_{j>i}^{n} \hat{v}(i,j) + \hat{V}_{NN} \tag{2.13}$$

The main approximation in the HF method is the assumption that the electrons do only interact with each other on a mean filed approach ($\hat{V}_{ee} = 0$). Therefore, the wave function can be written as a product of one electron orbitals $\phi_i(i)$ using a Slater determinant $\Phi_{\text{HF}}(1,2,...,n)$:

$$\Phi_{\text{HF}}(1,2,...,n) = (n!)^{-\frac{1}{2}} \begin{vmatrix} \phi_1(1) & \phi_2(1) & \dots & \phi_n(1) \\ \phi_1(2) & \phi_2(2) & \dots & \phi_n(2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(n) & \phi_2(n) & \dots & \phi_n(n) \end{vmatrix} \tag{2.14}$$

where all n electrons are distributed over all (n!) orbitals ϕ_n. In contrary to the Hartree Product, the Slater determinant accounts for the Pauli principle[32], where a system of electrons is described by an antisymmetric wave function. To obtain the Slater determinant with the lowest energy, the variational method[33] is used. After some mathematical transformations (which are omitted here), the Hartree-Fock eigenvalue equation can be written as:

$$\hat{h}(i)\phi_i + \sum_{i=1}^{n} \underbrace{\langle\phi_i|\mathbf{r}_{ij}^{-1}|\phi_i\rangle}_{\hat{\text{J}}} \phi_i - \sum_{i=1}^{n} \underbrace{\langle\phi_i|\mathbf{r}_{ij}^{-1}|\phi_j\rangle}_{\hat{\text{K}}} \phi_i = \sum_{i=1}^{n} \epsilon_{ij}\phi_i(i) \tag{2.15}$$

where ϵ_{ij} is the energy eigenvalue of the orbital ϕ_i, $\hat{J}$ the coulomb operator, and $\hat{K}$ the exchange operator. Defining the Fock-operator $\hat{f} = \hat{h}_i + \sum_{i=1}^{n} \hat{J}_i - \sum_{i=1}^{n} \hat{K}_i$ the eigenvalue equation can be written as:

$$\hat{f}|\phi_i\rangle = \epsilon_i|\phi_i\rangle \tag{2.16}$$

The Hartree-Fock equation (2.16) can be solved numerically. However, this is extremly time intensive and therefore not practical. To circumwent this problem Roothaan and Hall introduced the basis set approach, where the HF wave function Φ_{HF} is approximated with a basis set expansion:

$$|\Phi_{\mathrm{HF}}\rangle = \sum_{k=1}^{K} c_{ki}|\psi_i\rangle \tag{2.17}$$

using the local combination of atomic orbital (LCAO) approach, every c_{ik} of a molecular orbital would be the coefficient for an atomic orbital ψ_i. A summary of different basis sets can be found in this review[34]. Using equation 2.17 and multiplying with $\langle\psi_i|$, the HF equation (2.16) can then be rewritten as:

$$\sum_{k=1}^{K} c_{ki}\langle\psi_i|\hat{f}|\psi_i\rangle = \epsilon_i \sum_{k=1}^{K} c_{ki}\langle\psi_k|\psi_i\rangle \tag{2.18}$$

These k equations can now be summarized in a matrix form leading to the Roothaan-Hall equation:

$$\mathbf{FC} = \mathbf{SC}\epsilon \tag{2.19}$$

where $\mathbf{F}$ is the fock matrix with matrix element $F_{ki} = \langle\psi_k|\hat{f}|\psi_i\rangle$, $\mathbf{S}$ is the overlap matrix with matrix element $S_{ki} = \langle\psi_k|\psi_i\rangle$, $\mathbf{C}$ is the coefficient matrix, and ϵ is the eigenvalue diagonal matrix. This matrix equation can now be solved iteratively using the self-consistent-field (SCF) method: Figure 2.1 shows a flow chart with the working principle of the Hartree-Fock self-consistent-field method. The individuall steps are listed below:

1. Choose a basis function ψ_i and calculate the integrals $\hat{h}$, $\hat{J}$, $\hat{K}$, and $\hat{S}$.
2. Use an initial guess for the coefficients c_{ki} and calculate the Fock matrix.
3. Solve the Roothaan-Hall equation (2.19).
4. Check for convergence.
5. If converged → calculate molecular properties
6. If not converged → go back to step 3 using the newly acquired coefficients c_{ki}

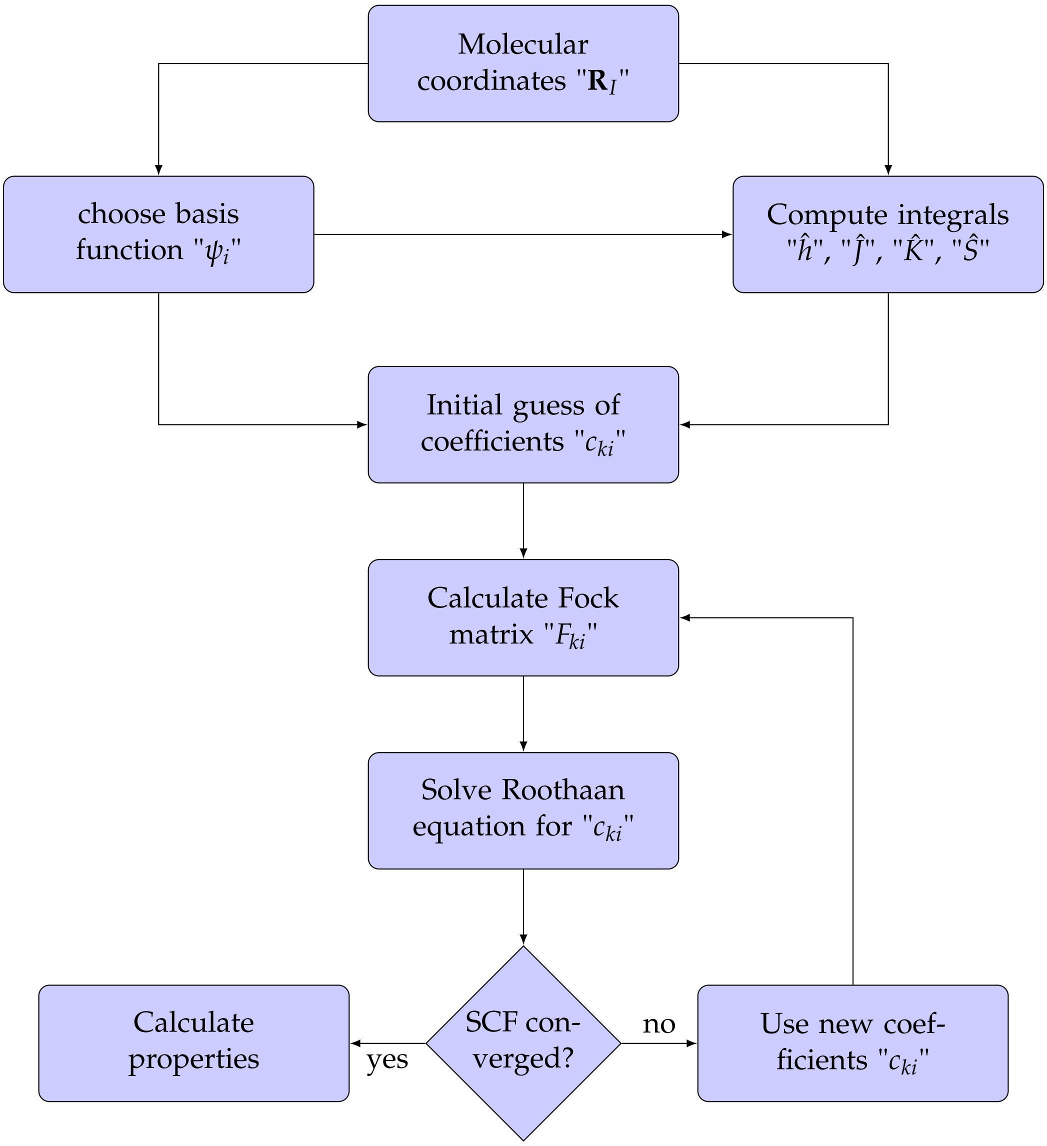

Molecular coordinates "$\mathbf{R}_I$"
choose basis function "ψ_i"
Compute integrals "$\hat{h}$", "$\hat{J}$", "$\hat{K}$", "$\hat{S}$"
Initial guess of coefficients "c_{ki}"
Calculate Fock matrix "F_{ki}"
Solve Roothaan equation for "c_{ki}"
SCF converged?
yes
no
Calculate properties
Use new coefficients "c_{ki}"

FIGURE 2.1: Flow chart showing the working principle of the self consistent filed method to solve the Roothaan-Haal equation.

2.4 Electron Correlation

Since HF does not consider electron correlation effects, significant errors can emerge from such approximations. Therefore, multiple post HF methods were developed to introduce electron correlation. The introduction of electron correlation improves the accuracy of QM calculations but also comes with a trade-off, namely the time to solution scales exponentially w.r.t. the number of electrons. In the following sections the four most common electron correlation methods are briefly discussed: The three post Hartree-Fock methods: The Møller-Plesset perturbation theory (MP2), the coupled cluster (CC) method, the configuration interaction (CI) method, as well as the density functional theory (DFT).

2.4.1 Post Hartree-Fock methods

Møller-Plesset perturbation theory:
In perturbation theory, a problem is solved by taking an already known solution of

a related and simpler problem (the unperturbed system). Then treating the problem to be solved with a power series approach. The first term is the solution of the unperturbed system. Every following term describes the perturbation to the unperturbed system. This power series is then truncated after a chosen number of terms. Keeping the first three terms results in a perturbation theory of second order, where the first term is the unperturbed system. In the *Moller-Plesset perturbation theory*, the unperturbed wave function is the Hartree-Fock wave function (Φ_{HF}). The perturbed Hamiltonian $\hat{H}'$ is the difference between the HF Hamiltonian $\hat{H}^0$ and the true molecular electronic Hamiltonian $\hat{H}$:

$$\hat{H}' = \hat{H} - \hat{H}^0 = \sum_i \sum_{j>i} \frac{1}{r_{ij}} - \sum_{j=1}^{n} \sum_{k=1}^{n} [\hat{J}_k(j) - \hat{K}_k(j)] \tag{2.20}$$

The perturbed Hamiltonian introduces the electronic repulsion which is missing in the Hartree-Fock interelectronic potentials (which is only an average potential). Since the first order correction to the energy ($E_0^{(0)} + E_0^{(1)}$) recovers the HF energy[35], the second order term has to be taken into account to improve the Hartree-Fock energy:

$$E_0^{(2)} = \sum_{s \neq 0} \frac{|\langle \psi_s^{(0)} | \hat{H}' | \Phi_{\mathrm{HF}} \rangle|^2}{E_0^{(0)} - E_s^{(0)}} \tag{2.21}$$

with $\psi_s^{(0)}$ being the unperturbed functions formed from n different spin orbitals. To perform an MP2 calculation, one first obtains Φ_{HF}, E_{HF}, and virtual orbitals from an SCF MO calculation. Then the second order energy correction ($E^{(2)}$) is calculated, which gives the total MP2 energy:

$$E_{MP2} = E_{HF} + E^{(2)} \tag{2.22}$$

In general, the second order perturbation theory (MP2) is used. The higher order terms can be taken into account but although their contribution is not small, they are usually omitted because the computational cost is too demanding.

Configuration interaction method:
Another method to tackle the electron interaction is the *configuration interaction* (CI) method. Here, the wave function is expanded in linear combinations of slater determinants.

$$\Psi_{\mathrm{CI}} = \sum_{i=0} c_i \psi_i = c_0 \psi_0 + c_1 \psi_1 + c_2 \psi_2 + ... \tag{2.23}$$

The first term corresponds to the HF wave function ($\Phi_{\mathrm{HF}} = c_0 \psi_0$). The following terms are slater determinants where electrons are mixed with virtual orbitals to account for the electron interaction. If all possible linear combinations are used a full CI (FCI) calculation is obtained. In practice, due to the enormous computational cost of a FCI, the wave function is truncated after several terms. For a CI singles doubles (CISD) calculation only the first three terms are considered, the HF determinant and the first, as well as the second excited determinant.

An important problem of the truncation is the size consistency which is not fulfilled anymore. A quantum chemical method should yield the same result for two systems A and B, once calculated at infinite separation (no interaction between the systems) and the other in two seperate calculations. This is not fulfilled in the CI method[36]. Here, the Davidson correction[37] can account for the missing higher order excitations to correct for the truncation error:

$$\Delta E_{\mathrm{Q}} = (1 - c_0^2)(E_{\mathrm{CISD}} - E_{\mathrm{HF}}) \tag{2.24}$$

where c_0 is the coefficient of the HF wavefunction, E_{CISD} and E_{HF} are the energies of the respective methods and ΔE_{Q} is the correction to the E_{CISDTQ} energy. In a multi reference CI (MRCI) calculation several ground state reference determinants are considered to account for a better description of the system at hand.

Coupled-Cluster method:
The coupled-cluster theory describes the electron correlation through excitation of electrons i and j to virtual orbitals a and b with an amplitude t_{ij}^{ab}, this *cluster* can be written as:

$$|\Phi_{\mathrm{HF}}\rangle + \sum_{a<b} t_{ij}^{ab} \hat{\tau}_{ij}^{ab} |\Phi_{\mathrm{HF}}\rangle \tag{2.25}$$

where $|\Phi_{\mathrm{HF}}\rangle$ is the HF wave function and $\hat{\tau}_{ij}^{ab}$ is the excitation operator. The cluster ij can be *coupled* with all the other clusters to get the *coupled cluster* wave function $|\mathrm{CC}\rangle$:

$$|\mathrm{CC}\rangle = \Pi_{a>b;i>j}(1 + t_{ij}^{ab} \hat{\tau}_{ij}^{ab})|\Phi_{\mathrm{HF}}\rangle \tag{2.26}$$

We can define a *cluster operator* $\hat{T}$ (not the kinetic operator) which can be written as:

$$\hat{T} = \sum_{\mu} \hat{\tau}_{\mu} \tag{2.27}$$

using this *Ansatz* we can simplify equaton 2.24 to $|CC\rangle = \exp(\hat{T})|\Phi_{\mathrm{HF}}\rangle$. If only the first two terms in $\hat{T}$ are considered, we obtain a Coupled-Cluster singles doubles wave function with $\hat{T} = \hat{T}_1 + \hat{T}_2$ which can be written as:

$$\Psi_{\mathrm{CCSD}} = \exp(\hat{T}_1 + \hat{T}_2)|\Phi_{\mathrm{HF}}\rangle \tag{2.28}$$

$$= \exp(\hat{T}_1)\exp(\hat{T}_2)|\Phi_{\mathrm{HF}}\rangle \tag{2.29}$$

In practice the similarity transformed Hamiltonian ($\exp(-\hat{T})\hat{H}\exp(\hat{T})$) is used. This way, not the wave function but the Hamiltonian is parameterized and we can write the energy expression:

$$E_{\mathrm{CC}} = \langle \Phi_{\mathrm{HF}} | \exp(-\hat{T})\hat{H}\exp(\hat{T}) | \Phi_{\mathrm{HF}} \rangle \tag{2.30}$$

which can be simplified to the compact notation:

$$E_{\mathrm{CC}} = \langle \Phi_{\mathrm{HF}} | \hat{H}^T | \Phi_{\mathrm{HF}} \rangle \tag{2.31}$$

The gold standard for a long time (and still) is the CCSD(T) calculation where the triple excitation enters in a perturbation theory approach. This method is commonly used for bench mark calculations to compare new developed methods, eg. functionals in DFT calculations (which will be discussed in the following section).

2.4.2 Density functional theory

In density functional theory (DFT) the electronic energy of a system can be calculated using the electron density $\rho(\mathbf{r})$ instead of the wave function Ψ. The density only depends on three spatial coordinates. The first Hohnerg-Kohn theorem[38] proves that the density is sufficient to describe the system. Therefore, the electronic energy $E_{\text{el},0}$ can be written as a functional of the electron density $\rho(\mathbf{r})$ resulting in $E_{\text{el}}[\rho(\mathbf{r})]$. The second Hohnberg-Kohn theorem[38] allows to use the variational principle to find $\rho(\mathbf{r})$ through minimization of said functional. From the electronic SDE (equation 2.10) it can be seen that the energy contains three terms: The kinetic energy, the electron-electron interaction, and the external potential (interaction between electrons and nuclei). Therefore, the energy functional can be written as:

$$E_{\text{el},0}[\rho] = T[\rho] + V_{\text{ext}}[\rho] + V_{ee}[\rho] \tag{2.32}$$

Using $T_s[\rho] = -\frac{1}{2}\sum_i^n \langle \phi_i | \nabla^2 | \phi_i \rangle$ as the kinetic energy of non interacting electrons (denoted with the subscript s), the external potential $V_{\text{ext}}[\rho] = \int \hat{V}_{\text{ext}}\rho(\mathbf{r})d\mathbf{r}$, and the electron-electron interaction as the coulomb interaction $J[\rho(\mathbf{r})] = \frac{1}{2}\int \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2$, inspired by the HF eigenvalue equation (equation 2.16). The energy functional can thus be rewritten as:

$$E_{\text{el},0}[\rho] = T_s[\rho] + V_{\text{ext}}[\rho] + J[\rho] + E_{xc}[\rho] \tag{2.33}$$

where we introduced the *exchange correlation* functional:

$$E_{xc}[\rho] = (T[\rho] - T_s[\rho]) + (V_{ee}[\rho] - J[\rho]) \tag{2.34}$$

The exchange correlation functional is the term collecting all the unknown expressions or the error that is made by treating the electron-electron interaction classically (non-interactinc kinetic part and coulomb potential). Using the density written in non-interacting Kohn-Sham orbitals θ_i^{KS} ($\rho(\mathbf{r}) = \sum_i^n |\theta_i^{\text{KS}}|^2$) and applying the variational theorem the Kohn-Sham equation is obtained:

$$\left[-\frac{1}{2}\nabla^2 + v_{\text{ext}}(\mathbf{r}) + J(\mathbf{r}) + v_{\text{xc}}(\mathbf{r})\right]\theta_i^{\text{KS}}(\mathbf{r}) = \epsilon_i \theta_i^{\text{KS}}(\mathbf{r}) \tag{2.35}$$

where we introduced the *exchange correlation* potential $v_{xc} = \frac{\partial E_{xc}[\rho]}{\partial \rho}$. This set of equations are similar to the HF equations (2.16) with the exchange operator ($\hat{K}$) being replaced by the exchange correlation potential ($v_{xc}(\mathbf{r})$).

The analytical expression of $E_{xc}[\rho]$ and $v_{xc}(\mathbf{r})$ is not known. Therefore a lot of work was invested in finding accurate approximations (fits) for the exchange correlation functional, a guide through the functional zoo can be found in[39]. Figure 2.2 shows Jacob's ladder for the different types of functionals.

Starting from the bottom of Figure 2.2, the first step of the ladder is the local density approximation (LDA) where we assume a homogeneous electron gas (ρ varies slowly with position) resulting in:

$$E_{xc}^{\text{LDA}}[\rho] = \int \rho(\mathbf{r})\epsilon_{xc}(\rho)d\mathbf{r} \tag{2.36}$$

where $\epsilon_{xc}(\rho)$ is the exchange and correlation energy per electron in said homogeneous electron gas. Taking the derivative leads to the exchange correlation potential:

$$v_{xc}^{\mathrm{LDA}} = \frac{\partial E_{xc}^{\mathrm{LDA}}}{\partial \rho} = \epsilon_{xc}(\rho(\mathbf{r})) + \rho(\mathbf{r})\frac{\partial \epsilon_{xc}(\rho)}{\partial \rho} \tag{2.37}$$

ϵ_{xc} can be written as a sum of exchange and correlation parts:

$$\epsilon_x(\rho) = -\frac{3}{4}\left(\frac{3}{\pi}\right)^{1/3} \cdot (\rho(\mathbf{r}))^{1/3} \qquad \epsilon_c(\rho) = \epsilon_c^{\mathrm{VWN}}(\rho). \tag{2.38}$$

where $\epsilon_c^{\mathrm{VWN}}(\rho)$ is a known function[40]. For the exchange energy functional we get:

$$E_x^{\mathrm{LDA}} = \int \rho \epsilon_x d\mathbf{r} = -\frac{3}{4}\left(\frac{3}{\pi}\right)^{1/3} \int [\rho(\mathbf{r})]^{4/3} d\mathbf{r} \tag{2.39}$$

For open shell molecules the local spin density approximation (LSDA) is used. The difference to LDA is that electrons with opposite spin do not have to share a spatial orbital. Densities of electrons with α and β spin are treated separately:

$$E_{xc} = E_{xc}[\rho^\alpha, \rho^\beta] \tag{2.40}$$

and the exchange energy functional can be written as:

$$E_x^{\mathrm{LSDA}} = -\frac{3}{4}\left(\frac{6}{\pi}\right)^{1/3} \int [(\rho^\alpha)^{4/3} + (\rho^\beta)^{4/3}] d\mathbf{r} \tag{2.41}$$

The next step on the ladder is the generalized gradient approximation (GGA). In the LDA approach we assumed no change of the density w.r.t. to position. This is now accounted for in the GGA approach by using gradients:

$$E_{xc}^{\mathrm{GGA}}[\rho^\alpha, \rho^\beta] = \int f(\rho^\alpha, \rho^\beta, \nabla\rho^\alpha, \nabla\rho^\beta) d\mathbf{r} \tag{2.42}$$

Also, the GGA energy functional can be split into an exchange and a correlation part:

$$E_{xc}^{\mathrm{GGA}} = E_x^{\mathrm{GGA}} + E_c^{\mathrm{GGA}} \tag{2.43}$$

One example of a GGA exchange functional was developed by Perdew and Wang in 1988 (B^{88}) which is a gradient correction to the E_x^{LSDA} functional:

$$E^{\mathrm{B88}} = E_x^{\mathrm{LSDA}} + \Delta E_x^{\mathrm{B88}} \tag{2.44}$$

with :

$$\Delta E_x^{\mathrm{B88}} = -b \sum_{\sigma=\alpha,\beta} \int \frac{(\rho^\sigma)^{4/3}\chi_\sigma^2}{1 + 6b\chi_\sigma \ln[\chi_\sigma + (\chi_\sigma^2 + 1)^{1/2}]} d\mathbf{r} \tag{2.45}$$

where $\chi_\sigma = |\nabla\rho^\sigma| / (\rho^\sigma)^{4/3}$ and b is an empirical parameter fitted to the HF exchange energy.

The next step on the ladder goes one step further and adds second derivatives and are the so called metaGGA's (or mGGA's), which are also referred to as the kinetic energy density. These functionals have the form:

$$E_{xc}^{\mathrm{mGGA}}[\rho^\alpha, \rho^\beta] = \int f(\rho^\alpha, \rho^\beta, \nabla\rho^\alpha, \nabla\rho^\beta, \nabla^2\rho^\alpha, \nabla^2\rho^\beta, \tau_\alpha, \tau_\beta) d\mathbf{r} \tag{2.46}$$

where $\tau_{\alpha/\beta}$ are the *Kohn-Sham kinetic energy densities* for the α and β spin electrons, respectively:

$$\tau_{\alpha/\beta} = \frac{1}{2}\sum_i |\nabla\theta^{\mathrm{KS}}_{\alpha/\beta,i}|^2 \tag{2.47}$$

Hybrid functionals, the next step on the ladder, add parts together from E_x and E_c from GGA or mGGA with the *exact* exchange (from the HF exchange operator $\hat{K}$ using KS orbitals):

$$E_{xc} = E_x^{\mathrm{GGA}} + c_x E_x^{\mathrm{exact}} + E_c^{\mathrm{GGA}} \tag{2.48}$$

where c_x is an empirical parameter fitted on different data sets.

Another improvement are the double hybrid functionals using the MP2 second order energy correction terms to account for electron correlation by adding virtual orbitals:

$$E_{xc} = E_{xc}^{\mathrm{hybrid}} + (1-a)E_c^{\mathrm{KS-MP2}} \tag{2.49}$$

where $E_c^{\mathrm{KS-MP2}}$ is calculated from the MP2 equation (2.21) using KS-orbitals and the parameter a is fitted to a specific data set to yield good results for a particular set of molecules.

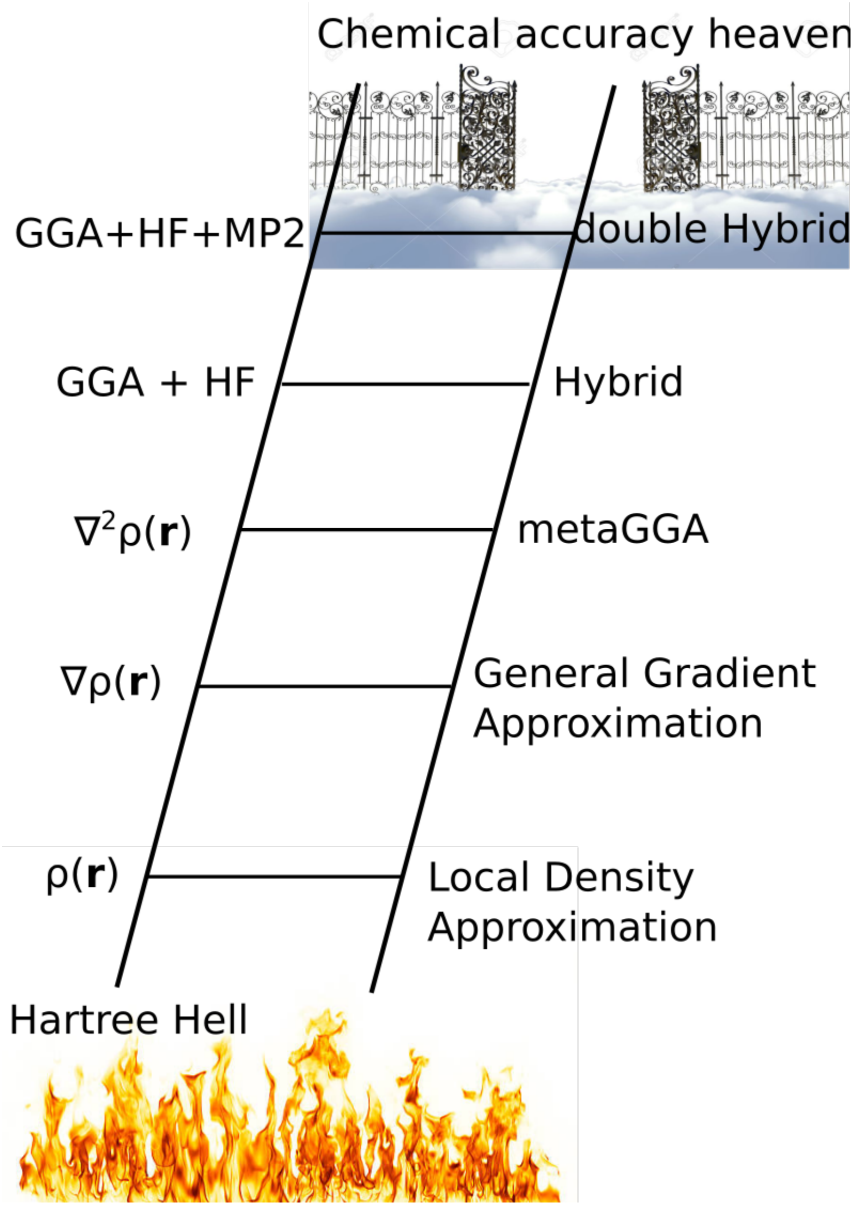


FIGURE 2.2: Jacob's ladder showing different approximations for the exchange correlation functional. First step (LDA) using only the density $\rho(\mathbf{r})$, second step (GGA) using the density $\rho(\mathbf{r})$ and gradients $\nabla\rho(\mathbf{r})$, third step using the density $\rho(\mathbf{r})$, the gradients $\nabla\rho(\mathbf{r})$, and second derivatives $\nabla^2\rho(\mathbf{r})$ or kinetic energy density $\tau_{\alpha/\beta}$, the fourth step adds exact HF exchange, and the last steps adds virtual KS orbitals using a MP2 approach.

Chapter 3

Quantum Machine Learning

The first part of the following section relies on the book of Vapnik[41].

3.1 Introduction

The mathematical analysis of machine learning processes began in the 1960's with the introduction of the perceptron by F. Rosenblatt[42,43]. The idea itself was not new, and already derived in the neurophysiologic literature[44], but this was the first time a model was created to solve pattern recognition problems using a program - an algorithm[41]. Using this model, it was possible to solve the classification problem with a simple rule to separate data of two different categories when given examples - training data. The perceptron has n input data $x = (x^1, ..., x^n) \in \mathbf{X} \subset \mathbf{R}^n$ and is connected via neurons to one output $y \in \{-1, 1\}$ as shown in Figure 3.1.

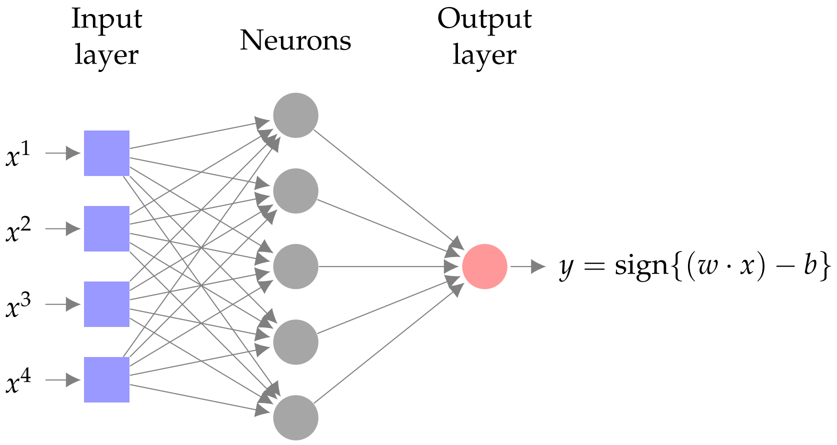


FIGURE 3.1: Illustration of a perceptron with input layers $\{x^1, ..., x^4\}$ (blue rectangles), neurons (gray circles), and output layer y (red circle).

The connection between the output and the inputs is given by:

$$y = \text{sign}\{(w \cdot x) - b\} \tag{3.1}$$

where $(w \cdot x)$ is the inner product of two vectors, b is a threshold value, and $\text{sign}(u)$ is the sign function returning 1 if $u > 0$ and -1 if $u \geq 0$. The neuron divides the space $\mathbf{X}$ into two subspaces where y takes either the value 1 or -1. These two subspaces are separated by the hyperplane:

$$(w \cdot x) - b = 0 \tag{3.2}$$

During the learning, the perceptron chooses the hyperparameter w and b. In this case, learning means finding the optimal combination of hyperparameters for all neurons given a training set. When in 1990's gradient based techniques were introduced[45], the discontinuous function (e.q. 3.2) was substituted with a continuous function using the following conditions:

$$y = S\{(w \cdot x) - b\} \qquad S(-\infty) = -1, \quad S(+\infty) = 1 \tag{3.3}$$

with S being the $\tanh(u)$ function. This way, the training of the perceptron was improved by the gradient approach using the so called back propagation. This was the birth of the neural networks, as they are known today.

The field of machine learning grew fast over the past decades and great milestones were achieved, some famous examples are: In 1997 Deep Blue beating a chess grand master[46], in 2016 AlphaGo wins 99.8% of all games against state of the art GO-engines and won 5-0 games against the human European GO champion[47], and in 2017 AlphaZero defeats the chess engine stockfish[48]. Furthermore, in 2017 DeepMinds AlphaStar beats top human Starcraft 2 (SC2) players[49]. This was a breakthrough because, in contrary to the board games, SC2 adds an additional difficulty to the learning tasks, since the entire map is not know *a priori* and therefore has to be discovered, hence not all data is initially available.

With ML growing bigger, it also attracted the attention of the fields of computational sciences. From 2010 to 2019 the number of publications containing the key words "machine learning" and either "chemistry" or "materials" grew from ~100 towards ~1'500 publications per year[50].

The most important quantity in quantum chemistry is the energy of a molecule from which most other properties can be derived. To get the energy of a system the electronic Schrödinger equation for a given set of atom coordinates $\mathbf{R}_I$ has to be solved:

$$\{Z_I, \mathbf{R}_I\} \xrightarrow{\hat{H}\Psi} E \tag{3.4}$$

where Z_I is the nuclear charge, $\hat{H}$ the hamilton operator, Ψ the wave function, and E the energy of the system. Finding the solution of the Schrödinger equation (SDE) is an NP hard problem[51] and the time to solution scales exponentially with the number of electrons of the systems. As described in the previous chapter, the more accurate results are targeted the more time has to be invested. The scaling of the Hartree-Fock (HF) method is N^4, with N being the number of electrons. The density functional theory (DFT) has a similar scaling but generally outperformes the HF method, which lead to the Nobel price for Walter Kohn and John A. Pople in 1998. Post HF methods implementing electron correlation like Moller-Plesset perturbation theory (MP), Coupled-Cluster (CC) and Configuration Interaction (CI) are more accurate but with the cost of N^5, N^6, and N^7, respectively.

Hence, to circumvent the problem of solving the SDE for all molecules in a given data set, machine learning techniques allow us to infer energies of similar molecules and therefore reduce the calculation time. Machine learning models take the same input ($Z_I, \mathbf{R}_I$) from quantum chemical calculations, but use training data to predict energies of unknown compounds, as shown below ($\hat{H}\Psi \rightarrow$ QML):

$$\{Z_I, \mathbf{R}_I\} \xrightarrow{QML} E \tag{3.5}$$

In an early attempt, von Lilienfeld *et. al.*[52] introduced a ML model using kernel ridge regression (KRR) which takes the coulomb term as a representation and estimated atomization energies of an out of sample test set with an accuracy of ∼10kcal/mol. Although neural networks are a crucial part in quantum machine learning, kernel ridge regression models offer an advantage when a system is well known (prior knowledge). Given this information, representations in KRR are able to compete and even outperform neural networks, a summary can be found in Faber *et al*[53]. A summary of machine learning models in chemical compound space can be found in[54] and the review[9] shows the recent developments in quantum machine learning in chemical reaction space. Although, in the beginnings of machine learning neural networks were predominant, in computational sciences, especially in quantum chemistry, kernel ridge regression became a fundamental part in quantum machine learning. In this thesis kernel ridge regression models were applied to chemical compound space, especially quantum chemical reaction space.

This chapter begins with an introduction to quantum machine learning, in particular, kernel ridge regression. First, a general introduction to the theory of KRR is given, then in section 3.2.1 the representations used in this work are introduced, followed by section 3.2.2 covering the training and evaluation of models using an example machine learning task, and finally section 3.3 which covers the data sets used in this work.

3.2 Kernel Ridge Regression

Ridge regression belongs to the supervised learning techniques, where the input space is mapped to a feature space within which fitting is applied. This transformation has to be found individually for every system and can grow computationally expensive. To circumvent this problem the "kernel trick" is applied where the inner product of the representations of two compounds are replaced by the so-called kernel function. The kernel uses the data in the input space and returns the dot product of the transformed vectors in the feature space. For example, the kernel trick for a second degree polynomial is derived below. Let $\phi(a)$ and $\phi(b)$ be the transformations of the two points **a** and **b** and said transformation be:

$$\begin{aligned}
\phi(a)^T \cdot \phi(b) &= \begin{pmatrix} a_1^2 \\ \sqrt{2}a_1a_2 \\ a_2^2 \end{pmatrix} \cdot \begin{pmatrix} b_1^2 \\ \sqrt{2}b_1b_2 \\ b_2^2 \end{pmatrix} && (3.6)\\
&= a_1^2b_1^2 + 2a_1b_1a_2b_2 + a_2^2b_2^2 && (3.7)\\
&= (a_1b_1 + a_2b_2)^2 && (3.8)\\
&= \left(\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right)^2 && (3.9)\\
&= (\mathbf{a} \cdot \mathbf{b})^2 && (3.10)
\end{aligned}$$

In this case the kernel function is a polynomial kernel of second degree:

$$k(a,b) = (a \cdot b)^2 \tag{3.11}$$

When given a set of N training instances $(\mathbf{x}_i, y_i)$ a query property $y_q^{\text{est}}(\mathbf{x}_q)$ can be estimated:

$$y_q^{\text{est}}(\mathbf{x}_q) = \sum_i^N \alpha_i k(\mathbf{x}_i, \mathbf{x}_q) \tag{3.12}$$

where $\mathbf{x}_i$ is the representation (discussed further below), $k(\mathbf{x}_i, \mathbf{x}_q)$ is the kernel element between the training compounds i and the query compound q, and α_i is the set of regression coefficients which can be obtained with the following matrix equation:

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y} \tag{3.13}$$

with λ beeing the regularizer, $\mathbf{I}$ the identity matrix, and $\mathbf{K}$ the kernel matrix of the training compounds (i, j) with the kernel element $k(\mathbf{x}_i, \mathbf{x}_j)$. The two kernels used throughout this work are the gaussian (3.14) and the laplacian (3.15) kernel:

$$k_{\mathrm{g}}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\sigma^2}\right) \tag{3.14}$$

$$k_{\mathrm{l}}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_1}{\sigma}\right) \tag{3.15}$$

The kernel element is a similarity measurement between two representations. The estimation of a query property $y_q^{\text{est}}(\mathbf{x}_q)$ is achieved by comparing the representation of the target compound $(\mathbf{x}_q)$ to all the training compounds $(\mathbf{x}_i)$, summing up the contributions of the individual training points as described in equation 3.12.

In QML only one property (in general the energy of a molecule) is learned. In geometry optimizations or transition state searches both, energies and forces, are required. The operator quantum machine learning approach[19,20] allows to train a model, in a similar way as shown in equation 3.13, simultaneously on energies and forces using following loss function:

$$J(\boldsymbol{\alpha}) = \left\| \begin{bmatrix} \mathbf{y} \\ \mathbf{f} \end{bmatrix} - \begin{bmatrix} \mathbf{K} \\ -\frac{\partial}{\partial \mathbf{r}}\mathbf{K} \end{bmatrix} \boldsymbol{\alpha} \right\|_2^2 \tag{3.16}$$

After training, the regression coefficients $\boldsymbol{\alpha}$ can be used to predict the energies $(\mathbf{y}^{\text{est}})$:

$$\mathbf{y}^{\text{est}} = \mathbf{K}_s \boldsymbol{\alpha} \tag{3.17}$$

and the forces $(\mathbf{f})$:

$$\mathbf{f}^{\text{est}} = -\frac{\partial}{\partial \mathbf{r}} \mathbf{K}_s \boldsymbol{\alpha} \tag{3.18}$$

where $\mathbf{K}_s$ is the test kernel containing training and test instances.

3.2.1 Representations

The representation $\mathbf{x}$ is the numerical way the environment (e.g. a molecule) is encoded in the ML model. The representation should fulfill certain conditions: i) it should to be unique, meaning that two molecules (e.g. enantiomers) that have different energies, also differ in the representation space, ii) the representation has to be invariant with respect to the translations or rotations.

Coulomb matrix[55]

The CM is a two body representation that takes the coulomb potential (nuclear charge) of the atoms scaled by their distances as off diagonal elements. The diagonal terms in the CM are the nuclear charges of the atoms to the power of 2.4 which leads to:

$$\mathbf{x}_{IJ} = \begin{cases} 0.5 Z_I Z_J^{2.4}, & \text{if } I = J \\ \frac{Z_I Z_J}{||\mathbf{R}_I - \mathbf{R}_J||}, & \text{otherwise} \end{cases} \tag{3.19}$$

where Z is the nuclear charge and R is the position of the atom.

Bag of Bonds (BoB)[56]

BoB is also a two body representation and was derived from the CM. BoB uses the nuclear coulomb terms from the CM and groups them into different bins, the so called bags, for all the elemental atom pair combinations.

SLATM[57]

The "Atomic Spectrum of London-Axilrod Teller Muto Potential" (SLATM) representation contains two- and three body terms. The two-boy terms uses the London dispersion function, rather than coulomb repulsion like CM and BoB. The three-body part consists of the Axilrod-Teller-Muto potential[58,59].

FCHL

FCHL18 contains one-, two-, and three-body terms, where the one-body term encodes the position of the element in the PSE (group and period). Its two-body terms contain all atom-to-atom distances (R) scaled by R^{-4}, and the three-body term contains all angles between each three atom combination scaled by R^{-2}. FCHL19[60] is an updated version of FCHL18[61] but limits itself to two- and three-body terms.

One-hot encoding

The one-hot encoding is a simple geometry free representations, generally used in prediction of properties of proteins[62,63]. The representation is a vector containing 0's and 1's - a bit vector - encoding if feature is present or not. For example a protein can be encoded by its amino acids, or a chemical reaction using the same scaffold but a variety of functional, attacking, or leaving groups, can be encoded as well.

A detailed review of physics-inspired structural representations can be found in[64].

3.2.2 Training & Testing

Representation & data set split

Throughout this work, the QMLcode[65] was used to train QML models and predict properties. Therefore, as an example, three models using the representations CM, BoB, and FCHL19, were trained on the conformer data set from Ramakrishnan *et. al*[66] to illustrate how QML models are generated, trained, and tested. The data set contains 6095 constitutional isomers of the sum formula $C_7H_{10}O_2$. The property for this example is the atomization energy on the B3LYP/6-31G(2df,p) level of theory.

To train and test a model the data set is commonly split into two parts: training and test set as illustrated in Figure 3.2. First, the representations for all compounds was computed and then the data set was randomly split into 5000 training and 1095 test compounds. The following code shows the generation of the coulomb matrix and the split into training and test set. The other representations are created in a similar fashion. For the sake of simplicity, only clips of the code are shown below. The entire code can be found on github[67].

```python
# calculate the representation (CM)
for mol in mols:
    mol.generate_coulomb_matrix(size=5) # equation 3.19
# get representation and split the data set
X = np.array([mol.representation for mol in mols])
X_training, X_test = X[:5000], X[5000:]
```

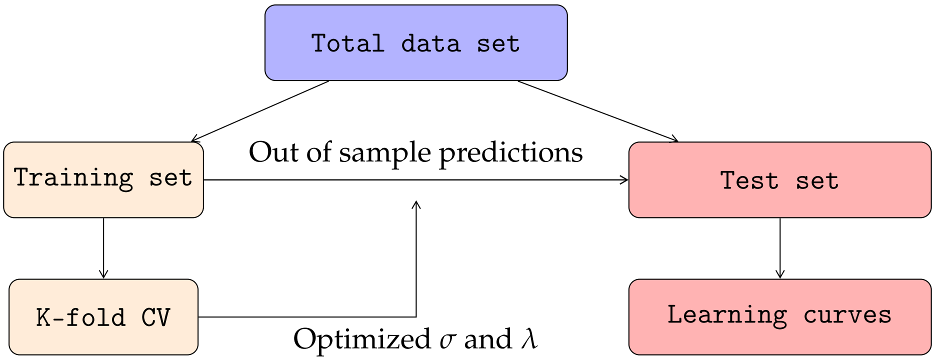


FIGURE 3.2: Scheme showing how the data set is split into training and test set. Training set is used to optimize hyperparameters σ and λ using k-fold cross validation where the training set is split again into training and validations.

Cross Validation

To optimize the hyperparameters a 5-fold cross validation, a scan over different hyperparameters σ and λ, for the three models was performed on the training set as shown in Figure 3.2. Figure 3.3 shows the working principle of the k-fold cross validation. Here, the training set is split into five folds. Then, the model is trained on all but one fold and the left out fold is used as validation set. This is done iteratively over all the folds (five times), calculating the MAE at every step. This way all the training instances are part of the training as well as the validation. The code below shows an example for one fold (eg. 1st Iter in Figure 3.3) and one set of hyperparameters (σ and λ), starting by calculating the training kernel:

```
X_train, X_validation = X_training[:4000], X_training[4000:]
sigma = 25.6
llambda = 1e-7
K_train = gaussian_kernel(X_train, X_train, sigma) # equation 3.14
K_train[np.diag_indices_from(K_train)] += llambda
```

The next step is to train the model using equation 3.13:

```
Y_training = Y[:5000]
Y_train, Y_validation = Y_training[:4000], Y_training[4000:]
alpha = np.inverse(K_train) * Y_train # equation 3.13
```

Finally the predictions can be made (using equation 3.12) and compared to the validation set:

```
K_validation = gaussian_kernel(X_train, X_validation, sigma) # equation 3.14
Y_predicted = np.dot(K_validation, alpha)                    # equation 3.12
print(np.mean(np.abs(Y_predicted - Y_validation)))
```

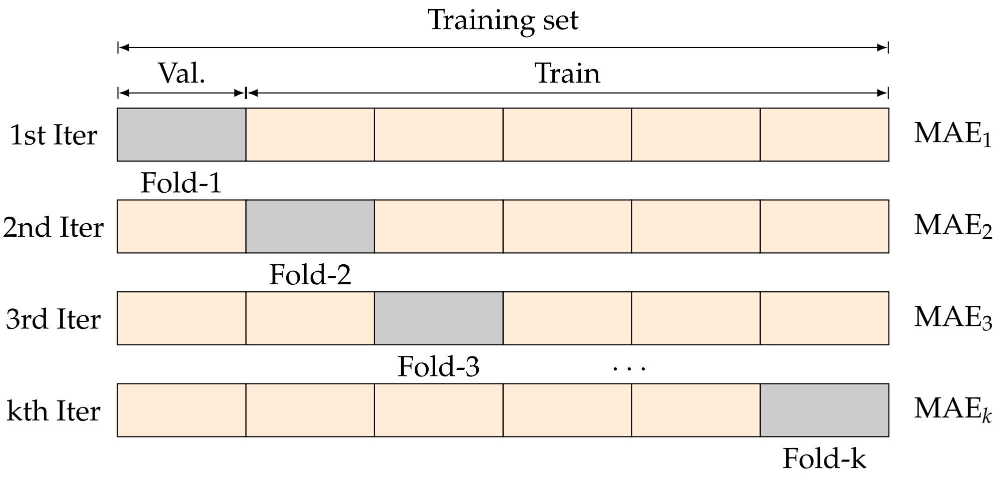


FIGURE 3.3: Scheme of working principle of k-fold cross validation for one set of hyperparameters σ and λ.

If this is done for all combinations of σ and λ and all folds, we can plot the results in a heat map. Figure 3.4 shows the resulting heatmap for the FCHL19 representation.

This procedure allows a direct and uncomplicated way to find the optimal set of hyperparameters. After optimizing the hyperparameter, the transferability and performance of the model can be tested using learning curves which will be discussed in the following section.

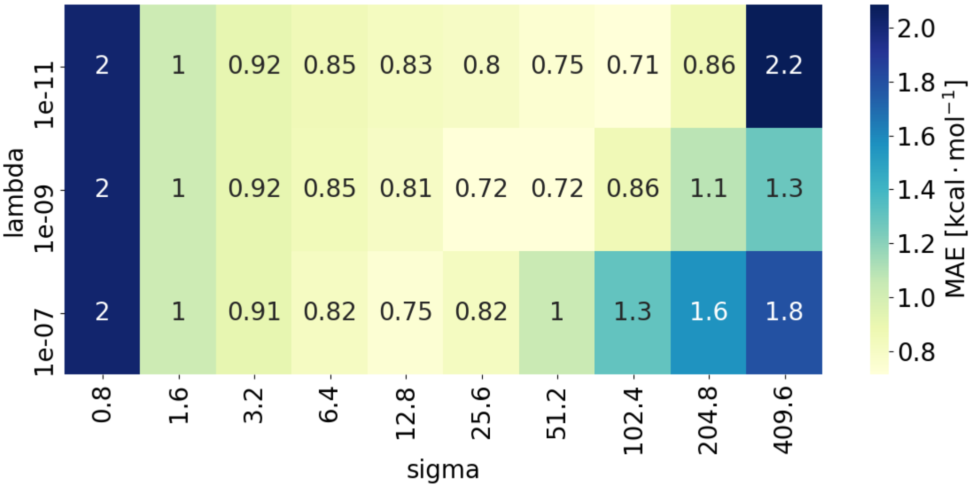


FIGURE 3.4: Results of the 5-fold cross validation shown as a heat map with encoded MAE in the color map for FCHL19.

Learning Curves

In general, the out of sample test error ϵ of a model decreases[41] with increasing number of training points N as shown in equation 3.20

$$\epsilon \propto bN^{-a} \tag{3.20}$$

Plotting the error ϵ vs. the training set size N on a log-log plot results in learning curves as shown in equation 3.21 and Figure 3.5.

$$\log(\epsilon) = -a\log(N) + b + HOT \tag{3.21}$$

a corresponds to the slope, b is the off set and HOT stands for higher order terms, which are negligible[41]. Good models, as shown in Figure 3.5 (left), decay linearly with respect to the training set size. For an inferior model, the learning curve flattens out at the end. Therefore, it does not improve any further by adding more training instances and preventing the model to reach the target accuracy. For good models, learning curves also give an estimate on how much more data is needed to obtain the desired accuracy, e.g. chemical accuracy (kcal/mol).

After obtaining the optimal combination of hyperparameters, the model is used to predict the out of sample test data (Figure 3.2), to evaluate its performance and transferability. To generate the learning curves the random sub-sampling cross validation[68] was used. The training set is shuffled after every iteration and the first N instances are taken for training with N being the number of training points. Both cross validation methods are commonly used, as well as applied during this work. Figure 3.5 (right) shows the performance of the three aforementioned models (CM, BoB, and FCHL19). Although, learning is achieved for all representations, only one reached

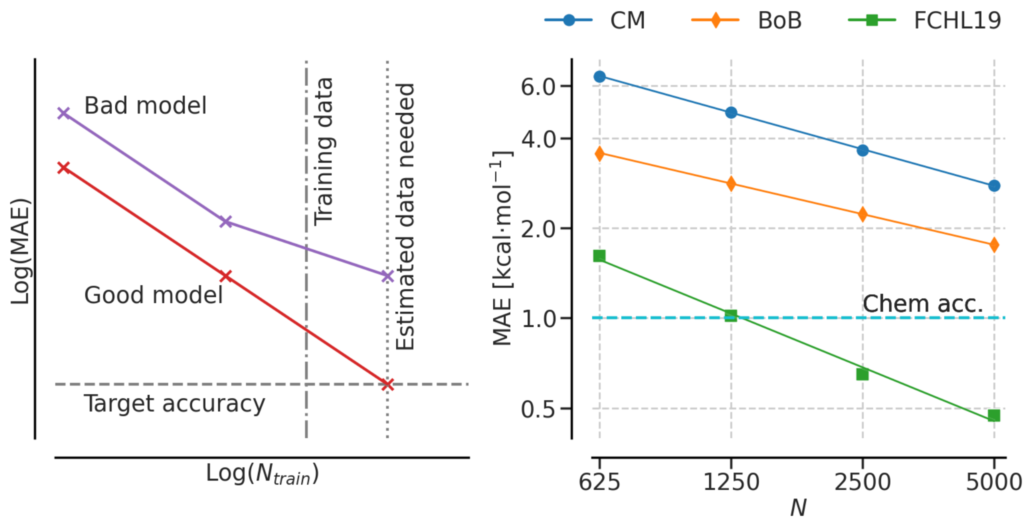


FIGURE 3.5: Fictitious learning curves showing basic principles of good/bad performing models, as well as basic principles of learning curves (left) and learning curves for the constitutional isomers of QM9 (right) for three representations.

chemical accuracy of 1kcal/mol, namely the more sophisticated FCHL19 representation. The data set contains constitutional isomers which are best described by interatomic distances and angles (FCHL19), rather than only using interatomic distances (CM and BoB), leading to a better performance of FCHL compared to the other representations.

3.3 Data sets

QM9[66]

The QM9 data set contains computed geometric, electronic and thermodynamic data for 134k stable small organic compounds including the heavy atoms: carbon, nitrogen, oxygen and fluorine. SMILES, containing up to nine heavy atoms (non hydrogen atoms), were chosen from the GDB-17 data set[69]. These SMILES[70,71] were then optimized using PM7 semi-empirical level of theory and subsequently were relaxed on B3LYP/6-31G(2df,p) level of theory.

QMrxn20[3]

This reaction data set contains 1’286 E2 (elimination) and 2’361 S_N2 (substitution) LCCSD/cc-pVTZ//MP2/6-311G(d) activation barriers. The leaving groups, for both reactions, are the halogens: fluorine, chlorine, and bromine. The nucleophiles (attacking group) are the halogens: fluorine, chlorine, bromine, as well as hydride. The functional groups are: $-CH_3$, $-NH_2$, -CN, and $-NO_2$.

The initial reactant geometries from the reaction data set were obtained by generating the unsubstituted molecule (hydrogen atoms instead of functional groups and fluorine as leaving group) without the nucleophile. Subsequently substituting the hydrogen atoms with functional groups span the chemical space. For every reactant a conformer search using the universal force field (UFF) was performed and the lowest lying conformer geometries were then further optimized on MP2/6-311G(d) level of theory.

Similarly, as described in the previous paragraph, the starting geometries for the transition state (TS) search were obtained. A transition state search was performed on the unsubstituted case and from the found TS the chemical space was spanned by exchanging the hydrogen atoms with functional groups.

QMspin[72]

A carbene chemical space of roughly 8'000 small organic molecules derived from 4'000 randomly selected QM9 molecules by abstracting two hydrogens from every saturated carbon center. The resulting geometries were relaxed for both states, singlet and triplet. For the triplet case the geometry was optimized using B3LYP with the def2-TZVP molecular basis and the def2-TZVPP density fitting basis. For the singlet case the geometry was optimized using the complex active space self consistent filed (CASSCF) method using the cc-pVDZ-F12 molecular orbital and the cc-pVTZ density fitting basis.

Subsets

For chapter 4 seven subsets were generated, derived from the aforementioned three data sets (QM9, QMspin, and QMrxn) containing five different levels of theories (CCSD(T), MRCI, B3LYP, MP2, CASSCF) using four basis sets, as well as three different calculation methods (singlepoint (SP), geometry optimization (GO), and transition state (TS) search calculations). The subsets are: $\mathbf{QM9}^{\mathbf{SP}}_{\mathbf{CC/DZ}}$, $\mathbf{QM9}^{\mathbf{SP}}_{\mathbf{CC/TZ}}$, $\mathbf{QM9}^{\mathbf{GO}}_{\mathbf{B3LYP}}$, $\mathbf{QMrxn}^{\mathbf{GO}}_{\mathbf{MP2}}$, $\mathbf{QMrxn}^{\mathbf{TS}}_{\mathbf{MP2}}$, $\mathbf{QMspin}^{\mathbf{SP}}_{\mathbf{MRCI}}$, and $\mathbf{QMspin}^{\mathbf{GO}}_{\mathbf{CASSCF}}$. The main term states the data set from where the compounds were taken, the subscript defines the method (level of theory), and the superscript defines the type of calculation. A more detailed description of the subsets can be found in chapter 4 section 4.3.3.

Chapter 4

Machine learning the computational cost of quantum chemistry

Machine learning the computational cost of quantum chemistry
S. Heinen, M. Schwilk, G.F. von Rudorff, O.A. von Lilienfeld; Machine Learning: Science and Technology, 1, (2), 025002

4.1 Abstract

Computational quantum mechanics based molecular and materials design campaigns consume increasingly more high-performance compute resources, making improved job scheduling efficiency desirable in order to reduce carbon footprint or wasteful spending. We introduce quantum machine learning (QML) models of the computational cost of common quantum chemistry tasks. For single point, geometry optimization, and transition state calculations the out of sample prediction error of QML models of wall times decays systematically with training set size. We present numerical evidence for thousands of organic molecular systems including closed and open shell equilibrium structures, as well as transition states. Levels of electronic structure theory considered include B3LYP/def2-TZVP, MP2/6-311G(d), local CCSD(T)/VTZ-F12, CASSCF/VDZ-F12, and MRCISD+Q-F12/VDZ-F12. In comparison to conventional indiscriminate job treatment, QML based wall time predictions significantly improve job scheduling efficiency for all tasks after training on just thousands of molecules. Resulting reductions in CPU time overhead range from 10% to 90%.

4.2 Introduction

Solving Schrödinger's equation, arguably one of the most important compute tasks for chemistry and materials sciences, with arbitrary accuracy is a NP hard problem[51]. This leads to the ubiquitous limitation that accurate quantum chemistry calculations typically suffer from computational costs scaling steeply and non-linearly with molecular size. Therefore, even if Moore's law was to stay approximately valid[6], scarcity in compute hardware would remain a critical factor for the foreseeable future. Correspondingly, chemistry and materials based compute projects have been consuming substantial CPU time at academic high-performance compute centers on national and local levels worldwide. For example, in 2017 research projects from chemistry and materials sciences used ∼25 and ∼35% of the total available resources at Argonne Leadership Computing Facility[73] and at the Swiss National Supercomputing Center (CSCS)[74], respectively. In 2018, ∼30% of the resources at the National Energy Research Scientific Computing Center[75] were dedicated to chemistry and materials sciences and even ∼50% of the resources of the ARCHER[76] super computing facility over the past month (May 2019). Assuming a global share of ∼35% for the usage of the Top 500 super computers (illustrated in Figure 4.1) over the last 25 years, this would currently correspond to ∼0.5 exaFLOPS (floating point operations per seconds) per year. But also on most of the local medium to large size university or research center compute clusters, atomistic simulation consumes a large fraction of available resources. For example, at sciCORE, the University of Basel's compute cluster, this fraction typically exceeds 50%. Acquisition, usage, and maintenance of such infrastructures require substantial financial investments. Conversely, any improvements in the efficiency with which they are being used would result in immediate savings. Therefore a lot of work is done to constantly improve hardware and software of HPCs, e. g. at the International Supercomputing Conference NVIDIA announced the support of the Advanced RISC Machines (Arm) CPUs, which allows to build extremely energy efficient exascale computers, by the end of the year[77]. Compute applications on such machines commonly rely on schedulers optimizing the simultaneous work load of thousands of calculations. While these schedulers are highly optimized to reduce overhead, there is still potential for application domain specific improvements, mostly due to indiscriminate and humanly biased run time estimates specified by users. The latter is particularly problematic when it comes to ensemble set-ups characteristic for molecular and materials design compute campaigns with very heterogeneous compute needs of individual instances. One could use the scaling behaviour of methods to get sorted lists w.r.t wall times and improve scheduling by grouping the calculations by run time. For example the bottleneck of a multi-configuration self-consistent field calculation (MCSCF) is in general the transformation of the Coulomb and exchange operator matrices into the new orbital basis during the macro-iterations. This step scales as nm^4 with n the number of occupied orbitals and m the number of basis functions. All Configuration Interaction Singles Doubles (CISD) schemes that are based on the Davidson algorithm[37] scale formally as n^2m^4, where n the number of correlated occupied orbitals and m the number of basis functions[78]. As these methods (and basis sets) contain different scaling laws and geometry optimizations additionally depend on the initial geometry, a more sophisticated approach was applied: In this paper, we show how to use quantum machine learning (QML) to more accurately estimate run times in order to improve overall scheduling efficiency of quantum based ensemble compute campaigns.

Since the early 90's, an increasing number of research efforts from computer science has dealt with optimizing the execution of important standard classes of algorithms that occur in many scientific applications on HPC platforms[79–81], but also with predicting memory consumption[82], or, more generally, the computational cost itself.Such predictive models may even comprise direct minimization of the estimated environmental impact of a calculation as the target quantity in the model[83]. ML has already successfully been applied, however, towards improving scheduling itself[84], or entire compute work flows[85,86]. Furthermore, a potentially valuable application in the context of quantum chemistry may be the run time optimization of a given tensor contraction scheme on a specific hardware by predictive modelling techniques[87]. Another noteworthy effort has been the successful run time modeling and optimization of a self-consistent field (SCF) algorithm on various computer architectures in 2011[88] using a simple linear model depending on the number of retired instructions and cache misses. Already in 1996, Papay et al. contributed a least square fit of parameters in graph based component-wise run time estimates in parallelized self consistent field computations of atoms[89]. Other noteworthy work in the field of computational chemistry is the prediction of the run time of a molecular dynamics code[90], or the prediction of the success of DFT optimizations of transition metal species as a classification problem by Kulik and coworkers[91]. In the context of quantum chemistry and quantum mechanical solid state computations, very little literature on the topic is found. This may seem surprising, given the significant share of this domain on the overall HPC resource consumption (cf. Figure 4.1). To the best of our knowledge, there is no (Q)ML study that predicts the computational cost (wall time, CPU time, FLOP count) of a given quantum chemical method across chemical space.

Today, a large number of QML models relevant to quantum chemistry applications throughout chemical space exists[92–94]. Common regressors include Kernel Ridge Regression[53,95–99] (KRR), Gaussian Process Regression[100] (GPR), or Artificial Neural Networks[53,101–105] (ANN). For the purpose of estimating run times of new molecules, and contrary to pure computer science approaches, we use the same molecular representations (derived solely from molecular atomic configurations and compositions) in our QML models as for modeling quantum properties. As such, we view computational cost as a molecular "quasi-property" that can be inferred for new, out-of-sample input molecules, in complete analogy to other quantum properties, such as the atomization energy or the dipole moment.

4.3 Data

All QML approaches rely on large training data sets. Comprehensive subsets of the chemical space of closed shell organic molecules have been created in the past, e. g., the QM9[66] data set which is derived from a subset of the GDB17[69]. Further relevant data sets in the literature include, among others, reaction networks[106], closed shell ground state organometallic compounds[1], or non-equilibrium structures of small closed shell organic molecules[107]. Yet, regions of chemical space that may involve more sophisticated and costly quantum chemistry methods, such as open shell and strongly correlated systems[108,109] or chemical reaction paths, are still strongly underrepresented. For this study, we have generated and used timings of the computational cost associated to seven tasks which reflect variances of three common use cases: single point (SP), geometry optimization (GO) and transition state (TS) search calculations.

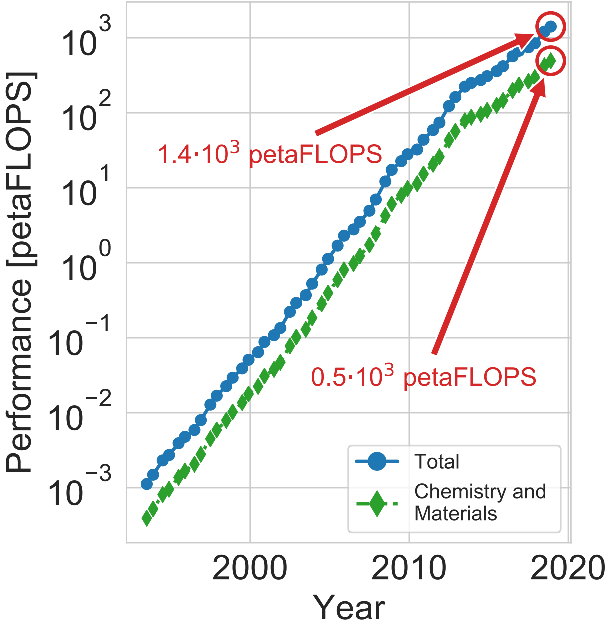


FIGURE 4.1: Compute resource growth of 500 fastest public supercomputers.[8] Estimated use by chemistry and materials sciences corresponds to 35%, corresponding to 2017 usage on Swiss National Supercomputing Center.[74].

4.3.1 Quantum Data Sets

We have considered coordinates coming from three different data sets (QM9, QMspin, QMrxn) corresponding to five levels of theory (CCSD(T), MRCI, B3LYP, MP2, CASSCF) and four basis set sizes. Molecules in the three different data sets consist of the following:

i) QM9 contains 134k small organic molecules in the ground state local minima with up to nine heavy atoms which are composed of H, C, N, O, and F. All coordinates were published in 2014[66]. Here, we also report the relevant timings.

ii) QMspin consists of carbenes derived from QM9 molecules containing calculations of the singlet and triplet state, respectively, with a state-averaged CASSCF(2e,2o) reference wave function (singlet and triplet ground states with equal weights). The entirety of this data set will be published elsewhere, here we only provide timings and QM9 labels.

iii) QMrxn consists of reactants and S_N2 transition states of small organic molecules with a scaffold of C_2H_6 which was functionalized with the following substituents: $-NO_2$, -CN, $-CH_3$, $-NH_2$, -F, -Cl and -Br. The entirety of this data set will be published elsewhere, here we only provide timings and geometries.

4.3.2 Toy System

To demonstrate that it is possible to learn the number of steps of an optimization algorithm, we apply our machine learning method to two cases from function optimization theory: quantifying the number of steps for an optimizer. The functions in question are the Rosenbrock function[110]

$$f(x,y) = (1-x)^2 + 100(y-x^2)^2 \tag{4.1}$$

and the Himmelblau function[111]

$$f(x,y) = (x^2+y-11)^2 + (x+y^2-7)^2 \tag{4.2}$$

The fucntions are shown in the top row of Figure 4.4 b) and c). We applied three representative optimizers in their SciPy 1.3.1[112] implementation on both functions: the "NM" simplex algorithm (Nelder-Mead[113]), the gradient based "BFGS" algorithm[114], and an algorithm using gradients and hessians (Conjugate Gradient with Newton search "N-CG"[115]). For every function and optimizer we performed 10200 optimizations from different starting points on a cartesian grid over the domain $-5 \leq x, y \leq 5$ in steps of 0.1. The minimum of the Rosenbrock function and the four minima of the Himmelblau function lie within this domain. Figure 4.4 b) row two, three, and four show a heatmap of the number of optimization steps for NM, BFGS, and N-CG, respectively, for Rosenbrock (left column) and Himmelblau (right column). Generally, the minimum searches on the Himmelblau function required much fewer steps (mostly reached after a few tens of iterations). While the gradient based optimizer BFGS clearly outperforms NM for both functions, the N-CG optimization of the Rosenbrock function did not converge with a iteration limit of 400 for a set of points in the region of $x < -0.5$ and $y > 2.5$. A very small step size for the N-CG algorithm implementation in SciPy in the critical region is responsible for the slow convergence.

4.3.3 Quantum Chemistry Tasks

The three data sets were then divided into the seven following tasks for which timings were obtained (See also Table 4.1):

$\mathbf{QM9^{SP}_{CC/DZ}}$ – 5736 PNO-LCCSD(T)-F12/VDZ-F12[116–118] single point energy timings. Details of the calculation results other than timings are subject of a separate publication[119].

$\mathbf{QM9^{SP}_{CC/TZ}}$ – 3497 PNO-LCCSD(T)-F12/VTZ-F12 single point energy timings.

$\mathbf{QMspin^{SP}_{MRCI}}$ – 2732 single point calculations using MRCISD+Q-F12/VDZ-F12[120–123]. Details of the calculation results other than timings are subject of a separate publication.[124]

$\mathbf{QM9^{GO}_{B3LYP}}$ – 3724 geometry optimization timings with initial B3LYP/6-31G*[125,126] geometries optimizing at the B3LYP/def2-TZVP level of theory.

$\mathbf{QMrxn^{GO}_{MP2}}$ – 8148 geometry optimization timings on MP2/6-311G(d) level of theory.

$\mathbf{QMspin^{GO}_{CASSCF}}$ – 1595 CASSCF(2e,2o)[Singlet]/VDZ-F12[127,128] geometry optimization timings.

$\mathbf{QMrxn^{TS}_{MP2}}$ – 1561 timings of transition state searches on MP2 level of theory.

Further details on the data sets can be found in section 1 of the supporting information (SI). A distribution of the properties (wall times) of the seven tasks is illustrated

TABLE 4.1: Seven tasks used in this work generated from three data sets (QM9, QMspin, QMrxn), using three use cases (SP, GO, TS) on different levels of theory and basis sets.

Task	$\mathbf{QM9^{SP}_{CC/DZ}}$	$\mathbf{QM9^{SP}_{CC/TZ}}$	$\mathbf{QMspin^{SP}_{MRCI}}$	$\mathbf{QM9^{GO}_{B3LYP}}$	$\mathbf{QMrxn^{GO}_{MP2}}$	$\mathbf{QMspin^{GO}_{CASSCF}}$	$\mathbf{QMrxn^{TS}_{MP2}}$
Use case	SP			GO			TS
Data set	QM9		QMspin	QM9	QMrxn	QMspin	QMrxn
Level	CCSD(T)	CCSD(T)	MRCI	B3LYP	MP2	CASSCF	MP2
Basis set	VDZ-F12	VTZ-F12	VDZ-F12	def2-TZVP	6-311G(d)	VDZ-F12	6-311G(d)
Size	5736	3497	2732	3724	8148	1595	1561
Code	Molpro	Molpro	Molpro	Molpro	ORCA	Molpro	ORCA

in Figure 4.2. Single point calculations (the two $\mathbf{QM9^{SP}_{CC}}$ tasks) and the geometry optimization (task $\mathbf{QM9^{GO}_{B3LYP}}$) have wall times smaller than half an hour. In general, the smaller the variance in the data, the less complex the problem and the easier it is for the model to learn the wall times. For geometry optimizations and more exact (also more expensive) methods (task $\mathbf{QMspin^{SP}_{MRCI}}$ and $\mathbf{QMspin^{GO}_{CASSCF}}$) the average run time is $\sim$ 9 hours. With a larger variance in the data the problem is more complex (higher dimensional) and the learning is more difficult (higher off-set).

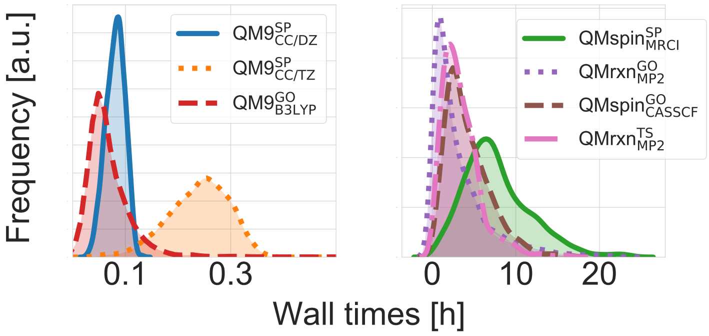


FIGURE 4.2: Wall time distribution of all tasks using kernel density estimation.

4.3.4 Timings, Code, and Hardware

The calculations were run on three compute clusters, namely our in-house compute cluster, the Basel University cluster (sciCORE) and the Swiss national supercomputer Piz Daint at CSCS. We used two electronic structure codes to generate timings. Molpro[129] was used to extract both CPU and wall times for data sets i) and ii), and ORCA[130] was used to extract wall times for data set iii). Further information of the data sets, the hardware, and the calculations can be found in section 3 to 4 of Appendix B. The retired floating point operations (FLOP) count of the local coupled cluster calculation task $\mathbf{QM9^{SP}_{CC/DZ}}$ was obtained as follows: The number of FLOPs have been computed with the *perf* Linux kernel profiling tool[131] for data set $\mathbf{QM9^{SP}_{CC/DZ}}$. *perf* allows profiling of the kernel and user code at run time with little CPU overhead and can give FLOP counts with reasonable accuracy. FLOP count is an adequate measure of the computational cost when the program execution is CPU bound by numerical operations, which is given in the PNO-LCCSD(T)-F12 implementation[116–118,132] in Molpro.

4.4 Methods

4.4.1 Quantum Machine Learning

In this study, we used kernel based machine learning methods which were initially developed in the 1950s[133] and belong to the supervised learning techniques. In ridge regression, the input is mapped into a feature space and fitting is applied there. However, the best feature space is *a priori* unknown, and its construction is computationally hard. The "kernel trick" offers a solution to this problem by applying a kernel k on a representation space $\mathcal{R}$ that yields inner products of an implicit high dimensional feature space: the Gram matrix elements $k(\mathbf{x}_i, \mathbf{x}_j)$ of two representations $\mathbf{x} \in \mathcal{R}$ between two input molecules i and j are the inner products $\langle i, j \rangle$ in the feature space. For example,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_1}{\sigma}\right) \tag{4.3}$$

or

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\sigma^2}\right) \tag{4.4}$$

with σ as the length scale hyperparameter, represent commonly made kernel choices, the Laplacian (eq. 4.3) or Gaussian kernel (eq. 4.4). Fitting coefficients $\boldsymbol{\alpha}$ can then be computed in input space via the inverse of the kernel matrix $[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$:

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y} \tag{4.5}$$

where λ is the regularization strength, typically very small for calculated noise-free quantum chemistry data.

Hence, kernel ridge regression (KRR) learns a mapping function from the inputs $\mathbf{x}_i$, in this case the representation of the molecule, to a property $y_q^{\text{est}}(\mathbf{x}_q)$, given a training set of N reference pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. Learning in this context means interpolation between data points of reference data $\{(\mathbf{x}_i, y_i)\}$ and target data $\{(\mathbf{x}_q, y_q^{\text{est}})\}$. A new property y_q^{est} can then be predicted via the fitting coefficients and the kernel:

$$y_q^{\text{est}}(\mathbf{x}_q) = \sum_i^N \alpha_i \cdot k(\mathbf{x}_i, \mathbf{x}_q) \tag{4.6}$$

For the toy systems, a Laplacian kernel was used, the representation corresponding simply to the starting point ($\mathbf{x} = (x, y)$) of the optimization runs. For the purpose of learning of the run times, we used two widely used representations, namely Bag of Bonds (BoB)[56] with a Laplacian kernel. BoB is a vectorized version of the Coulomb Matrix (CM)[95] that takes the Coulomb repulsion terms for all atom to atom distances and packs them into bins, scaled by the product of the nuclear charges of the corresponding atoms. This representation does not provide a strictly unique mapping[98,134] which may deteriorate learning in some cases (*vide infra*). The second representation used was atomic FCHL[60] with a Gaussian kernel. FCHL accounts for one-, two-, and three-body terms (whereas BoB only contains two-body terms). The one-body term encodes group and period of the atom, the two-body term contains interatomic distances R, scaled by R^{-4}, and the three-body terms in addition contain angles between all atom triplets scaled by R^{-2}.

To determine the hyperparameters σ and λ, the reference data was split into two parts, the training and the test set. The hyperparameters were optimized only within the training set using random sub-sampling cross validation. To quantify the performance of our model, the test errors, measured as mean absolute errors (MAE), were calculated as a function of training set size. The leading error term is known to be inversely proportional to the amount of training points used:[135]

$$\text{MAE} \approx a/N^b \tag{4.7}$$

The learning curves should then result in a decreasing linear curve with slope b and offset $\log a$:

$$\log(\text{MAE}) \approx \log(a) - b\log(N) \tag{4.8}$$

where a is the target similarity which gives an estimate of how well the mapping function describes the system[98] and b is the slope being an indicator for the effective dimensionality[136]. Therefore, good QML models are linearly decaying, have a low offset $\log(a)$ (achieved by using more adequate representations and/or baseline models[137]), and have steep slopes (large b).

For each task, QML models of wall times were trained and subsequently tested on out-of-sample test set which was not part of the training. As input for the representations the initial geometries of the calculations were used. To improve the predictions of geometry optimizations for the task $\mathbf{QMspin}^{\mathbf{GO}}_{\mathbf{CASSCF}}$, we split the individual optimization steps into the first step (GO1) and the subsequent steps (GO2), because the first step takes on average ~20% more time than the following steps (for more details we refer to section 1.4 of the SI). For learning the timings of the geometry optimization task GO2, we took the geometries obtained after the first optimization step.

As input for the properties, wall times were normalized with respect to the number of electrons in the molecules. Figure 4.3 shows the wall time overhead (CPU time to wall time ratio) for calculations run with Molpro. To remove runs affected by heavy I/O, wall time overheads higher than 3%, 5%, 10%, 30%, and 50% were excluded from the tasks $\mathbf{QM9}^{\mathbf{SP}}_{\mathbf{CC/DZ}}$, $\mathbf{QM9}^{\mathbf{SP}}_{\mathbf{CC/TZ}}$, $\mathbf{QMspin}^{\mathbf{SP}}_{\mathbf{MRCI}}$, $\mathbf{QMspin}^{\mathbf{GO}}_{\mathbf{CASSCF}}$, and $\mathbf{QM9}^{\mathbf{GO}}_{\mathbf{B3LYP}}$, respectively. In order to generate learning curves for all the seven tasks, all timings were normalized with respect to the median of the test set to get comparable normalized mean absolute errors (MAE). The resulting wall time out-of-sample predictions were used as input for the scheduling algorithm. Whenever the QML model predicted negative wall times, the predictions were replaced by the median of all non-negative predictions.
All QML calculations have been carried out with QMLcode[65]. Wall times and CPU times (Molpro) and wall times (ORCA) for all the seven tasks, as well as QML scripts can be found in Appendix B.

4.4.2 Application: Optimal Scheduling

Job Array and Job Steps

In many cases, efforts in computational chemistry or materials design require the evaluation of identical tasks on different molecules or materials. Distributing those tasks across a compute cluster is typically done in one of two ways. When using job

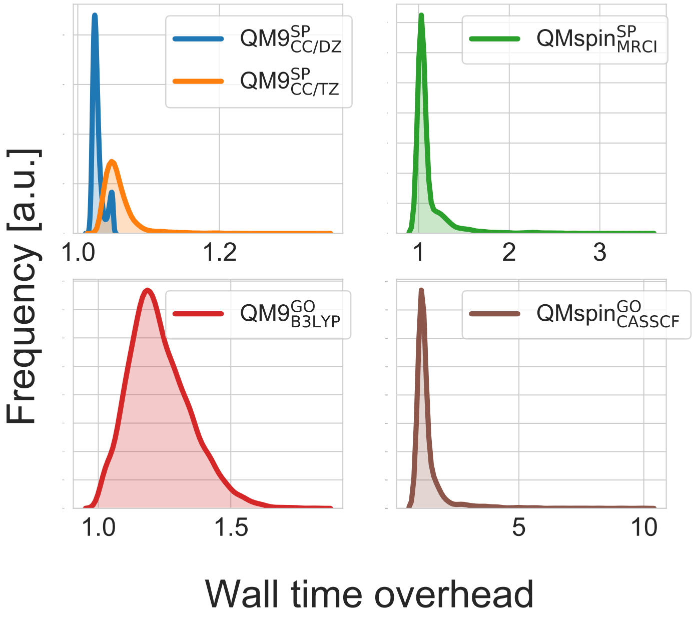


FIGURE 4.3: Wall to CPU time ratio (using kernel density estimation) for Molpro calculations to identify runs with high wall time overhead due to heavy I/O load on clusters.

arrays, the scheduler assigns compute resources to each calculation separately, such that the individual calculation is queued independently. This approach typically extends the total wall time, and has little overhead with the jobs themselves but leads to inefficiencies for the scheduler since the individual wall time estimate of each job needs to be (close to) the maximum job duration.

In the second approach, there are only few jobs submitted to the scheduler and tasks are executed in parallel as job steps. The first approach has little overhead with the jobs themselves but can lead to inefficiencies. The second approach yields inefficiencies due to lack of load balancing. These two common methods require no knowledge of the individual run time of each task, and usually rely on a conservative run time estimate in practice.

Scheduling Simulator

Using the QML based estimated absolute timings turns the scheduling of the remaining calculations into a bin packing problem. For this problem we used the heuristic first fit decreasing (FFD) algorithm which takes all run time estimates for all tasks, sorts them in decreasing order and chooses the longest task that fits into the remaining time of a compute job (for more details on FFD, see section 2 in the SI). If there is no task left that is estimated to fit into a gap, then no task is chosen and resources are released early.

We implemented a job scheduling simulator assuming idempotent uninterruptible tasks for all three job schedulers: Conventional job arrays, conventional job steps, and our new QML based job scheduler. Using a simulator is particularly useful because the duration of the job array and job step approaches depend on the (random) order of the jobs, and therefore requires averaging over multiple runs. We used this simulator in the context of two environments: our university cluster sciCORE (denoted S) where users are allowed to submit single-core jobs and the Swiss national supercomputer (CSCS, denoted L) where users are only allowed to allocate entire compute nodes of 12 cores. In all cases, we assumed that starting a new job via the scheduler takes 30 seconds and that every job queues for one hour. These numbers have been observed for queuing statistics of sciCORE and CSCS.

4.5 Results and Discussion

4.5.1 Toy System

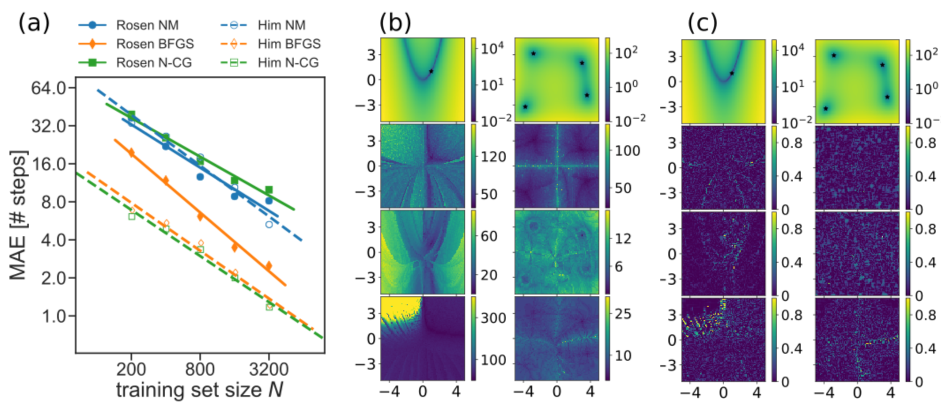


FIGURE 4.4: 2D non-linear toy systems consisting of the Rosenbrock (“Rosen”) and Himmelblau (“Him”) functions and minimum search with three optimizers (Nelder-Mead (NM), BFGS, and Newton-CG (N-CG)). a) Learning curves showing the prediction error of KRR for Rosen (solid lines) and Him (dashed lines) function using starting point (x, y) as representation input. b) Top row shows the function values for Rosen (left) and Him (right). Row two, three, and four show the number of optimization steps (encoded in the heat map) for 10200 starting points for NM, BFGS, and N-CG, respectively. c) Row two, three, and four show the relative prediction error of the ML model trained on the largest training set size $N = 3200$ for NM, BFGS, and N-CG, respectively.

From the total data set (10200 optimizations) 3200 were chosen randomly for every combination of optimizer and function and the prediction error was computed for different training set sizes N. Figure 4.4 a) shows the learning curves for the Rosenbrock (“Rosen”) and the Himmelblau (“Him”) functions. Well behaved learning curves were obtained for both functions and all optimizers. The ML models for Him-BFGS and Him-N-CG have a lower offset because the variance of the data set is smaller (between 0 and 25 optimization steps) than for the others (∼50-120 steps). The offset of Rosen-Newton-CG can be explained by the truncated runs which caused a non smooth area in the function space ($x < -0.5$ and $y > 2.5$) which leads to higher errors.

In addition to the learning curves, we computed the relative prediction errors of the different optimization runs. These results are shown in Figure 4.4 c). As expected, the errors get larger when the starting point is close to a saddle point: small

TABLE 4.2: QML results (normalized prediction errors) for seven task and both representations (BoB and FCHL) for largest training set size (N_{max}).

Calculation	**SP**			**GO**			**TS**
Label	$\text{QM9}^{\text{SP}}_{\text{CC/DZ}}$	$\text{QM9}^{\text{SP}}_{\text{CC/TZ}}$	$\text{QMspin}^{\text{SP}}_{\text{MRCI}}$	$\text{QM9}^{\text{GO}}_{\text{B3LYP}}$	$\text{QMrxn}^{\text{GO}}_{\text{MP2}}$	$\text{QMspin}^{\text{GO}}_{\text{CASSCF}}$	$\text{QMrxn}^{\text{TS}}_{\text{MP2}}$
$\boldsymbol{N_{\text{max}}}$	5000	3200	2000	3200	6400	1200	1000
BoB [%]	2.0	3.3	32.7	42.5	40.5	47.8	32.9
FCHL [%]	1.3	1.6	30.9	37.6	38.9	39.8	27.0

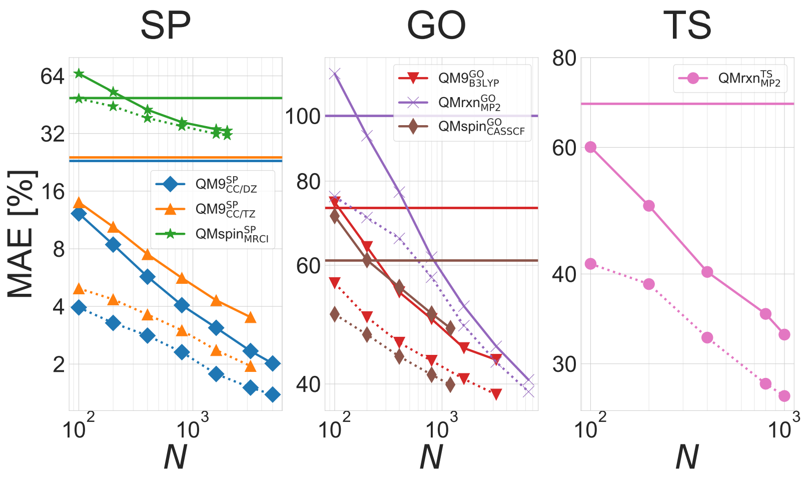


FIGURE 4.5: Learning curves showing normalized test errors (cross validated MAE divided by median of test set) for seven tasks using BoB (solid) and FCHL (dashed) representations. The model was trained on wall times normalized w.r.t. number of electrons. Horizontal lines correspond to the performance estimating all calculations have mean run time (standard deviation divided by mean wall time of the task).

changes in the starting point coordinates may lead to very different optimization paths. These discontinuities naturally occur for any optimizer based on the local information at the starting point and can be consistently observed in Figure 4.4 b). Additional discontinuities can also be observed depending on the optimizer. For all these regions larger relative errors for KRR can be observed [shown in Figure 4.4 c)] illustrating that small prediction errors rely on a reasonably smooth target function. In summary, we can show that KRR is capable of learning the discrete number of optimization steps which is a strong indication that the computational cost of quantum chemistry geometry optimization and transition state searches should be learnable in principle .

4.5.2 Quantum Machine Learning

Single Point (SP) Wall Times

In the following, learning of the wall times for the different quantum chemistry tasks is discussed, the learning of the corresponding CPU times has also been investigated and results of the latter are given in Appendix B. Figure 4.5 (left) shows the performance of QML models of wall times using learning curves for the SP use case. For the two similar tasks $\mathbf{QM9^{SP}_{CC/DZ}}$ and $\mathbf{QM9^{SP}_{CC/TZ}}$, the timings of the smaller basis set was consistently easier to learn, i.e. smaller training set required to reach similar predictive accuracy. Similarly to physical observables[60], the use of the FCHL representation results in systematically improved learning curve off-set with respect to BoB. It is substantially more difficult to learn timings of multi-reference calculations (task $\mathbf{QMspin^{SP}_{MRCI}}$), nevertheless, learning is achieved, and BoB initially also exhibits a larger off-set than FCHL, but the learning curves of the respective two representations converge for larger training set sizes. More specifically, for training set size $N = 1'600$, BoB/FCHL based QML models reach an accuracy of 3.1/1.8, 4.3/2.4, and 33.7/31.8 % for $\mathbf{QM9^{SP}_{CC/DZ}}$, $\mathbf{QM9^{SP}_{CC/TZ}}$, and $\mathbf{QMspin^{SP}_{MRCI}}$, respectively. Corresponding respective average wall times in our data-sets, distributions shown in Fig. 4.2, average at ∼6, 15, and 480 minutes. To the best of our knowledge, such predictive power in estimating compute timings has not yet been demonstrated for common quantum chemistry tasks.

The extraordinary accuracy that our model can reach in the prediction of the wall times for the $\mathbf{QM9^{SP}_{CC/DZ}}$ and $\mathbf{QM9^{SP}_{CC/TZ}}$ quantum chemistry tasks may be explained by the undlying quantum chemical algorithm. The tensor contractions in the local coupled cluster algorithm are sensitively linked to the chemically relevant many-body interactions expressed in the basis of localized orbitals. Therefore, the computational cost can be suitably encoded by atom-based machine learning representations.

In order to investigate the relative performance of BoB vs. FCHL further, we have performed a principal component analysis (PCA) on the respective kernels (training set size $N = 2'000$) for task $\mathbf{QMspin^{SP}_{MRCI}}$. The projection onto the first two components is shown in Figure 6.3, color-coded by the training instance specific wall times, and with eigen-value spectra as insets. For FCHL, the decay of the eigenvalues is very rapid (tenth eigenvalue already reaches 0.1). From the PCA projection, the number of heavy atoms emerges as a discrete spectrum of weights for the first principal component. The second principal component groups constitutional isomers. This reflects the importance of the one-body terms in the FCHL representation. The data covers well both components and the color various monotonically. All of this indicates a rather low dimensionality in the FCHL feature space which facilitates the learning. The kernel PCA plot of the FCHL representation shows that the learning problem is smooth in representation space and that there is a correlation between the property (computational cost) and the representation space. By contrast, the BoB's PCA projection onto the first two components displays a star-wise pattern with linear segments which indicate that more dimensions are required to turn the data into a monotonically varying hypersurface. The eigenvalue spectrum of BoB decays much more slowly with even the 100^{th} eigenvalue still far above 1.0. All of this indicates that learning is more difficult, and thereby explains the comparatively higher off-set.

Geometry Optimization (GO) Wall Times

Learning curves in Figure 4.5 (middle) shows that it is, in general, possible to build QML models of GO timings for the tasks considered. We obtained accuracies for BoB/FCHL for $N = 800$ of 50.0/43.3, 61.7/57.6, and 50.7/41.2% for tasks $\mathbf{QM9^{GO}_{B3LYP}}$, $\mathbf{QMrxn^{GO}_{MP2}}$, and $\mathbf{QMspin^{GO}_{CASSCF}}$, respectively.

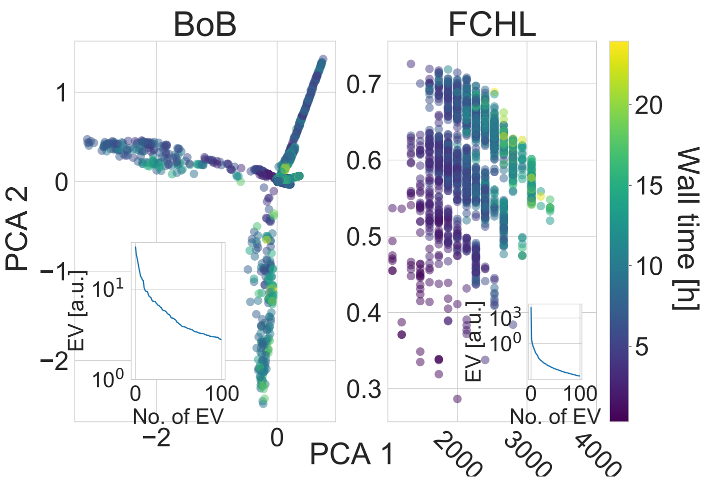


FIGURE 4.6: PCA plots of kernel elements for BoB (left) and FCHL (right) for data set $\mathbf{QMspin}^{\mathbf{SP}}_{\mathbf{MRCI}}$. The weights of the two first principal components for the molecules in the data sets are plotted against each other and corresponding wall times are encoded as a heat map. Insets show the first 100 eigenvalues on a log scale.

Interestingly, the comparatively larger off-set in the learning curves, however, indicates that it is more difficult to learn GO timings than SP timings. This is to be expected since GO timings involve not only SP calculations for various geometries but also geometry optimization steps. In other words, the QML model has to learn the quality of the initial guesses for subsequent GO optimizations. This can not be expected to be a smooth function in chemical space. Furthermore, the mapping from an initial geometry (used in the representation for the QML model) to the target geometry can vary dramatically when the initial geometry happens to be close to a saddle point (or a second order saddle point in the case of TS searches, see next section): Very slight changes in the initial geometry (or in the setup of the geometry optimization) may lead to convergence to very different stationary points on the potential energy surface. This makes the statistical learning problem much less well conditioned than for single point calculations, which also reflects in the larger variance of the geometry optimization timings compared to single point calculations. As such, GO timings represent a substantially more complex target function to learn than SP timings. Note that for any task (even for the toy system applications) we require a different QML model. The cost of the GO depends on the initial geometry and the convergence criteria. The latter varies only slightly within a data set. The former is part of the representation of the molecular structure and therefore captured by our model. The input structures for the task $\mathbf{QMrxn}^{\mathbf{GO}}_{\mathbf{MP2}}$ are derived from the same molecular skeleton and are therefore very similar. The same holds for task $\mathbf{QM9}^{\mathbf{GO}}_{\mathbf{B3LYP}}$ and $\mathbf{QMspin}^{\mathbf{GO}}_{\mathbf{CASSCF}}$ which are derived from QM9 molecules. The convergence criteria also stay the same for all calculations within a data set and would only cause a more difficult learning task if a machine was trained over several different data sets. We also showed with the toy system that it is possible to learn the number of steps for different optimizer starting from different areas on the surface (see Figure 4 b)). To further improve the performance of our model of task $\mathbf{QMspin}^{\mathbf{GO}}_{\mathbf{CASSCF}}$, we split the GO into the first GO step (GO1) and all subsequent steps (GO2). This choice has been motivated by our observation that most of the variance stemmed from the first GO step (requiring to build the wave-function from scratch), while the subsequent steps for themselves have a substantially smaller variance. The resulting learning curves are shown in Figure 4.7 and justify this separation in leading to an improvement of the QML model to reach errors of less than 25% at N = 800 (rather than more than 40%), as well as further improved job scheduling optimization (shown below in Figure 4.10).

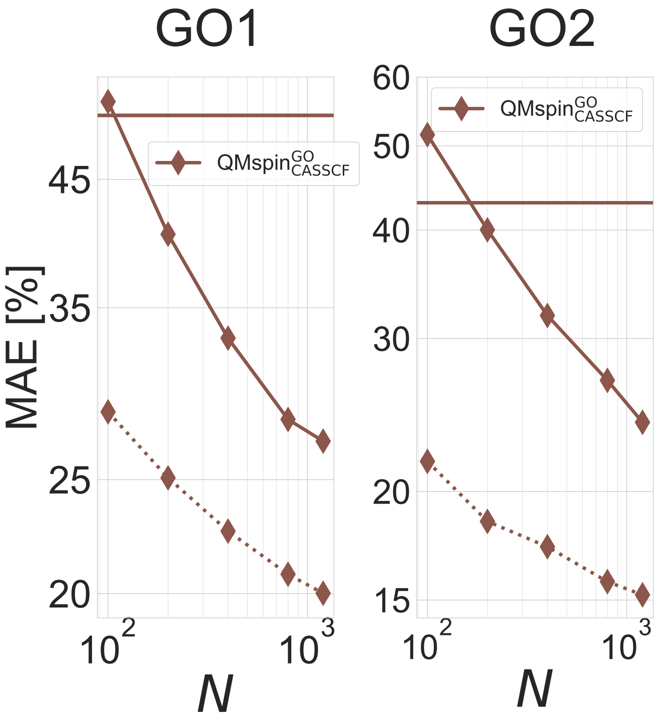


FIGURE 4.7: Learning curves showing normalized test errors (cross validated MAE divided by median of test set) for the first two geometry optimization steps on task **QMspin$^{\mathbf{GO}}_{\mathbf{CASSCF}}$** using BoB and FCHL as representations. The model was trained on CPU times divided by the number of electrons. Horizontal lines correspond to the performance estimating all calculations have mean run time (standard deviation divided by the mean wall time of the data set).

Transition State (TS) Wall Times

Transition state search timings were slightly easier to learn than geometry optimization timings (see Figure 4.5 (right)). Particularly for the larges training set size (N_{max} = 1000) for BoB/FCHL we obtained MAEs of 32.9/27.0% and reduced the off-set by ∼ 10% compared to learning curves for the GO use case. As already discussed in the previous section, the run time of GO and TS timings not only scales with the number of electrons but also depends on the initial structure. For the transition state search, the scaffold (which is close to a transition state) was functionalized with the different functional groups. Since the initial structures were closer to the final TS the offset of the learning curves is lower than for learning curves of the GO use case, where the initial geometries were generated with a semi empirical method (PM6) for task **QMrxn$^{\mathbf{GO}}_{\mathbf{MP2}}$**, carbenes were derived from QM9 molecules for task **QMspin$^{\mathbf{GO}}_{\mathbf{CASSCF}}$**, and geometries for task **QM9$^{\mathbf{GO}}_{\mathbf{B3LYP}}$** were obtained with a different basis set.

A summary of the results for all tasks for the largest training set size (N_{max}) can be found in Table 4.2.

Timings, Code, Hardware

Regarding hardware dependent models, within one data set we only used one electronic structure code which is also consistent with the general handling of the data set generation. The noise that is generated using different infrastructures affects the learning only in a negligible amount in our case, since the difference in hardware capabilities is minimal. When looking at the task $\mathbf{QMrxn}^{\mathbf{TS}}_{\mathbf{MP2}}$ where we used five different CPU types on two clusters (Table 1 in the SI), we could not find any evidence that different hardware affects the learning compared to other GO tasks that ran on only one CPU type and cluster. However the hardware for these calculations is still very similar. When it differs to a greater extant, the noise level will rise. The noise does not only depend on the cluster itself but also on other calculations running on the cluster which is non-deterministic and will limit the transferability of the ML models. For this reason we removed some of the timings with large I/O overhead using Figure 3. For the $\mathbf{QM9}^{\mathbf{SP}}_{\mathbf{CC}}$ tasks, the run time difference using the Intel MKL 2019 library[138] and OpenBlas 0.2.20[139] were computed for a few cases and are found to be only within a few percents of the wall time. Furthermore, run times of a native build of the Molpro software package version 2018.3 with OpenMPI 3.0.1[140], GCC 7.2.0[GCC], and GlobalArrays 5.7[141,142] and the shipped executable were compared and yielded run times within a few percents of difference. The FLOP calculations on the $\mathbf{QM9}^{\mathbf{SP}}_{\mathbf{CC}}$ data set have been performed on a compute node with 24 processors [Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz (Broadwell)]. The significant part of the FLOP clock cycles constituted of vectorized double precision FLOP on the full 256 bit FLOP register, i. e. the essential numerical operations of the quantum chemistry algorithm were directly measured. Hence, FLOP count constitutes a valuable measure of the compute cost in our case.[143] We anticipate that Hardware specific QML models will be used in practice.

Single Point (SP) FLOPs

To provide unequivocal numerical proof that it is justifiable to learn wall times we applied our models to FLOP counts for the task $\mathbf{QM9}^{\mathbf{SP}}_{\mathbf{CC/DZ}}$, shown in Figure 4.8. FLOP count as a "clean" measurement (almost no noise) for computational cost was slightly easier to learn than wall times and the learning curves show similar behaviour: The model trained on the same task $\mathbf{QM9}^{\mathbf{SP}}_{\mathbf{CC/DZ}}$ reaches ∼4% MAE already with just 400 training samples, while ∼1000 training samples were required in the case of wall times using BoB. For FCHL, the performance is similar but the slope is steeper for the FLOP model which indicates a faster learning or less noise.

4.5.3 Application: Optimal Scheduling

Job Array and Job Steps

For the scheduling optimization for all seven tasks ($\mathbf{QM9}^{\mathbf{SP}}_{\mathbf{CC/DZ}}$, $\mathbf{QM9}^{\mathbf{SP}}_{\mathbf{CC/DT}}$, $\mathbf{QMspin}^{\mathbf{SP}}_{\mathbf{MRCI}}$, $\mathbf{QM9}^{\mathbf{GO}}_{\mathbf{B3LYP}}$, $\mathbf{QMrxn}^{\mathbf{GO}}_{\mathbf{MP2}}$, $\mathbf{QMspin}^{\mathbf{GO}}_{\mathbf{CASSCF}}$, $\mathbf{QMrxn}^{\mathbf{TS}}_{\mathbf{MP2}}$), the QML model with the best representation (lowest MAE with maximum number of training points) was used which in all cases was FCHL. For the FFD algorithm absolute timing predictions are needed to make good decisions. The lower panel of Figure 9 shows the accuracy of the QML predictions. While the individual predictions (absolute not relative) are in

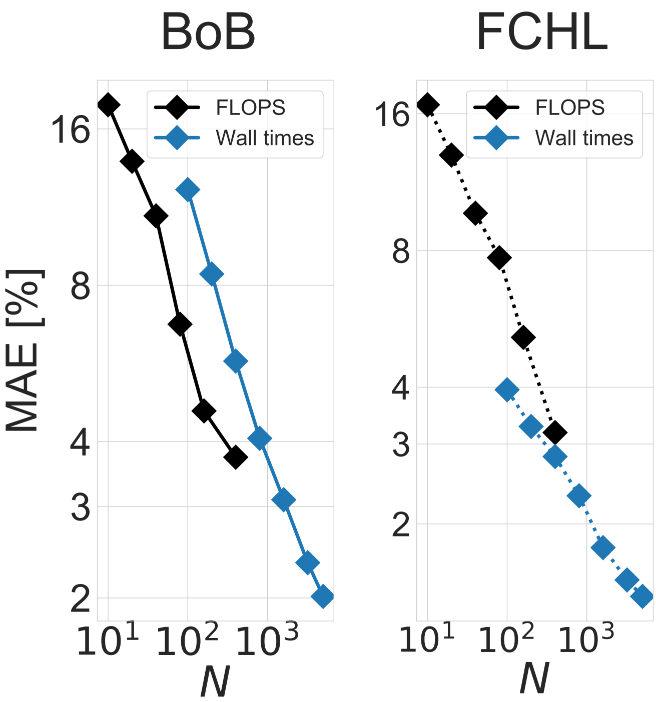


FIGURE 4.8: Learning curves showing normalized prediction errors (cross validated MAE divided by median of test set) for FLOP count and wall times on task $\mathbf{QM9^{SP}_{CC/DZ}}$ using BoB and FCHL representations.

many cases not perfect and partially still exhibit a significant MAE (cf. Figure 5), this level of accuracy is already sufficient to reduce the overhead of the job scheduling. The lower panel of Figure 4.9 shows the accuracy of the QML predictions. While the individual predictions (absolute not relative) are in many cases not perfect and partially still exhibit a significant MAE (cf. Figure 4.5), this level of accuracy is already sufficient to reduce the overhead or the wall time limits of the job scheduling. In particular, in the limit of a large number of cores working in parallel, our approach typically halved the computational overhead (data sets with closed shell systems and TS searches) while also reducing the time to solution by reducing the total wall time. This shows that for the scheduling efficiency problem, it is not required to obtain perfect estimates for the individual job durations, but rather reasonably accurate estimates. However, if there was the need for better accuracy, by virtue of the ML paradigm (prediction error decay systematically with training set size) this could easily be accomplished by decreasing the error simply through the addition of more training data.

When comparing the different methods in the upper panel of Figure 4.9, we see that the job array approach had no overhead for cases where single-core jobs can be submitted separately. While this is true it means that every job needs to wait in the queue again, thus increasing the total time to solution. For large task durations, this effect is less pronounced but typically the job array approach doubles the wall time which renders this approach unfavourable.

Using job steps alone becomes inefficient if the task durations are long, since the assumption that all tasks are roughly of identical duration will mean that interruptions

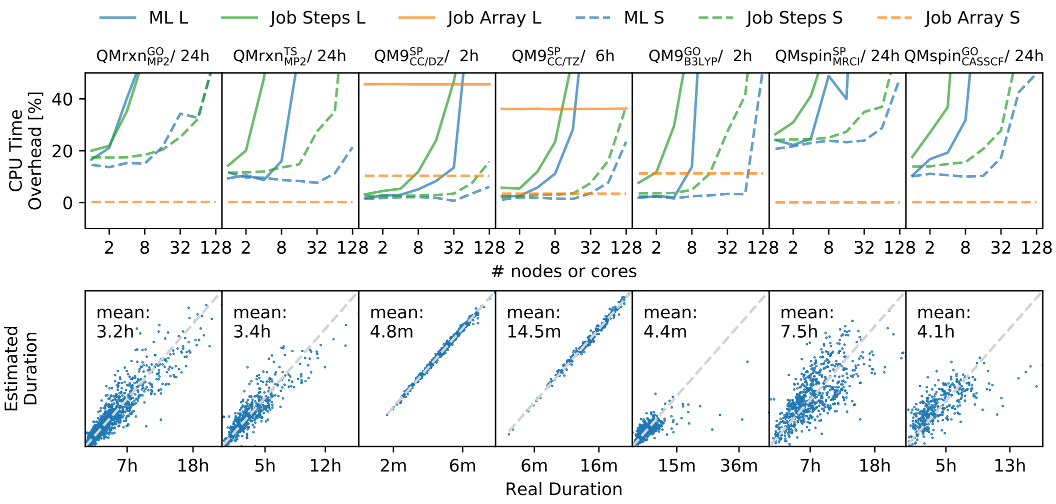


FIGURE 4.9: Scheduling efficiencies for the seven different tasks (columns) assuming a certain per-job wall time limit specified in column title. Infrastructure assumptions correspond to either a large (solid lines, L) compute center or a small (dashed lines, S) university compute center. Top row reports CPU time overhead reduction when using the QML based (blue) rather than the conventional (green, orange) packing. Results are given relative to the total CPU time needed for the calculations of each data set for established methods (job array and jobs steps, see text) and our suggested method (QML). Bottom row shows actual vs. predicted times (using FCHL as representation) for all calculations in each data set using maximum training set size.

of unfinished calculations occur more often. Having a more precise estimate allows for more efficient packing. This becomes important on large compute clusters where only full nodes can be allocated: In this case, the imbalance of the durations of calculations running in parallel further increases the overhead. Our method typically gave a parallelization overhead of 10-15% for a range of data sets. For example, in the task $\mathbf{QMrxn^{GO}_{MP2}}$, our approach allowed us to go to two orders of magnitude more compute resources and have the same overhead as job step parallelization. This is a strong case for using QML based timing estimates in a production environment – in particular, since the number of training data points required is very limited (see Figure 4.5).

Geometry Optimization Steps

Given that the number of steps of a geometry optimization is difficult to learn (see lower panel of Figure 4.9), the ability to accurately predict the duration of a single geometry optimization step allows to increase efficiency via another route. On hybrid compute clusters, the maximum duration of a single compute job is limited. We suggest to check during the course of a geometry optimization whether the remaining time of the current compute job is sufficient to complete another step. If not, it is more efficient to relinquish the compute resources immediately rather than committing them to the presumably futile undertaking of computing the next step. We refer to these strategies as the “simple approach” (take all CPU time you can, give nothing back) and the “QML approach” (give up resources early). Figure 4.10 shows the advantage of the QML approach: it allows to go towards shorter compute jobs and reduces the CPU time overhead by up to 90% for small wall time limits

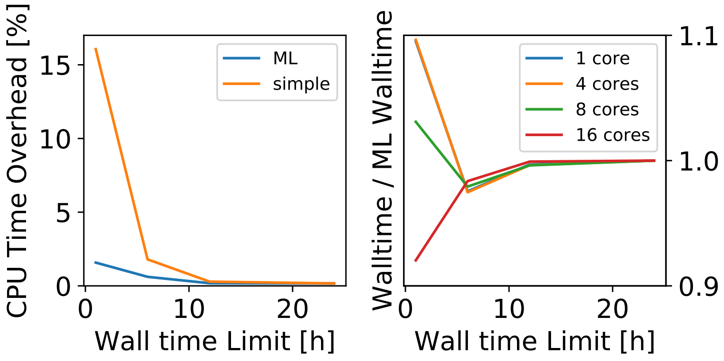


FIGURE 4.10: CPU time overhead and wall time for geometry optimizations compared between the simple approach and the QML approach. See text for details of the strategies. CPU time overhead given in percent relative to the bare minimum of CPU time needed. Wall time given relative to the wall time resulting from using the QML approach. All geometry optimizations come from task $\mathbf{QM9^{GO}_{CASSCF}}$.

using the job array approach. This is more efficient for the scheduler and increases the likelihood of the job being selected by the backfiller, further shortening the wall time. Using the QML approach does not severely affect the wall time, i.e. the time-to-solution. This is largely independent of the extent of parallelization employed in the calculation (see right hand side plot in Figure 4.10). We suggest to implement an optional stop criterion in quantum chemical codes where an external command can prematurely stop the progress of the geometry optimization to be resumed in the next compute job. This change can drastically improve computational efficiency on large scale projects. Estimating the current consumption to be on the order of at least $5{\cdot}10^5$ petaFLOPS (see discussion above in section **??**) for computational chemistry and materials science this approach may lead to potentially large savings in economical cost.

4.6 Conclusion

We have shown that the computational complexity of quantum chemistry calculations can be predicted across chemical space by QML models. First we looked at a 2D non-linear toy system consisting of example functions which are known to be difficult to optimize. Using these test functions and three optimizers, we build a first ML model and the learning curves show that it is possible to learn the number of optimization steps using only the starting position (x, y). Representations are designed to efficiently cover all relevant dimension in the given chemical space. Hence, if the computational cost is learnable by QML models, it is a reasonably smooth function in the variety of chemical spaces that we considered. This is a fundamental result.

Our approach succeeds in estimating realistic timings of a broad variety of representative calculations commonly used in quantum chemistry work-flows: single-point calculations, geometry optimizations, and transition state searches with very different levels of theory and basis sets. The machine learning performance depends on the quantum chemistry method and on the type of computational cost that is learned (FLOP, CPU, wall time). While the accuracy of the prediction is shown to be strongly dependent on the computational method, we could typically predict the total run time with an accuracy between 2% and 30%.

Exploiting QML out-of-sample predictions, we have demonstrably used compute clusters more efficiently by reordering jobs rather than blindly assuming all calculations of one kind to fit into the same time window. Without significant changes in the time-to-solution, we reduced the CPU time overhead by 10% to 90% depending on the task. With the scheme presented in this work, compute resource usage can be significantly optimized for large scale chemical space compute campaigns. To support this case, all relevant code, data, and a simple-to-use interface is made available to the community online[144].

We believe that our findings are important since it is not obvious that established QML models, designed for estimating physical observables, are also applicable to more implicit quantities such as computational cost.

Chapter 5

Quantum Chemistry: E2 vs. S_N2

Thousands of reactants and transition states for competing E2 and S2 reactions

G.F. von Rudorff, S. Heinen, M. Bragato, O.A. von Lilienfeld; Machine Learning: Science and Technology, 1, (4), 045026

5.1 abstract

Reaction barriers are a crucial ingredient for first principles based computational retro-synthesis efforts as well as for comprehensive reactivity assessments throughout chemical compound space. While extensive databases of experimental results exist, modern quantum machine learning applications require atomistic details which can only be obtained from quantum chemistry protocols. For competing E2 and S_N2 reaction channels we report 4,466 transition state and 143,200 reactant complex geometries and energies at respective MP2/6-311G(d) and single point DF-LCCSD/cc-pVTZ level of theory covering the chemical compound space spanned by the substituents NO_2, CN, CH_3, and NH_2 and early halogens (F, Cl, Br) as nucleophiles and leaving groups. Reactants are chosen such that the activation energy of the competing E2 and S_N2 reactions are of comparable magnitude. The correct concerted motion for each of the one-step reactions has been validated for all transition states. We demonstrate how quantum machine learning models can support data set extension, and discuss the distribution of key internal coordinates of the transition states.

5.2 Introduction

Reactions are the very core of chemistry and their understanding is crucial for molecular design problems: Even if a compound has been identified to be interesting for a certain application, a reaction pathway has to be found to connect abundant compounds to the desired target molecule. Large experimental databases of reaction paths with associated barriers and yields have been compiled to that end[145] and have been proven to be useful in the design of reaction steps[146,147] or for the optimization of reaction environments[148].

These databases however, rely on careful experimental work and would benefit from a computational perspective, since their extension relies on manual work. As a consequence, they are of limited detail and size when compared to chemical space. High-throughput calculations are one way of obtaining reaction paths, but pose another complex problem: Finding the relevant transition state geometries is technically difficult, in particular if the reaction pathway is not known beforehand, since it requires finding the saddle points on the potential energy surface[149–151]. As a consequence, previous computational work reporting on transition state configurations covered only a modest number of cases, and employed a wide range of levels of theory[152–161]. Additionally, an accurate representation of the Minimum Energy Path requires knowledge of the conformational space spanned by the reactant and products, a challenging task by itself[162,163]. Furthermore, not all established quantum chemistry methods are suitable for yielding accurate potential energies of reactive processes[152,162]. Direct comparison of calculated energy barriers to experiment in itself is often impracticable since the relevant barriers require the calculation of ensemble-averaged free energies in explicit solvent. This task on its own is already challenging just for a single molecule[164] and might be computationally prohibitive for large numbers of reactions. In the reverse picture, gas-phase reaction experiments are particularly challenging but possible in some cases.[165,166]

With recent successes of machine learning models in the context of exploration of chemical space[93] e.g. non-covalent interactions[167], response properties[168], and molecular forces[169], it would be desirable to also explore reaction space. Some initial work in this direction has been done already[170–178]. For any machine learning approach, consistent data sets are of high value for training and validation. Typically, a single study in literature gives about five (experimental) to fifty (computational) transition state geometries or energies. This is insufficient for the training of converged and meaningful quantum machine learning models. Furthermore, atomistic details (geometries) are often lacking in the case of experimental data, while level of theory used in the case of theoretical studies can often no longer be considered to be state of the art. While it is possible to merge reaction data from different sources or to learn their respective differences in the potential energy surface by means of Delta machine learning (Δ-ML)[137], multi-fidelity machine learning models[179], multi-level combination grid technique[179] or transfer learning[180], the resulting multilevel approaches require at least part of the data to be evaluated in many different sources. Thus there is considerable need for one large consistent data set which subsequently could be used as a basis for multilevel machine learning models and their application in reaction design. When assessing possible reactions from a given reactant, it is not always sufficient to be able to quantify just one particular pathway. Rather, several competing reaction channels need to be estimated at the same time to decide which reactions will occur with which weight. To enable such modeling, a homogeneous data set for competing reactions is desirable.

Starting from the lowest lying conformers of the organic molecules listed in the GDB-7[181] data set, Grambow et al[16] have just recently generated 12k transition state geometries using the double-hybrid ωB97X-D3 density functional approximation, allowing for any feasible reaction mechanisms. In contrast, we here focus on the narrow reaction space obtained for typical substitutions and attacking and leaving groups of the competing textbook reactions E2 and S_N2 with the specific intent to enable more thorough, systematic and comprehensive explorations of the nature of the corresponding chemical compound space. Often, S_N2 was used as a benchmark reaction due to its iconic, well established mechanism[182–186], and having the advantage of a less complex transition state over its competing reaction E2[187]. Even though the overall reaction mechanisms are well understood, their competition in terms of exploring the chemical compound space defined by specific combinations of substituents, leaving groups, and nucleophiles has not yet been studied in a systematic manner—to the best of our knowledge.

We include geometries of reactant and product conformers, reactant complexes, and transition state geometries. For our calculations, we chose the MP2/6-311G(d)[188–192] level of theory since benchmark studies have found this level to be a good compromise between accuracy and computational effort for the reactions under investigation in particular with regards to geometries[152,193,194]. DFT methods have been found to exhibit significant deviations for both energies and geometries[195]. Even for hybrid functionals, it is known for a long time that their share of exact exchange should be different for reactants and for saddle points in order to yield best accuracy[196] which renders them inapplicable for activation energies. MP2 has been shown to be more accurate for saddle point geometries, all else being equal[196,197]. For e.g. nucleophilic substitution, the MP2 error in energies is nearly half the error of typical DFT methods[183]. In order to further improve on the accuracy of the MP2 energies, we also performed single-point DF-LCCSD calculations for every transition state geometry, as well as for their reactants.

We see the main use case of this data set in the context of assessing competing reactions with machine learning methods. This is key to chemical synthesis design where competing reactions could have a strong impact on the yield. With most existing data sets focused on (near-)equilibrium geometries and associated properties, the current work offers access to a larger part of potential energy surfaces. This is particularly challenging as the ideal machine learning model would only require the reaction type and reactant information to estimate the transition state geometry or its energy, since an explicit search for each transition state geometry is expensive (as shown below). This requires strategies to estimate a property at a different point of the potential energy surface than the explicit query configuration. To develop such strategies, this data set might prove particularly useful. Moreover, the reaction data set is directly applicable to cases where the low ambient temperature renders the potential energy dominant for reaction barriers, e.g. interstellar environments. For these cases, a list of potential reactions taking place can be derived directly from the activation energy data in this work.

5.3 Methods and Computational Details

In our database, we have considered all 7,500 reactant molecules that can be built from ethane with the substituents listed in Table 5.1 using the positions shown in Figure 5.1.

	A	**B**	**C**	**D**	**E**
R*k*	H	NO_2	CN	CH_3	NH_2
X	F	Cl	Br		
Y	H	F	Cl	Br	

TABLE 5.1: Chemical space for our reaction database: substituents R, leaving groups X and the nucleophiles Y^-. Molecular skeleton is ethane, see also Figure 5.1. The letters refer to the labels in our data set files.

These substituents were selected for their following properties: i) electronic effects should be maximized and ii) steric hindrance minimized. More precisely, while being as small as possible in order to make the reaction center sufficiently accessible to the nucleophile, electron donating groups and withdrawing groups should cover weak as well as strong inductive effects.

5.3.1 Machine Learning

In this study we used delta machine learning (Δ-ML) in kernel ridge regression (KRR) implemented in the QMLcode[65]. Kernel based methods were introduced in the 1950s by Kriging *et al.*[133]. KRR uses as input a kernel function with the feature vector $\mathbf{x}$ to learn a mapping function to a property $y_q^{\text{est}}(\mathbf{x}_j)$ given a training set of N reference pairs $\{\mathbf{x}_i, y_i\}^N$:

$$y_q^{\text{est}}(\mathbf{x}_j) = \sum_i^N \alpha_i k(\mathbf{x}_i, \mathbf{x}_j) \tag{5.1}$$

where α is the regularization coefficient and $k(\mathbf{x}_i, \mathbf{x}_j)$ a gaussian kernel element:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\sigma^2}\right) \tag{5.2}$$

A more detailed discussion of the KRR method employed in this work and pertinent references can be found in Heinen *et al.*[2]. In the context of Δ-ML, the procedure stays the same and only the property (y) changes from a molecular property to a difference in properties, e.g. from $y^{\text{est}} \widehat{=} E_{\text{a}}$ to $y^{\text{est}} \widehat{=} \Delta E_{\text{a}}$.

The feature vector or representation $\mathbf{x}$ we used is one-hot encoding[68], which is a bit vector. For every substitution site R*k*, nucleophile Y and leaving group X, we denote presence of a given combination with ones. In our case, this means that for any transition state, six out of the 27 entries of the representation vector are ones, the rest zeros.

5.3.2 Reactants and Products

We started from the unsubstituted case fluoroethane optimized with openbabel[198] using the universal force field (UFF)[199] and functionalized the substituent sites R*k* in Figure 5.1 using the C++ interface of openbabel. Again, each resulting structure

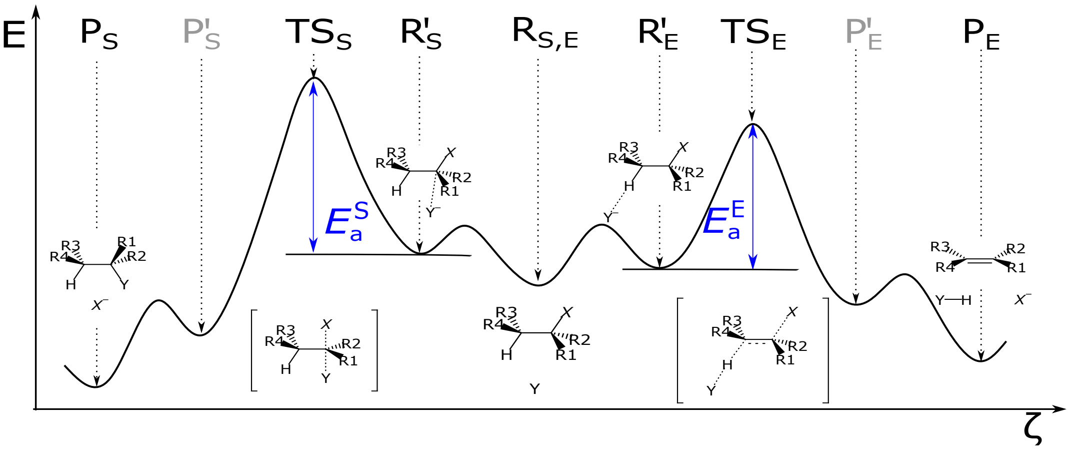


FIGURE 5.1: Energy diagram of the competing E2 and S_N2 reactions, exemplifying kinetic vs. thermodynamic control, respectively. Reactant conformers ($R_{S,E}$) are shared between the reactions, while transition states ($TS_{S/E}$), product conformers ($P_{S/E}$), reactant complexes ($R'_{S/E}$) ,and product complexes ($P'_{S/E}$) are specific to each reaction. For each reaction, the energy difference between transition state and reactant complex is the activation energy $E_a^{S/E}$.

was optimized with UFF to remove potential bad contacts. Using the Experimental-Torsion Knowledge Distance Geometry (ETKDG) method as implemented in RDKit[200], we searched for 1,000 conformer geometries. They subsequently were ordered by UFF energy. Starting from the most stable conformer, all those configurations were included in the followings steps if and only if their root mean squared difference (RMSD) to the previously accepted configuration was at least 0.01 Å or the energy difference between the two was at least 0.1 kcal/mol.

The resulting conformer candidate configurations were relaxed at MP2/6-311G(d) level with ORCA 4.0.1[188–190,201–203] to be compatible with the level of theory to be employed for the transition state search. For each of these minimized configurations, all possible nucleophiles given in Table 5.1 were placed along the expected axis of the CH bond in Figure 5.3. With the nucleophile being constrained to that axis, the geometries were optimized to obtain an estimate of the reactant complex geometry.

For each of these reactant complexes, we subsequently lifted the constraint and relaxed further. This was helpful as the potential energy landscape around the reactant complex is comparably shallow and therefore direct optimization to the free reactant complex was often ineffective.

Each unconstrained reactant complex was validated using a variety of geometrical criteria to ensure that the more than 100,000 minimum energy geometries represented meaningful configurations. The overall procedure is shown in Figure 5.2. First, we required the reactant complex to constitute two fragments based on the topology obtained from MDAnalysis[204] where one fragment needed to be of exactly one atom, i.e. the nucleophile. This is to avoid erroneous fragmentation where e.g. a proton is abstracted from the reactant. In the case of E2 reactions, we required that the angle C–H$\cdots$Y must not be smaller than 178 degrees since configurations with larger angles indicate trapping of the nucleophile by other hydrogen atoms of the reactant not involved in this particular reaction channel.

For S_N2, more validations are required.

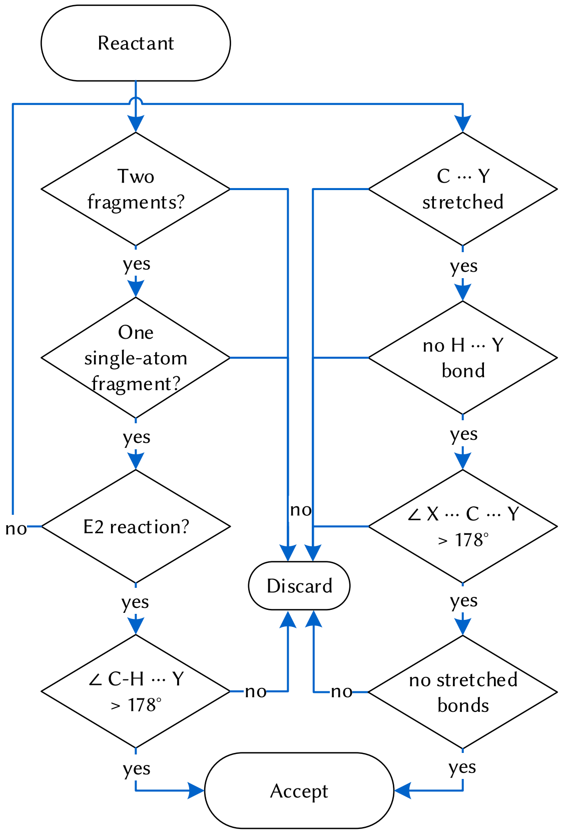


FIGURE 5.2: Validation procedure for reactant geometries starting from a candidate reactant structure to the decisions whether to accept or discard that candidate geometry.

- The C$\cdots$Y distance had to be at least 1.14 Å, 1.41 Å, 1.86 Å, and 2.04 Å for hydrogen, fluorine, chlorine, and bromine, respectively. This avoids configurations that are actually product complexes. Due to the low activation energy for many such cases, a geometry optimization can end up in a product complex minimum from a reactant complex initial guess.
- To avoid trapping of the nucleophile by reactant hydrogen atoms, the distance between the nucleophile and the closest hydrogen of the reactant is required to be at least 0.78 Å, 0.96 Å, 1.33 Å, and 2.48 Å for hydrogen, fluorine, chlorine, and bromine, respectively.
- Since the S_N2 reaction requires nearly planar bonds for the reaction center, we require that the angle X$\cdots$C$\cdots$Y must be at least 178 degrees.
- We avoid artificially stretched geometries by requiring no carbon-carbon distance to be within 1.65–2 Å and no nitrogen-oxygen distance to be within 1.5–2.5 Å.

Whenever these validation steps were successful, the lowest such minimum from all conformers investigated is considered to represent the reactant complex. Otherwise, the lowest energy configuration from the constrained optimization is taken as an approximation of the reactant complex. In the latter case, Δ-ML[137] was employed to estimate the residual relaxation energy between the constrained and unconstrained reactant complex.

Duplicate reactant and product conformer geometries were identified using the FCHL19[169] representation. By that measure, only unique geometries are retained. This test was not applied to reactant complexes as their local minima energies and geometries can be very similar yet distinct.

5.3.3 Transition States

Using Gaussian09[205] for an initial transition state geometry with B3LYP/6-31G*[125,126,206–209] and subsequently ORCA 4.0.1 for a final transition state with MP2, we first found the transition state of the unsubstituted case with chloride as nucleophile. Functionalization followed the same procedure as for the reactants. Using these starting geometries, transition states were obtained via eigen mode following as implemented in ORCA. After a transition state was found, the local Hessian matrix was obtained from a numerical frequency calculation by finite displacements as implemented in ORCA.

Once a transition state was found for a combination of the four substituents, this geometry was employed as starting geometry for further transition state searches for missing cases where exactly one out of the four substituents was different from the case where a validated transition state has been found. This scheme was used only for those molecules where the substituent that was to be replaced did not have the same functional group as the neighbouring substituent on the same carbon atom. For some cases, this procedure was employed several times in a row, each time resulting in an additional set of transition states which served as starting guesses. Similarly, the nucleophile of validated transition states was replaced to obtain promising starting geometries for the transition state search.

Once the transition state geometry has been found for any potential reaction target, the Hessian was evaluated to ascertain that the geometry in fact is a transition state with exactly one imaginary frequency. We only included a transition state in our data set if this frequency was at least 400 cm^{-1} and that the resulting motion corresponding with this one normal mode was as shown in Figure 5.3 (left column). The ethane skeleton features two carbon atoms C*k*, where the one with substituents R1 and R2 is numbered C1. For the E2 transition state, X, Y and the hydrogen atom were displaced along the normal mode and checked if the distances C2-H as well as C1-X were larger and the C2-Y distance was smaller compared to the non-displaced geometry. In the S_N2 transition state, the nucleophile and leaving group were displaced along the normal mode and C1-X was compared to C1-Y.

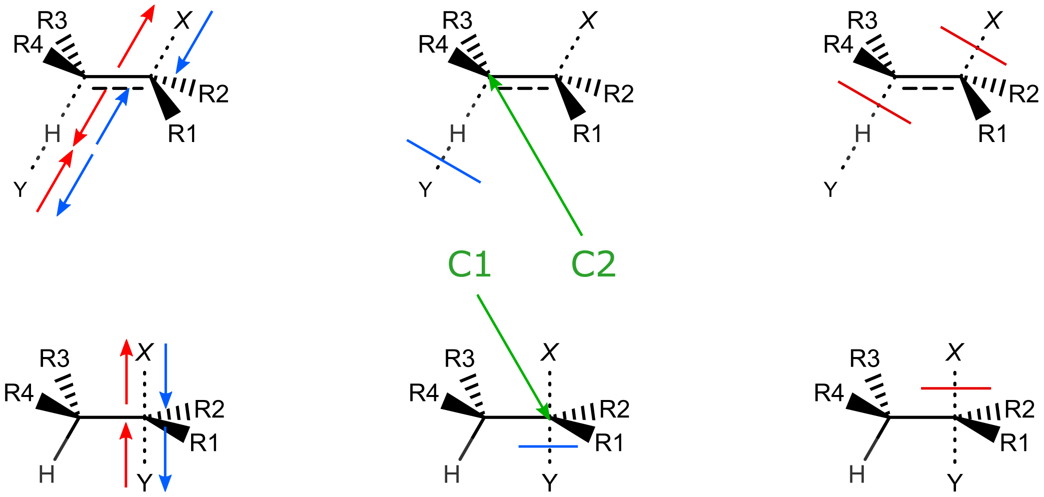


FIGURE 5.3: Illustration of validation procedures for generating E2 (top) and S_N2 (bottom) geometries. Normal mode requirements for transition states (left column) show concerted motions which are characteristic for the reaction in question (red arrows point towards product, blue arrows towards reactant). Bond cleavages tested for reactant complexes and product complexes are shown in the mid and right column, respectively. Blue perpendicular lines correspond to removal of Y^-, YH and X^- for E2 and X^- or Y^- for S_N2 leading to infinite separation as shown in Figure 5.1). Bond cleavage indicated by red perpendicular lines corresponds to product formation.

While the investigation of the normal modes alone ensures that the vibrational motion belongs to the main configurational change the molecule undergoes during each reaction, it is not a sufficient criterion that this particular transition state geometry actually connects reactant and product. We use the intrinsic reaction coordinate (IRC)[210] as final criterion to ensure that the transition state indeed connects a valid reactant complex with a valid product for the reaction in question. The IRC is commonly employed to find a reaction pathway starting from a transition state. The Cartesian IRC is given by the steepest descent path in forward and backward direction of the reaction. We use steepest descent as implemented in ORCA to trace the Cartesian IRC. If the energy curvature near the transition state and along the reaction coordinate is small, steepest descent paths can become subject to numerical instabilities. To avoid this issue, we approximate the IRC close to the transition state by a line scan in either direction based on the normal mode displacement of imaginary frequency. From the final point of the line scan, a regular steepest descent is followed until a local minimum has been reached.

Since the sign of the normal mode of imaginary frequency is not fixed with respect to the direction of the reaction, we analyze the minimum energy endpoints of the IRC to classify them as either close to reactant or close to product based on the bond length as shown in Figure 5.3. If and only if exactly one of the endpoints is found to be close in geometry to a reactant configuration and the other is found to be close in geometry to the product configuration, the corresponding transition state is included in our data set. To test whether the configurations are close to reactant or product, we measured C2-H distances for the E2 case and C1-X and C1-Y distances for the S_N2 reaction to ensure bonds have been broken as shown in Figure 5.3.

For cases where several validated transition states for the same reaction have been found, we consider the lowest one for the reaction barrier.

Finally, we performed single-point DF-LCCSD/cc-pVTZ calculations, as implemented in Molpro2018[211–217] using the extremal geometries as obtained with MP2/6-311G(d). All in all, the complete generation of the data set took about 2.8 million core hours.

5.4 Results

5.4.1 Data

Our resulting data set contains 4,466 validated transition state geometries, of which 2,785 are for S_N2 (TS_S) and 1,681 for E2 (TS_E). Based on 26,997 reactant conformers ($R_{S,E}$), we identified 81,950 constrained reactant complexes for E2 (R'_E) and 57,642 constrained reactant complexes for S_N2 (R'_S) which in turn have been refined to yield 2,030 unconstrained reactant complexes for E2 (R'_E) and 1,532 unconstrained reactant complexes for S_N2 (R'_S). Finally, we have found 15,706 S_N2 product conformers (P_S) and 9,588 E2 product conformers (P_E). All geometries are calculated at MP2/6-311G(d) level of theory and given as XYZ files in this work. Two additional files specify all individual energies and activation energies, respectively. The labels in the text files relate to the labels in table 1.
All data is available in the materials cloud (https://doi.org/10.24435/materialscloud:sf-tz).

5.4.2 Geometries

As shown in Figure 5.4, we were able to find many transition states for a variety of substituents, nucleophiles and leaving groups. This means that we have reached a substantial coverage of the chemical space in question, which is key for machine learning. The challenge here is the low success rate of the transition state search which might have been the key reason why such data sets have not yet been published earlier. In particular, machine learning models will benefit from the comparably low noise in the data set coming from our validation procedure. Moreover, the data set features many different combinations of substituents such that there is considerable promise that their interplay for the competing reactions can be analysed and understood.

As in any iterative optimization scheme, convergence thresholds influence the final results. This is the case for a transition state search as well and might potentially give rise to some small noise in the transition state geometries. Since we calculated the explicit Hessian matrix, we know that the transition state geometries reported in this data set are indeed saddle points, and that their mass-weighted normal modes

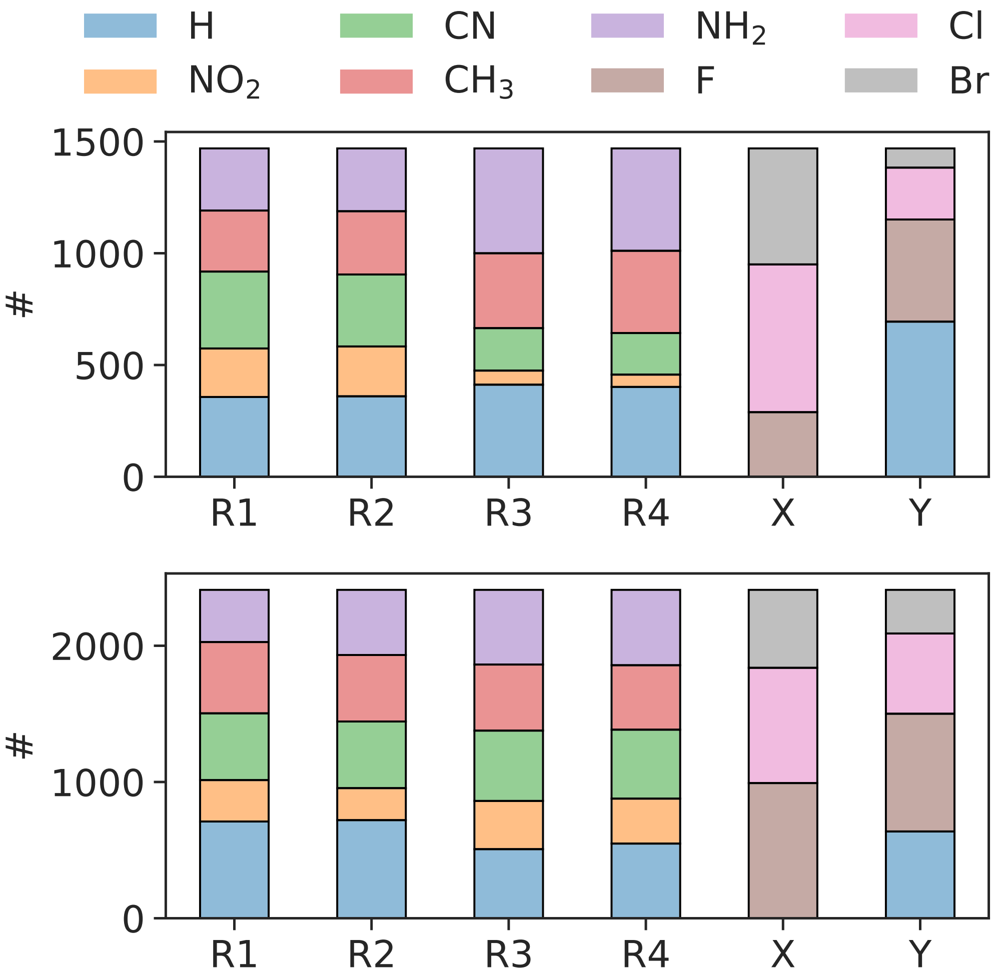


FIGURE 5.4: Distribution of substituents R*k*, leaving groups X, and nucleophiles Y for all activation energies in the data set. Top: distribution for E2, bottom: distribution for S_N2.

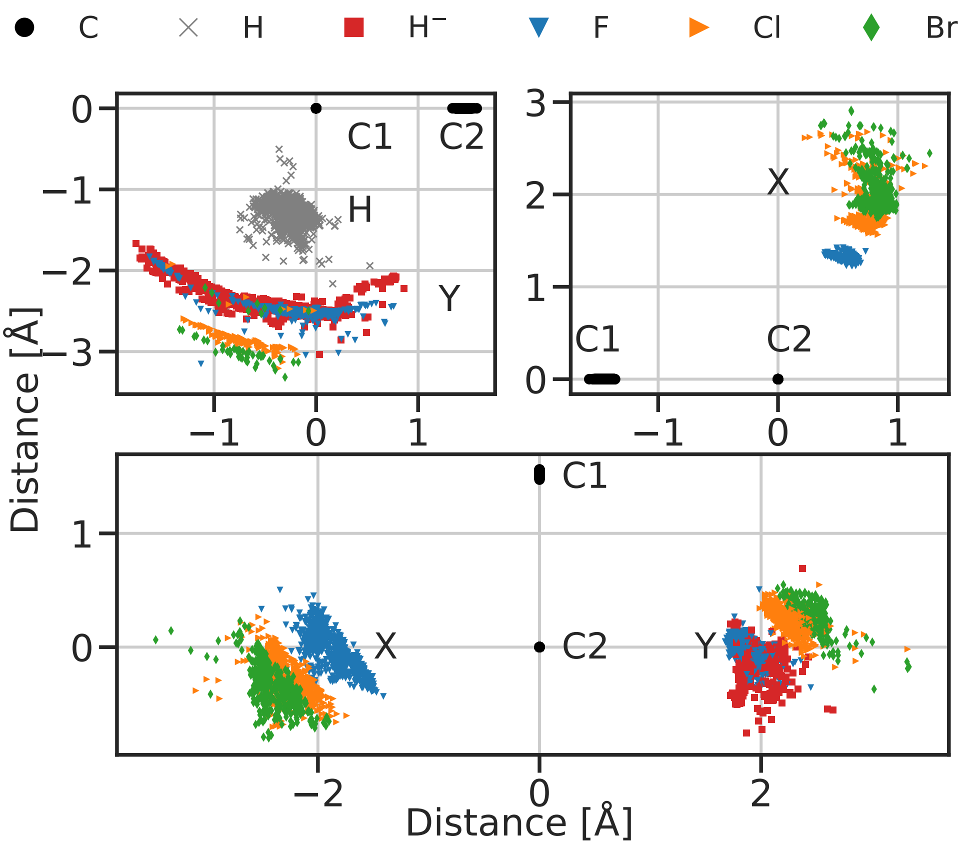


FIGURE 5.5: Overview scatter plot of atomic positions of scaffold carbon atoms and nucleophiles and leaving groups for all transition states for E2 (top row) and S_N2 (bottom). Transition state geometries have been translated such that C1 (top left) or C2 (top right and bottom) are in the origin. Additionally, they have been rotated such that all atoms shown with the exception of the hydrogen sites in the top left panel are planar. Coordinates of other atoms have been projected into the figure plane.

represent the concerted rearrangement expected for E2 and S_N2 reactions. Together with the tight convergence criteria required for transition state optimization, this means that our data set contains only highly compatible transition state geometries for all the validated combinations of substituents, nucleophiles and leaving groups.

This is demonstrated in Figure 5.5 which shows a scatter plot of the most important internal coordinates for the transition states. The reduction of dimensionality from the more complex 3D geometry is obtained by placing one of the two central carbon atoms in the origin and then aligning the carbon-carbon bond along one Cartesian axis. The other markers then show the position of one atom for each transition state found. For E2 reactions, the transition state geometry has been rotated such that all three points shown in the corresponding panels are exactly within one plane. For the S_N2 case, this is not possible, as the four atoms in question are not necessarily exactly in a plane even though they are very close to that. For this panel, the projection on the fitted plane through all four points is shown. For the internal carbon-carbon bond, the variance of the bond length is significantly higher for E2 than for S_N2, as shown in Figure 5.5. This can be explained by the nature of the two reactions: While E2 consists of a concerted action on both carbons, S_N2 happens only at one of the two carbons. We also see that each element for the nucleophile and leaving group has its own distribution of positions relative to the two central carbon atoms. This distribution reflects the impact of the different substituents on each transition state geometry. It is interesting that fluorine atoms exhibit much less spatial variation as leaving group than other halogens for E2 while this is not at all the case for the role of fluorine as nucleophile in the very same reaction. This is likely attributed to the comparably short bond distance of fluorine for the leaving group, since in the case of the nucleophile this distance is increased due to one intermediate hydrogen atom between the central carbon and the nucleophile. The reduced distance in the former case then would lead to a more pronounced Coulombic interaction with the molecule, effectively restraining the fluorine atom to a smaller volume of configurational space.

The centers of the positional distributions of the three halogens as leaving group increase with the period of the element, which is in line with typical bond radii for these elements. This is more pronounced in the case of the nucleophiles in E2 reactions where the intermediate hydrogen atom reduces the interaction between nucleophile and molecule. The result is that the nucleophile positions are spread out on arcs around the central carbon with most of the positional freedom captured by the intermediate hydrogen atom. Again, the radii of the halogen arcs follow the period of the elements, while a hydrogen as nucleophile is most flexible in regards to its distance from the central carbon.

For the distribution of internal coordinates for the S_N2 reaction in Figure 5.5, two features are most striking: the triangular domain of the positions of halogenic nucleophiles and the bimodal distribution of hydrogens in the same case which in turn is mirrored in a bimodal distribution for the leaving group positions for all elements.

The triangular domain for halogenic nucleophiles in Figure 5.5 can be explained by their electrostatic interaction with the reactant molecule in gas phase. For the transition state to be a saddle point, all but one degrees of freedom must yield an increase of energy. At the tip of the triangular domain, there are three bounds to observe. First, if the distance to the carbon forming the reaction center would decrease, then the binding energy gain would become dominant, so this distance needs to be slightly above the equilibrium bond length. Secondly, the direction towards the

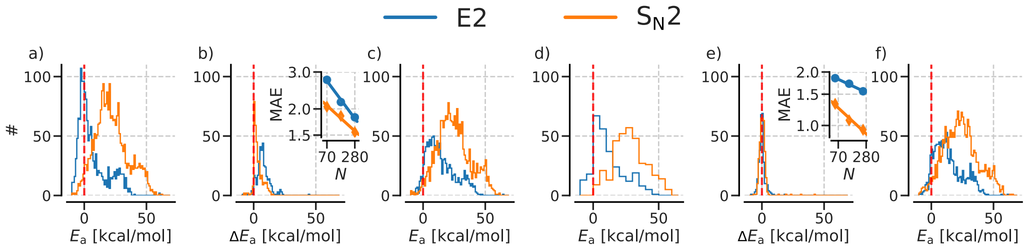


FIGURE 5.6: a) MP2 energies of constrained geometries. b) ΔE_a of constrained to unconstrained geometries as obtained from quantum chemistry calculations (training data for the ML models). Inset: Learning curves for the Δ-ML models (constrained to unconstrained energies) illustrating test errors (MAE in kcal/mol) vs training set size (N). c) ML shifted MP2 energies. d) LCCSD energies on unconstrained MP2 geometries. e) ΔE_a of MP2 to LCCSD energies as obtained from quantum chemistry calculations (training data for the ML models). Inset: Learning curves for the Δ-ML models (MP2 to LCCSD energies) illustrating test errors (MAE in kcal/mol) vs training set size (N). f) ML shifted LCCSD energies.

planar substituents R1 and R2 would reduce the distance between the partially negatively charged nucleophiles and the partially positively charged hydrogen atoms of the substituents. This Coulombic interaction is more pronounced in gas-phase and restricts the possible geometries for transition states in this direction. Finally, pushing more towards the other carbon atom of the reactant skeleton (upwards in Figure 5.5), would be unfavourable in the sp^2 hybridisation of the reaction center. Only for larger distances of the nucleophiles to the reaction center, deviations from the last two constraining factors become possible, hence the triangular shape of the domain for each element.

The bimodal distribution of the hydrogen nucleophiles for S_N2 as shown in Figure 5.5 correlates with the leaving group in the corresponding reaction. Only leaving groups of chlorine and bromine allow a distance C2-H larger than 2 Å. This could be linked to the substantially higher electronegativity of fluorine, pulling more of the C2 electron cloud towards the leaving group, allowing for a shorter distance to the H^-.

Results such as the triangular domains and the bimodal distribution can be easily identified in large homogeneous data sets such as this one and can be interesting test cases for machine learning models for phenomena resulting from the complex interplay of competing physical interactions.

5.4.3 Energies

Based on the conformational search for the reactant geometries and the validated transition states, we could calculate activation energies for both reactions. Figure 5.6 shows the broad distribution of said activation energies which span about 50 kcal/mol. In general, E2 activation energies are lower than S_N2 activation energies. Since the activation energies are defined as the difference in energy between the transition state and the reactant complex, the nature of the reactant complex is highly relevant. This is exemplified by the significant portion of negative activation energies if we consider the constrained approximation of the reactant complex alone (panel a) in Figure 5.6).

These spurious negative activation energies result from two aspects: the finite number of conformers tested as potential reactant complex geometry and the constraint enforcing the characteristic alignment of the nucleophiles with the molecule when forming a reactant complex. To alleviate the impact of the former effect, we searched for more conformers until the number of negative activation energies could not be reduced any further despite testing of additional conformers. Here, the small size of the molecular skeleton was helpful, as only a few conformers can be realized for each molecule in our chemical space. We dealt with the second reason for negative activation energies by removing the constraint for the characteristic alignment of the nucleophiles with the molecule. This constraint was needed initially to ensure that the relaxation (described in the Methods section above) did not converge to an irrelevant reaction complex where the nucleophiles would be trapped by the partially positively charged hydrogen atoms of the substituents. Since the minimum of the reaction complex is only shallow, this initial constraint drastically improved the success rate of finding reactant complexes matching the reaction mechanism.

Relaxing the reactant complexes further without the constraint again bears the risk of the substituents trapping the nucleophiles. Consequently, many but not all reactant complexes could be refined this way: 301 and 348 targets for E2 and S_N2, respectively. We expect that turning the constraint into a restraint that subsequently is reduced during the minimization until the unconstrained minimum is found could be one route to identify the correct relaxation energy for all reactant complexes in our data set. However, this would be extremely costly and is subject to many degrees of freedom, like the speed at which the restraint is removed such that this route is not feasible for the thousands of reactant complexes we have in our data set. Therefore, we trained a one-hot-encoding KRR machine learning model to take the explicit relaxation energies we have found and to predict the relaxation energies for the remaining compounds. These relaxation energies span about 15 kcal/mol. We could machine learn the relaxation energy down to prediction errors of 1.5 and 1.8 kcal/mol (for 280 randomly chosen training instances) for two separate models for S_N2 and E2 reactions, respectively (see inset panel b) of Figure 5.6). This is much less than the expected error of the quantum chemistry method that we use, MP2. We do expect that more sophisticated machine learning methods could possibly improve upon this accuracy.

Panel b) in Figure 5.6 shows the activation energies for those barriers where we were able to find the explicit minimum geometry for the unconstrained reactant complex. The fact that this exhibits nearly no negative activation energy is in line with our observation that searching for additional conformers as basis for the reactant complex did not yield any further change to the activation energies. Using the explicitly calculated relaxed reactant complexes where available and including a machine learned relaxation energy in the activation energy for all other reactions, we obtain our final MP2/6-311G(d) and ML corrected MP2/6-311G(d) numbers for the activation energy, shown in panel c) of Figure 5.6 which now span 60 kcal/mol for S_N2 reactions and 50 kcal/mol for E2 reactions.

Comparing panels a) and c) in Figure 5.6 shows how the number of negative activation energies has been greatly reduced by removing the constraint on the reactant complex, confirming that this was the main reason for negative activation energies in the initial case of panel a). Calculating activation energies directly from the reactants at infinite separation is no substitute for this complicated procedure of correcting for the constraint impact. This is due to the significant interaction energy gained in forming the reactant complex in the E2 case where the negatively charged nucleophile approaches a hydrogen. For S_N2, in few cases the reactant complex might be higher in energy than the reactants at infinite separation.

Given the documented quality of MP2 geometries for substitution reactions[152], the main difference to higher level of theory than MP2 is expected to come from higher-quality energies for MP2 geometries. Since higher level of theory calculations are not affordable in the context of the geometry optimizations for this many configurations, additional single points on top of MP2 geometries recover at least a substantial part of the difference in the potential energy landscape. For those cases where we have both the transition state and the unconstrained reactant complex, we performed DF-LCCSD/cc-pVTZ calculations. The explicit data is shown in panel d) of Figure 5.6. The difference to the MP2 data however is more interesting and shown in panel e) of the same figure. While the distribution of the corrections is centered around zero, the typical correction is on the order of a few kcal/mol with only few substantially larger values.

Explicit calculations of the LCCSD energy are only accessible for cases where we have an explicit molecular geometry. If the unconstrained geometry optimization did not successfully find the shallow minimum of the reactant complex, then this explicit molecular geometry is not available. To extend the coverage of the LCCSD correction which improves the accuracy of the activation energy data, we built a one-hot encoding machine learning model that predicts the LCCSD energy for the missing geometries. This Δ-ML approach exhibits learning with an error of less than 1 kcal/mol (S_N2) and 1.5 kcal/mol (E2) after training on 280 instances. After this second step, we obtain our final activation energies which have a slightly broader distribution than before, shown in panel f) of Figure 5.6. It is interesting to note that this final activation energy distribution of the E2 is dramatically more skewed towards very small values than the S_N2 which appears to be more normally distributed. This could be due to the symmetry in the case of the S_N2 (as also shown in Fig. 5.5) where one covalent bond is broken as the other is formed. The E2 reaction is less symmetric, effectively breaking one single bond while forming a double bond. The structural lack of symmetry is also on display in Fig. 5.5.

We also note that the learning curves for the activation energy of E2 display a higher off-set than for S_N2 even though, the E2 data has a smaller magnitude and variance. This latter aspect could be due to some extreme outliers in the E2 data set for which values larger than 50 kcal/mol have been observed, introducing severe bias in the mean absolute error. A median error measure might be better tempered for such a data set.

Panel f) of Figure 5.6 shows some remaining negative activation energies. For S_N2, there are 43 such negative energies, all but one of which are from machine learning predictions only. For E2, there are 120 such negative energies in total, 79 of which come from machine learning predictions. Therefore, for the majority of cases the machine learning model needs improvement, possibly by adding more explicit unconstrained reactant complexes. The cases where the explicitly calculated activation energies are still negative likely come from a finite search of conformer geometries, meaning that some unconstrained reactant complex minima have not been found. In our data set, we include these negative activation energies such that future machine learning models correcting e.g. the constrained to unconstrained relaxation can test whether they improve upon our approach.

5.5 Conclusion

We present a large comprehensive data set of key geometries for the two competing E2 and S_N2 reactions. We report energies and geometries obtained in a consistent and systematic manner such that this data set can serve as a playing ground for machine learning models dealing with competing reaction channels for a broad range of substituent combinations. The substituents have been chosen to reflect a substantial chemical diversity over a wide range of electron donating and electron withdrawing effect strengths. We have used the internal consistency of the data set to discuss the distribution of structural effects in transition state geometries. This was only made possible due to the large chemical space covered by our calculations. We have shown how simple machine-learning models can be used to reduce the computational cost and to curate and extend (imputation) the data set in such high-throughput efforts. The entire data set including geometries and energies at DF-LCCSD/cc-pVTZ//MP2/6-311G(d) and MP2/6-311G(d) level of theory is available as part of this publication.

Acknowledgements

We acknowledge support by the Swiss National Science foundation (No. PP00P2_138932, 407540_167186 NFP 75 Big Data, 200021_175747, NCCR MARVEL) and from the European Research Council (ERC-CoG grant QML). This work was supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID s848. Some calculations were performed at sciCORE (http://scicore.unibas.ch/) scientific computing core facility at University of Basel.

Chapter 6

Quantum Machine learning: E2 vs. S_N2

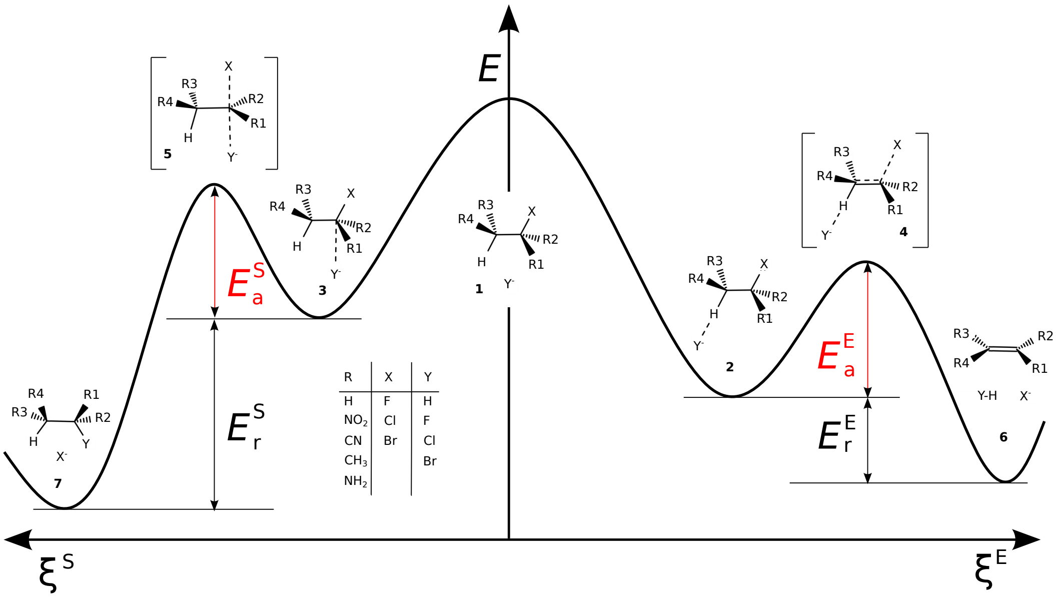
Toward the design of chemical reactions: Machine learning barriers of competing mechanisms in reactant space

S. Heinen, G.F. von Rudorff, O.A. von Lilienfeld; The Journal of Chemical Physics, 155, (6), 064105

6.1 Abstract

The interplay of kinetics and thermodynamics governs reactive processes, and their control is key in synthesis efforts. While sophisticated numerical methods for studying equilibrium states have well advanced, quantitative predictions of kinetic behavior remain challenging. We introduce a reactant-to-barrier (R2B) machine learning model that rapidly and accurately infers activation energies and transition state geometries throughout chemical compound space. R2B enjoys improving accuracy as training sets grow, and requires as input solely molecular graph of the reactant and the information of the reaction type. We provide numerical evidence for the applicability of R2B for two competing text-book reactions relevant to organic synthesis, E2 and S_N2, trained and tested on chemically diverse quantum data from literature. After training on 1k to 1.8k examples, R2B predicts activation energies on average within less than 2.5 kcal/mol with respect to Coupled-Cluster Singles Doubles (CCSD) reference within milliseconds. Principal component analysis of kernel matrices reveals the hierarchy of the multiple scales underpinning reactivity in chemical space: Nucleophiles and leaving groups, substituents, and pairwise substituent combinations correspond to systematic lowering of eigenvalues. Analysis of R2B based predictions of $\sim$11.5k E2 and S_N2 barriers in gas-phase for previously undocumented reactants indicates that on average E2 is favored in 75% of all cases, and that S_N2 becomes likely for nucleophile/leaving group corresponding to chlorine, and for substituents consisting of hydrogen or electron-withdrawing groups. Experimental reaction design from first principles is enabled thanks to R2B, which is demonstrated by the construction of decision trees. Numerical R2B based results for interatomic distances and angles of reactant and transition state geometries suggest that Hammond's postulate is applicable to S_N2, but not to E2.

6.2 Introduction



R	X	Y
H	F	H
NO_2	Cl	F
CN	Br	Cl
CH_3		Br
NH_2		

FIGURE 6.1: **Scheme for competing reactions E2 vs. S$_N$2.** Top row: Transition states E2 (**4**) and S$_N$2 (**5**). Middle row: Reactant and nucleophile at infinite separation (**1**). In gas phase the energy of the transition state often lies lower than the energy of the reactants at infinite separation[218]. Bottom row: Product geometries at infinite separation (**6** and **7**) and reactant complexes (**2** and **3**). Properties of interest for this work are activation energies E^E_a and E^S_a, reactants, reactant complexes, and transition states. Table shows substituents R, leaving groups X, and nucleophiles Y.

To accelerate robotic experimental materials synthesis, design, and discovery[219,220] a reliable operating system is necessary which can deploy robust virtual models of alternative chemical reaction channels. Rapid yet accurate predictions of the kinetic control of reaction outcomes for given reactants and competing reaction channels, however, are still an unsolved problem. Considerable efforts in quantum chemistry were already directed at the development of automated transition state (TS) searches and chemical reaction paths. However, calculation of the relevant parts of potential energy surfaces remains a difficult challenge under active research[221]. To this end, many TS search algorithms have been introduced which can be grouped into single or double ended methods[222,223]. An example of the former is the single-ended growing string method[224], which uses only the reactant as starting point and then searches minimum energy paths and transition states. Double-ended methods such as nudged elastic band[149,225] or the two-sided growing string method[226] employ both reactant and product geometries, to obtain a TS geometry. While successful, both approaches are computationally demanding, and in practice often limited to small systems with mostly single step reactions[16]. Recent advances in synthesis planning and modern machine learning techniques hold the promise for dramatic acceleration of such numerical challenges[227,228]. Already several artificial neural networks to predict reaction outcomes were introduced (see[9] for a recent review), including work based on molecular orbital interactions of reactive sites[10], molecular fingerprints (template based)[11], reaction site identifiers (template free)[12,13], scoring functions in search trees[14], sequence to sequence maps[15], and multiple fingerprint features[229]. However, all these machine learning models rely on experimental records, meaning that they are agnostic of the underlying kinetics which are known to be crucial for reliably predicting reaction outcomes. Neglecting the energetics of chemical reactivity can be problematic, however, due to the reaction rate's exponential dependency on the activation energy (cf. Arrhenius equation).

To use machine learning to go beyond experimental data records and towards more reliable virtual predictions of reaction outcomes for new chemistries, reaction conditions, catalysts, or solvents, access to substantial and systematic relevant training data of fundamental energetics, e.g. encoding kinetic or thermodynamic effects, is required[230]. Very recent first steps in the direction of quantum machine learning applied to reactivity included the prediction of H_2 activation barriers of Vaska's complexes[231], the effect of nucleophilic aromatic substitution to reaction barriers[232], the temperature dependency of coupled reaction rates[233], or the prediction of enantioselectivity in organocatalysts[234].

In this work, we demonstrate how the reactant-to-barrier (R2B) model effectively unifies the two directions (yield vs. energy) in order to deliver robust predictions of reaction outcomes of competing mechanisms. We show how R2B can be used to predict and discriminate competing reaction channels among two of the most famous text book reactions in chemistry, S_N2 vs. E2[235] (See Fig. 7.1) using a quantum data set from the literature encoding thousands of transition states obtained from high-level quantum chemistry[3]. Using our R2B model, we complete the data set for undocumented combinations for which transition state optimizers did not converge. We also demonstrate how decision trees based on R2B give actionable suggestions for experiments on how to control which reaction channel dominates, and thus the reaction outcome. On the synthetic chemistry side, an analysis of the predicted activation energies, as well as transition state and reactant complex geometries based on our models suggests that Hammond's postulate is not applicable for E2.

6.3 Methods

6.3.1 Kernel Ridge Regression

Ridge regression belongs to the family of supervised learning methods where the input space is mapped to a feature space within which fitting is performed. The transformation to the feature space is unknown *a priori* and computationally expensive. To circumvent this problem, the "kernel trick"[68] is applied where the inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ of the representations of the two compounds i and j are replaced by the so-called kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$. This results in kernel ridge regression (KRR). A kernel is a measurement of similarity between two input vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. In this work, we used the Gaussian kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\sigma^2}\right) \tag{6.1}$$

with the length scale hyperparameter σ and representation $\mathbf{x}$. Using the representation of a molecule as input space, KRR learns a mapping function to a property $y_q^{\mathrm{est}}(\mathbf{x}_q)$, given a training set of N reference pairs $(\mathbf{x}_i, y_i)$. The representation FCHL19 was optimized for the Gaussian kernel and currently represents state of the art for energy predictions within KRR based ML models. The property $y_q^{\text{est}}(\mathbf{x}_q)$ can be expanded in a kernel-basis set series centered on all the N training instances i,

$$y_q^{\text{est}}(\mathbf{x}_q) = \sum_i^N \alpha_i k(\mathbf{x}_i, \mathbf{x}_q) \tag{6.2}$$

where $\{\alpha_i\}$ is the set of regression coefficients which can be obtained as follows:

$$\alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y} \tag{6.3}$$

with the regularization strength λ, the identity matrix $\mathbf{I}$ and the kernel matrix $\mathbf{K}$ with kernel elements $k(\mathbf{x}_i, \mathbf{x}_j)$ for all training compounds. The kernel ($\mathbf{K}$) within a representation stays the same for both reactions and the difference in the R2B models (α) enters in the change of the label ($\mathbf{y}$)[236].

6.3.2 Representations

Here, we have selected four representations of varying complexity: the Bag of Bonds (BoB)[56], spectrum of London[237] and Axilrod-Teller-Muto[58,59] potentials (SLATM), FCHL19[60] and one-hot encoding[68].

BoB uses the nuclear Coulomb repulsion terms from the Coulomb matrix representation (CM[55]), and groups them into different bins (so-called bags) for all the different elemental atom pair combinations. SLATM[136] uses London dispersion contributions as two body term (rather than coulomb repulsion) and Axilrod-Teller-Muto potential as three body term. While the FCHL18 parameterization accounts for one-body effects in terms of the position of the element in the periodic table (group and period)[61], FCHL19 limits itself to two- and three-body terms for the sake of computational efficiency[60]. Its two-body terms contain interatomic distances R scaled by R^{-4}, and the three-body terms account for the angular information among all atom triples scaled by R^{-2}.

All three geometry-based representations have been tested extensively on close-to-equilibrium structures. Since reactive processes, by definition, deal with out of equilibrium structures, we have also included a simple geometry free representation, namely one-hot encoding. This representation has also been used to encode amino acids in peptides for artificial neural networks[238,239]. In one-hot encoding, the representation is a vector of zeros and ones (i.e. a bit vector), where only one entry is non zero per feature. To describe the molecules, we used a bit vector for every substitution site (R$i \in \{1,2,3,4\}$, and one for the nucleophiles (Y) and the leaving group (X), respectively. This results in a combined vector containing 6 bit vectors of total length of 27 bits.

6.3.3 Training & Testing: Learning curves

To train our R2B models, the data set was split into a training set and a test set to optimize the hyperparameters and evaluate the model, respectively. To get the optimal hyperparameters, we used k-fold cross validation[68]. We divide the training data into k folds and for each fold, we trained on all but one fold which was used for evaluating the model. This procedure was done in an iterative fashion over all the folds. We then calculated the averaged error over these folds. This was done for different combinations of hyperparameters σ and λ.

The input for all the geometry based R2B models was the reactants at infinite separation (Figure 7.1 compound **1**). For each reaction, different reactant conformers (yielding different reactant complexes, Figure 7.1 compound **2** and **3**) have been reported in the data set[3]. To obtain a uniquely defined problem for the ML models, we canonicalized the reactant complexes by always choosing the lowest-lying one from the source data base. Using compound **1** the kernel for both reaction channels is the same (K^{tot}), which contains 2 kernels: one for the molecule (M and M') and one for the attacking group (Y and Y') as shown in equation 4. Therefore, for both reactions, the same kernel can be used, and the difference in the training enters by the activation energy ($\mathbf{y}$) in equation 3.

$$K^{\text{tot}} = K(Y, Y') \circ K(M, M') \tag{6.4}$$

Since one-hot encoding does not depend on the geometry, the kernel can be calculated directly for the entire system.

In order to measure the accuracy of our R2B models, we picked the best set of hyperparameters and trained the model using different training set sizes N and plotted the mean absolute errors (MAE) vs. N (in a log-log plot), resulting in learning curves. Using learning curves allowed us to see the learning behavior of our R2B models and compare different representations. The error ϵ of a consistently improving ML model should decrease following a power law for increasing training set sizes N[41], in a logarithmic scale:

$$\log(\epsilon) = \log(a) - b \cdot \log(N) + ... \tag{6.5}$$

where a is the offset (an indicator of how well the selected basis functions fit reality) and b the slope of the learning curve which describes the speed of which the accuracy increases using larger training set sizes. Higher order terms (...) were neglected in this work, as commonly done.

6.3.4 Data & Scripts

The data extracted from QMrxn20[3] are available on github[240]. The scripts used to optimize the hyperparameters and to generate the learning curves are also available in the same git repository.

The data set QMrxn20[3] contains 1,286 E2 and 2,361 S_N2 machine learned LCCSD activation barriers (ΔE_a). From these reactions, 529 are overlapping reactions, meaning they start from the same reactant (**1**) and go over different reactant complexes (E2: **2** and S_N2: **3**) towards the corresponding transition states (E2: **4** and S_N2: **5**). All geometries in the data set had been optimized with MP2/6-311G(d)[188–192] and subsequently DF-LCCSD/cc-TZVP single point calculations (as implemented in Molpro2018) were performed[211–217]. The backbone scaffold of all reactants is an ethane molecule which is substituted by functional groups and a leaving group. The system also contains the nucleophile (attacking group). The chemical composition of the reactant complexes is shown in the table in Figure 1 and contain the functional groups -H, NO_2, -CN, CH_3, and NH_2, the leaving groups -F, -Cl, and -Br, as well as the nucleophiles H^-, F^-, Cl^-, and Br^-. The molecular system (e.g. the reactant complex) is negatively charged and contains at most 21 atoms (including hydrogens) or 16 heavy atoms (non hydrogen atoms). To ensure the data source[3] did not contain duplicated reactions, we verified this question by calculating the L2 norm of all pairwise differences between training and test compounds of the corresponding

FCHL19 representations and identified only three out of the 3,647 cases where that norm is very close to zero. We have inspected these 3 cases and they correspond to systems which only differ in the location of the same set of substituents. As such, they are distinct but are, due to their similarity, mapped to very similar regions in feature space. In any case, since they amount to only less than one per mille of the datapoints, we chose to work on the original data set for better comparison to literature.

6.4 Results and discussion

6.4.1 Learning Barriers

Conventionally, the first principles based prediction of activation energies requires the use of sophisticated search-algorithms which iteratively converge towards relevant transition state geometries which satisfy the potential energy saddle-point criterion[151,225,226]. The activation energy is then obtained as the energy differences between reactant and transition state geometry. By contrast, our R2B models solely rely on reactant information as input. We trained them using aforementioned geometry based representations BoB[56], SLATM[136], FCHL19[60], as well as one-hot-encoding, to predict activation energies solely based on reactants at infinite separation as input geometries (compound **1** in Figure 7.1). Resulting learning curves in Figure 6.2 indicate systematically improving activation energy predictions with increasing training set size N for E2 and S_N2. For both mechanisms, the most data-efficient R2B models (one-hot-encoding) reach prediction errors of 3 kcal/mol with respect to CCSD reference, i.e. on par with the deviation of MP2 from CCSD, already for less than 300 training instances. For 2'000 training instances, the prediction error approaches would 2 kcal/mol. Moreover, the lack of convergence suggests that chemical accuracy (1 kcal/mol) could be reached if several thousand training data points had been available. Insets in Figure 6.2 show true (E_a^{ref}) vs. predicted (E_a^{est}) activation barriers for both reactions. Barriers in the range of zero to fifty kcal/mol are predicted with decent correlation coefficients (0.89 and 0.94 for E2 and S_N2, respectively). In short, after training on reference activation energies obtained for explicit transition state geometries (taken from QMrxn20 data set[3]), the learning curves in Figure 6.2 amount to overwhelming evidence that it is possible to circumvent the necessity for explicit transition state structural search when predicting activation energies for out-of-sample reactants.

The trends among learning curves in Figure 6.2, are consistent with literature results for equilibrium structures: The accuracy improves when going from BoB to SLATM and FCHL19 for a given training set size[7]. Most surprisingly, however, all R2B models based on geometry dependent representations are less accurate than one-hot encoding. While still unique (a necessary requirement for functional R2B models[241,242]) one-hot encoding is devoid of any structural information, and its outstanding performance is therefore in direct conflict with the commonly made conclusion that a physics inspired functional form of the representation is crucial for the performance of R2B models[7,243,244]. Relying only on the period and group information in the periodic table to encode composition, other geometry-free representations have also been applied successfully to the study of elpasolite[245], or perovskite[246] crystal structures. Here, by contrast, one-hot encoding provides the compositional information for a fixed scaffold.

One can speculate about the reasons for the surprising relative performance of one-hot encoding. Due to its inherent lack of resolution which prohibits the distinction between reactant and transition state geometry it could be that one-hot encoding represents a more efficient basis which effectively maps onto a lower dimensionality with superior learning performance. In particular, the inductive effect (practically independent of specific geometric details) is known to dominate barrier heights for the types of reactions under consideration[247], and it is explicitly accounted for through one-hot encoding without imposing the necessity to differentiate it from the configurational degrees of freedom.

Figure 2 shows one outlier per reaction. For the E2 case, the molecule closest in one-hot encoding to the failed prediction (only differs in X and Y) has a much smaller barrier of 12 kcal/mol. Similarly, for the S_N2 reaction, the closest molecule (only differs in X and Y) has a barrier of 24 kcal/mol. As such, this scarcity of training instances in close vicinity to the outlier might be at the origin for such relatively large prediction errors. To get an idea of the inner workings of the one-hot encoding model, we performed a principal component analysis (PCA) of the kernel matrix of the predictions which can go either way, i.e. E2 or S_N2. For this subset it is the difference in activation energy which will determine the kinetically stabilized product. Color coding the first two components by the difference in reference activation barrier labels for the two reactions results in the graphic featured in Fig. 6.3. Confidence ellipsoids of the covariance using Pearson correlation coefficients encode intuitive clusters corresponding to leaving-group/nucleo-phile combinations, and suggest that substituents have less significant effect on trends in activation energies. However, the eigenvalue spectrum of the PCA in Figure 6.3 decays rapidly only after the 21 eigenvalue which indicates the number of effective dimensions of the model, and implies that the substituents, alhtough smaller, still have an effect on the activation barrier. This is consistent with the dimensionality of the one-hot encoding representation: the vector length is 27 (3 X's, 4 Y's and 4·5 R's), which is overdetermined, meaning e.g. the X part of the representation vector consists of three elements F: [1, 0, 0], Cl: [0, 1, 0], or Br: [0, 0, 1]. This could also be uniquely defined with F [0, 0], Cl: [1, 0], Br: [0, 1], which leads a dimension of 21 and is in agreement with the dimensionality of the representation. To further investigate the R2B model, we looked at the training set selection. It is known that for clustered data (see Figure 3) random splits as used in this work tend to perform better than splits along a cluster, even though random splits are more congruent with the nature of the reaction space under investigation. As comparison, in a first model we excluded the FG NO_2 at position R2 and in a second model at two positions R2 and R3 from the training to see how one-hot encoding and FCHL19 perform for known functional groups but unknown positions in the test set. Figure 4 shows the learning curves for both cases. Although there is still learning, one-hot encoding does not perform as good as a structural representation (FCHL19). For FCHL19 in the E2 case, the learning is not affected at all compared to random training set selection and the model reaches a similar MAE for 800 training instances. FCHL19 is able to infer the missing functional group at position R2 from training compounds where this FG is present at the neighbouring position R1, since the corresponding representation vectors are similar. Also, one-hot encoding shows learning but it is not the dominant model anymore. In this case, learning is possible because the functional groups contribute additively to the activation energy as described in Marco Bragato *et. al*[247]. This means, that all the other functional groups improve, except NO_2 at positions R2 and R3, since it has no corresponding training data. For S_N2 both models perform

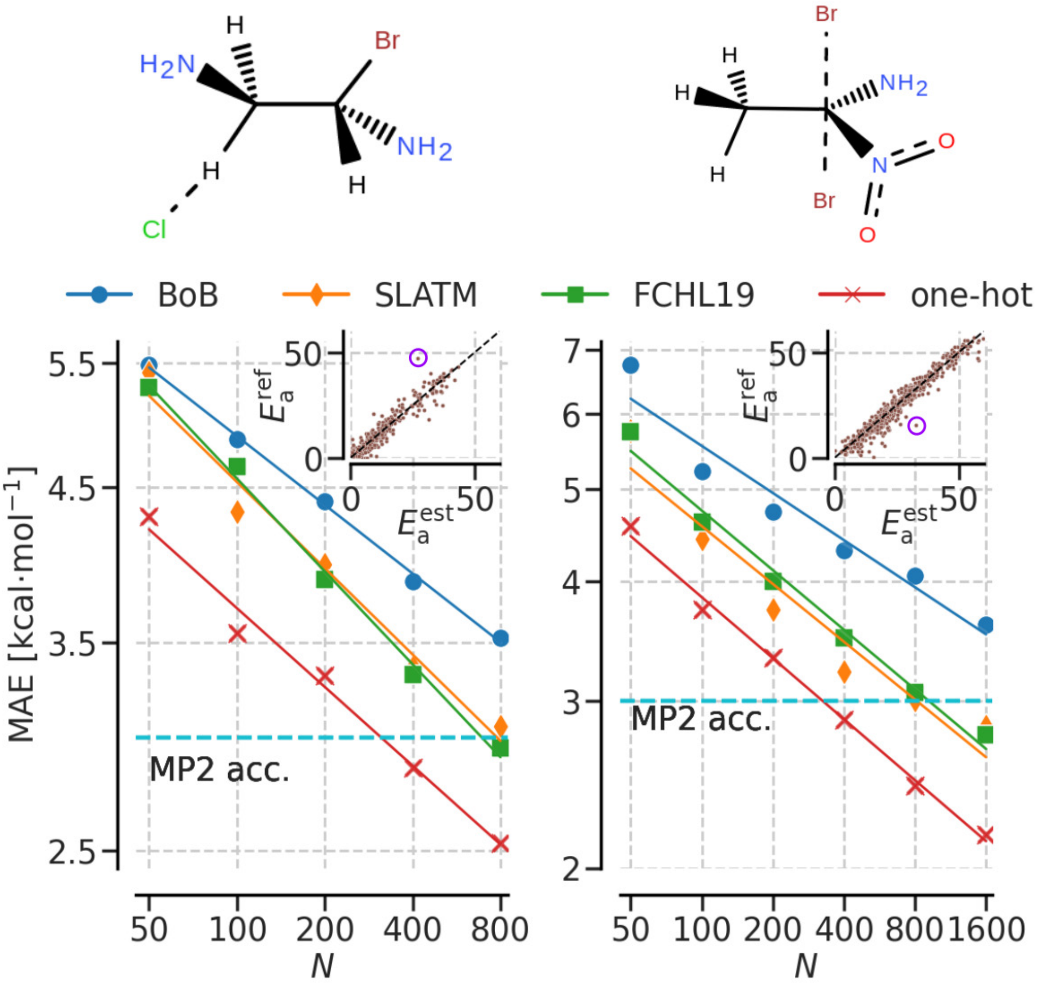


FIGURE 6.2: **Learning curves** Activation energy prediction errors (out-of-sample) as a function of training set size N for activation barriers (E_a) of E2 (left) and S_N2 (right) using reactant geometries as inputs only. Results are shown for four representations (BoB, SLATM, FCHL19, one-hot) used within KRR models. Training data reference level of theory corresponds to DF-LCCSD/cc-pVTZ//MP2/6-311G(d), and estimated MP2 error is denoted as a green dashed horizontal line. Insets: Reference vs. estimated activation barriers using one-hot-based predictions and R^2 values being 0.89 and 0.94 for E2 (left) and S_N2 (right), respectively.

worse when excluding a functional group, especially for the position at R2, which is closer to the reaction center and therefore contributes more to the barrier. This also explains why the models perform better if two functional groups are missing in the S_N2 reaction. The second functional group at position R3 adds more barriers to the test set with a smaller impact on the barrier (farther away form the reaction center), which makes the learning problem easier. For larger molecules, not all combinations of functional groups are present in the training data, rendering a cluster split a more realistic scenario. In those cases, one-hot encoding will be less applicable and likely outperformed by scalable approaches e.g. Amons.

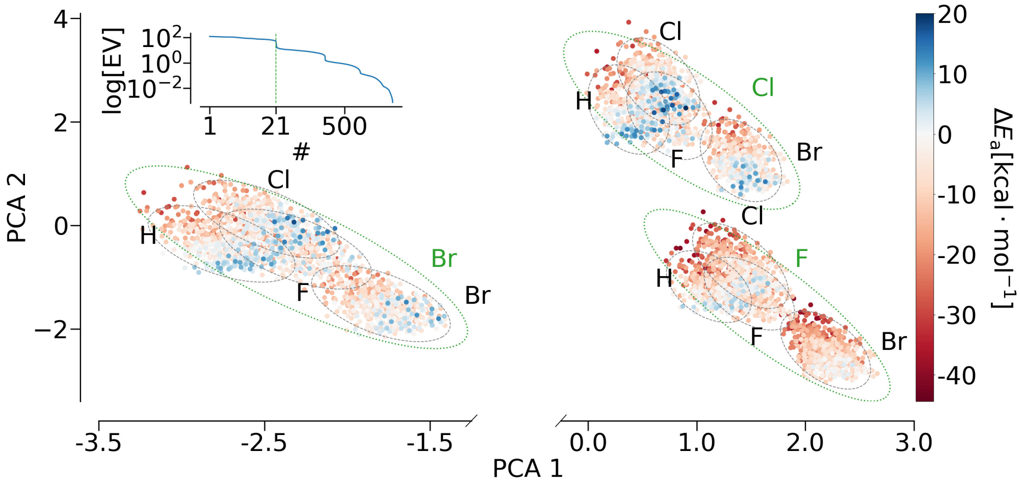


FIGURE 6.3: **Kernel PCA of the training set.** Kernel PCA of one hot encoding colored by the energy difference of activation energies of the two reactions $\Delta E_a = E_a^E - E_a^S$. Inset: Eigenvalues of the kernel PCA. Clusters represent most frequent combinations of leaving groups X (green) and nucleophiles Y (black).

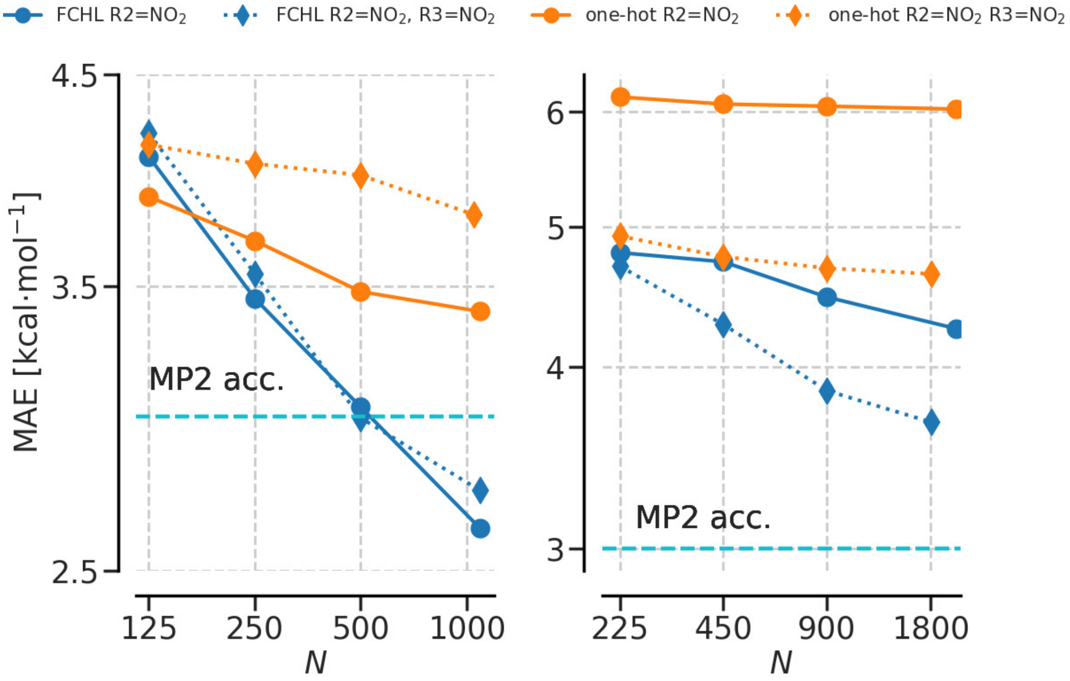


FIGURE 6.4: **Learning curves across clusters** test error (MAE) vs. training set size (N) excluding NO_2 from training at position R2 (spheres) and at positions R2 and R3 (diamonds) for both reactions E2 (left) and S_N2 (right). The test set only contains compounds with NO_2 at position R2 (spheres) or at positions R2 and R3 (diamonds).

6.4.2 New barrier estimates

Using one-hot encoding (leading to the most performing model) we have trained two models, corresponding to the 1′286 and 2′361 activation energies of E2 and S_N2

transition state geometries, respectively. Subsequently, these two models were used to predict 11’353 E2 and S_N2 activation barriers for which conventional transition state search methods had failed within the protocol leading up to the training data set[3]. A comparison of the Rogers-Tanimoto distances (see SI) between the QMrxn20 dataset and the missing data points showed that the dissimilarity within the QMrxn20 data set is comparable to the one of QMrxn20 vs the missing data points. Together with the learning curves shown above, this suggests that our model is applicable to the missing data points from QMrxn20. A summary of the difference in these predicted activation barriers is presented in Figure 6.5, where the *x*-axis corresponds to the nucleophiles Y, the *y*-axis to the leaving groups X. For every combination of X and Y, there are 5·5 squares for the functional groups at position R1 and R2. Within these, there are again 5·5 squares belonging to R3 and R4. Each of the squares represents one reaction for a given combination of R1-4, X, and Y. Simple heuristic reactivity rules emerge from inspection of these results: If the nucleophile and the leaving group are Cl, the preferred reaction is S_N2. If the nucleophile and the leaving group are F, the preferred reaction is E2. The functional groups at positions R1 and R2 favour the E2 due to their electron donating properties which disfavour a nucleophilic back side attack in the S_N2 reaction. A comprehensive overview is shown in Fig. 6.5. The same rules can be observed in Figure 6.6 which shows the distribution of the differences in activation barrier (ΔE^a) of the training, predicted and total data set. The molecules of the extreme cases, largest difference in activation energies, are shown for both reactions, E2 (left) and S_N2 (right). Figure 6.6 shows a favourization of the E2 reaction of a rate of roughly 75%. These results have to be taken with caution, since this shift in E2 can also have occurred due to the composition of the molecules in the training set, as well as the choice of small functional groups that minimizes steric effects. A more detailed discussion of the training and the data set completion with the R2B model can be found in Appendix C.

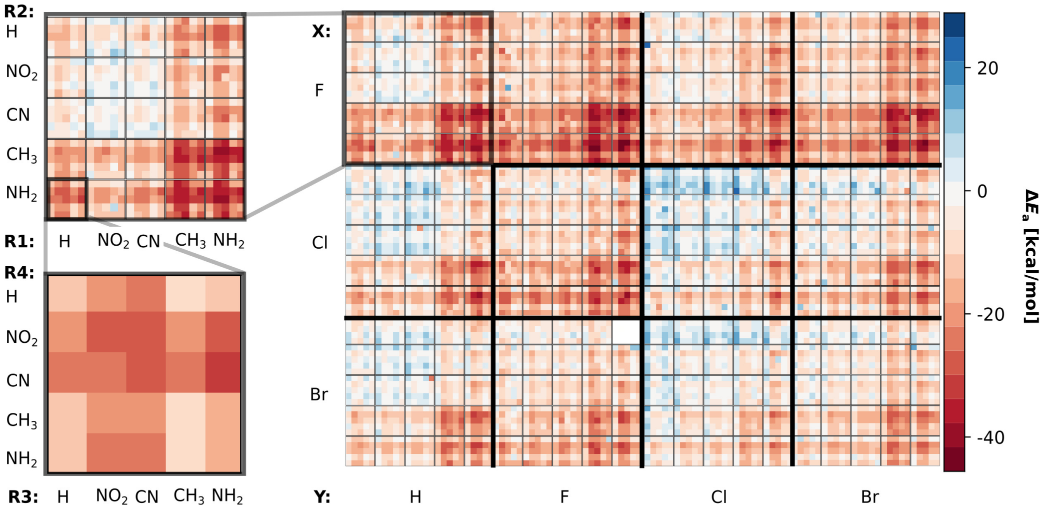


FIGURE 6.5: **Completion of data set using predictions of R2B models** Differences in activation energies ($\Delta E_a = E_a^E - E_a^S$) for all 7,500 reactions (calculated and predicted). Every square stands for a combination of R1-4, X, and Y shown in Figure 1. Positive values denote compounds that undergo a S_N2 reaction and negative values lead towards an E2 reaction.

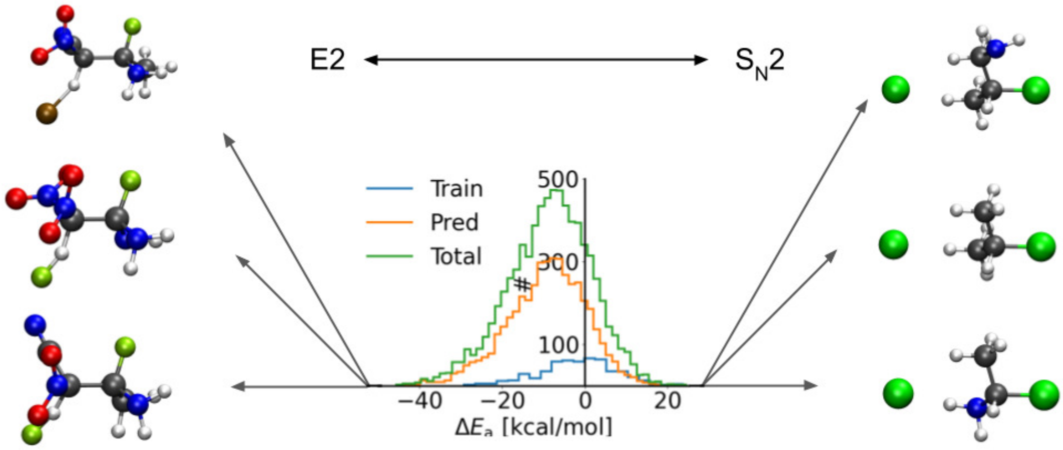


FIGURE 6.6: **Histogram of energy distribution of ΔE_a.** Differences in activation energies ($\Delta E_a = E_a^E - E_a^S$) of 529 overlapping training instances (blue), 11k predictions (orange) and all 7'500 reactions (green). Molecules of the three highest, respectively lowest barrier differences are shown as molecules.

6.4.3 Design rule extraction

So far, most studies based on artificial neural networks aimed at predicting chemical reactions using experimental data do not account for the kinetics of reactions. It is well known, however, that activation barriers are crucial for chemical synthesis and retrosynthesis planning. This is exemplified by a decision tree for the competing reactions E2 and S_N2 in Figure 6.7. The goal of such trees is to improve the search for better reaction pathways (lower activation barriers), by showing the estimated change in energy when changing functional groups, leaving groups, or nucleophiles. To extract such rules for the design problem, a large and consistent reaction data set is needed. After completing the data set[3], we are now able to identify (given a desired product) the estimated changes in the activation barrier, when substituting specific functional groups, leaving groups, or nucleophiles. This way, the yield of chemical reactions can be optimized by getting insights of the effects that functional groups have on a certain molecule. Furthermore, this insight could be used to direct reactions towards the desired product. Figure 6.7 shows such a possible decision tree to determine the change in barriers while exchanging substituents. Starting from the total data set (left energy level), the first decision considers the functional group NH_2 at position R1. Going down the tree means accepting the suggested change and the respective compounds, while going up means declining and removing these compounds from the data. Depending on which product is sought after, hints to improve the energy path can be found while constantly accepting (going down) or declining (going up) the tree. For example, if the desired reaction is E2, then the best way is to go down on the tree (decision accepted) which adds electron withdrawing groups to the R3 and R4 position, as well as electron donating groups to R1 and R2. In Figure 6.7 the first decision redirects the barrier towards E2 about ~8 kcal/mol by adding an electron withdrawing group (NO_2) on the α-carbon. On the other hand, electron donating group at the β-carbon favour the E2 reaction because they facilitate the abstraction of the leaving group, which is shown in the second and the third decision, where NH_2 was added in both positions, R1 and R2. In addition to the R2B predictions, which tell you the outcome of a specific combination of one reaction, a decision trees gives simple rules as an coarsened aggregation that can be used in reaction design to achieve a desired outcome.

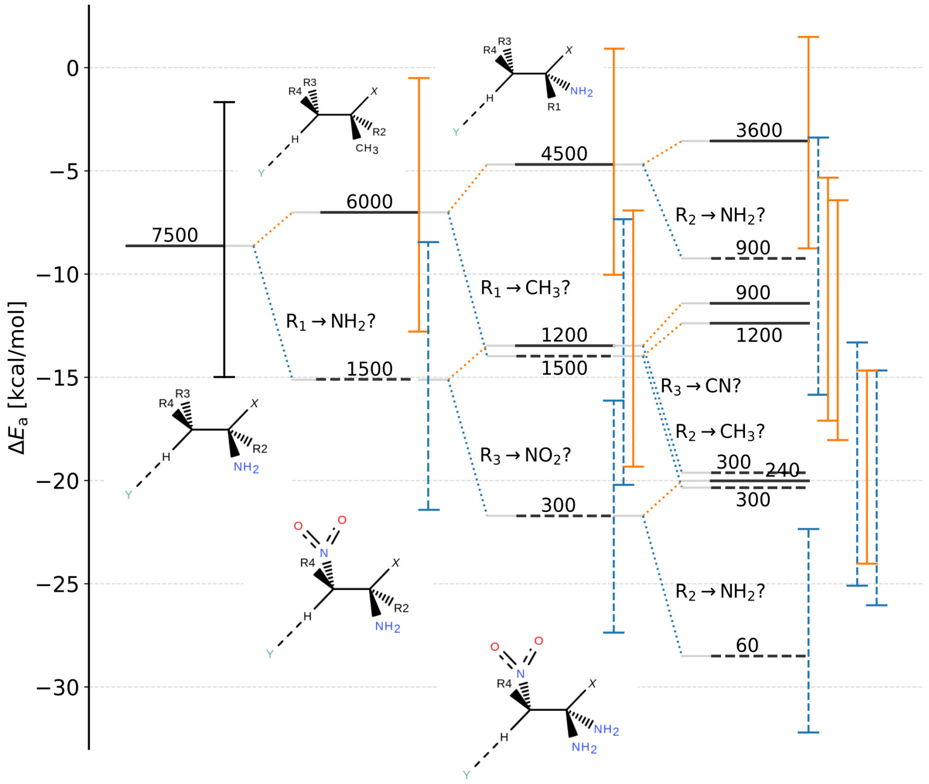


FIGURE 6.7: **Decision tree using extracted rules and design guidelines.** Decision tree using the R2B estimated activation barriers to predict changes in barrier heights by starting at all reactions (first energy level on the left) and subsequently apply changes by substituting functional groups, leaving groups and nucleophiles with E2 as an example. Blue dotted lines refer to an accepted change meaning only compounds containing this substituents at the position are considered. Orange dotted lines refer to substitution declined, meaning all compounds except the decision are kept. Vertical lines on the right of energy levels denote the minimum first (lower limit), and the third (upper limit) quartile of a box plot over the energy range. Numbers above energy levels correspond to the number of compounds left after the decision. Lewis structures resemble the decision in question.

6.4.4 Estimates of reactant and transition state geometries

Additionally to barriers, we analysed the geometries of the transition states as well as the geometries of the reactant complexes[3]. Choosing key geometrical parameters, such as distances, angles, and dihedrals, we were able to train R2B models to learn these properties using the one-hot encoding as representation. These parameters were extracted from the ethylene scaffold defining the key positions of the substituents, leaving groups, and nucleophiles shown in Figure 6.8 compounds **2** and **3** for the E2 and S_N2 reaction, respectively.

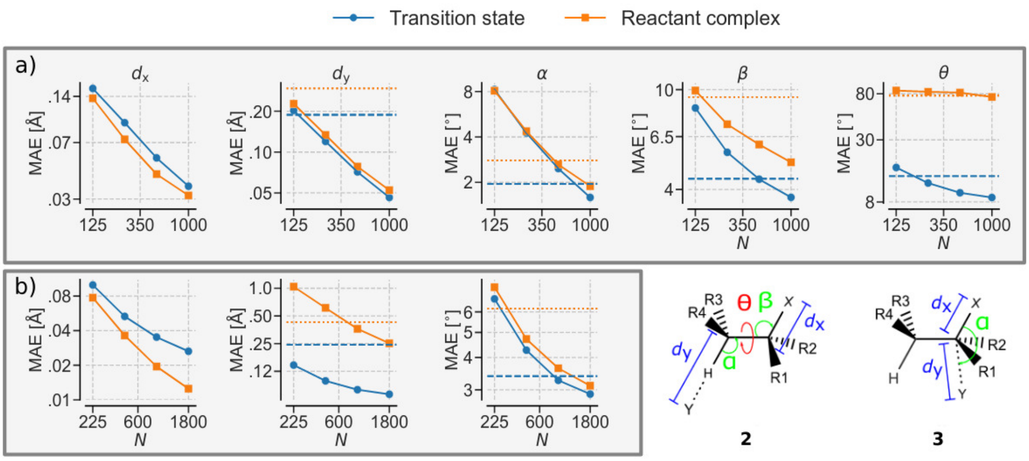


FIGURE 6.8: **Model evaluation of geometrical properties using learning curves.** Test errors (MAE) of distances $d_{x,y}$, angles α and β and dihedrals θ for both reactions E2 (a) and S_N2 (b). Horizontal lines correspond to the null model which uses the mean value of the training set for predictions. Compounds (**2** and **3**) illustrate the learned properties of the E2 reaction (**2**) and the S_N2 reaction (**3**) for reactant complexes and transition states.

The parameters for the E2 reaction are the C-X distance d_x, the C-Y distance d_y, the X-C-C angle α, the C-C-Y angle β, and the X-C-C-Y dihedral θ. Similarly for S_N2, we have the C-X distance d_x, the C-Y distance d_y, and the X-C-Y angle α. For every parameter, a separate model was trained using the one-hot representation. Although this representation does not contain any geometrical information, learning was achieved for every parameter. Figure 6.8 shows the learning curves and as horizontal dashed lines the null model which uses the mean of the training set for predictions. In the same way as for the transition state geometries, we also trained a model for the reactant complexes. Figure 6.8 shows the learning curves for both, transition states and reactant complexes. The results for both geometries are similar except for the dihedral of the reactant complexes. The poor performance results from the conformer search of the reactants. Compared to bond distances, dihedrals have multiple local minima which leads to larger differences between the reactant and transition state structures. The variance of the dihedrals are significantly higher which makes the learning task much harder. The one-hot representation does not contain any geometrical information and therefore is not able to learn the different geometries only using information about the constitution (R's, X's, and Y's) of the reactant complexes. The poor performance of the model on angles and especially on dihedrals renders the one-hot encoding impractical for 3D geometry predictions. The recently published Graph to Structure (G2S) QML model[248] seem to be more suitable for the 3D coordinate prediction problem in QMrxn20.

6.4.5 Hammond's postulate

To investigate Hammond's postulate we took the difference in the predicted geometries (d_x and d_y) for all 7,500 reactions for the E2 and the S_N2 reaction, respectively. Then we plotted these values against the activation energies of both reactions E_a^E and E_a^S (Figure 6.9). The distances Δd_x correlate well with the energies with R^2 values of 0.87 and 0.80 for E2 and S_N2, respectively. This is explained by the leaving group

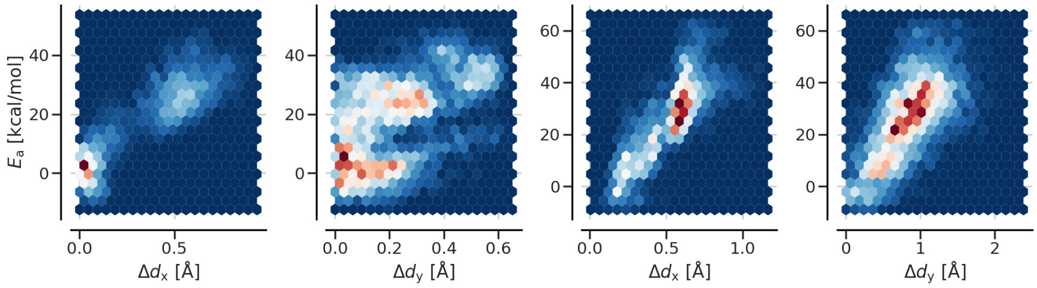


FIGURE 6.9: **Applicability of Hammond's postulate.**Frequency heat map of activation energies projected onto structural differences in distances (d_x and d_y) between reactant complex conformers and transition states for both reactions E2 (first two plots) and S_N2 (last two plots). For S_N2 a good linear correlation (R^2 are 0.8 and 0.65 for d_x and d_y, respectively) in agreement with Hammond's postulate can be observed, while for E2 only d_x shows good correlation ($R^2 = 0.86$) whereas d_y lacks correlation ($R^2 = 0.5$).

that is bonded to the carbon atom in the reactant complex and only small changes in distance happen moving towards the transition state geometry. For the S_N2 reaction, the backside attack of the nucleophile does not allow a broad distribution of distances and angles in the reactant complex and the transition state. Moreover, the changes in geometry between the reactant complex and the transition state are modest. Therefore, the parameter Δd_y for the S_N2 correlates well with the activation energy E_a^S, which results in an R^2 value of 0.65. The attack of the nucleophile on the hydrogen atom (E2 reaction) allows for a much broader distribution of the position of the nucleophile in the transition state. This makes the learning problem more difficult, especially for a representation not including geometrical information. These higher degrees of freedom result in an R^2 value of 0.50.

Hammond's postulate typically holds for the end points of an intrinsic reaction coordinate (IRC) calculation[249–251] which leads to a local minimum close to the transition state. Therefore, the reactant only needs a few reorganisations towards the transition state. For geometries that are farther away from the transition state (such as in our E2), Hammond's postulate cannot hold anymore. This means that even though more reorganization steps towards a transition state have to be made, the activation energy is not affected anymore. As a consequence, Hammond's postulate is no longer applicable.

6.5 Conclusion

We have introduced a new machine learning model dubbed Reactant-To-Barrier (R2B) to predict activation barriers using reactants as input only. This approach renders the model practically useful, as the dependency on the transition state geometry is only implicitly obtained at the training stage, and not explicitly required for querying the model. We find that one-hot-encoding, the trivial geometry free based representation, yields even better results than geometry based representations designed for equilibrium structures. As such, our results indicate that accounting only for the combinations of functional groups, leaving group, and nucleophile of the reaction is sufficient for promising data-efficiency of the model. Using R2B predictions, we completed the reaction space of QMrxn20[3]. Future work could include delta ML[18] to improve these results even further, as corroborated by preliminary results in Ref.[3], further improvements on the representation (as recently found to lead to improved barrier predictions for enantioselectivity in metal-organic catalysts[234]), or the inclusion of catalytic or solvent effects[252].

Using R2B predicted activation barriers, we have also introduced the notion of a decision tree, enabling the design and discrimination of either reaction channel encoded in the data. Such trees systematically extract the information hidden in the data and the model regarding the combinatorial many-body effects of functional groups, leaving groups, and nucleophiles which result in one chemical reaction being favoured over the other. As such, they enable the control of chemical reactions in the design space spanned by reactants. Finally, we also report on geometries of the reactant complexes consisting of different conformers, as well as on R2B based transition state geometry predictions. Using these results, we discuss the limitations of Hammond's postulate which does not hold for the E2 reactant complexes stored in the QMrxn20 data set[3].

Supplementary Material

The supplementary material (Appendix C) contains the results used to generate the learning curves for the barrier learning (Table 1 and 2), and the geometry learning (Table 3 and 4). It also gives a brief explanation how the models were trained and shows a heat map for a hyperparameter scan of sigmas and lambdas containing the training errors (Figure 1). Additionally, we added more learning curves (barrier learning) using different geometries as input for the representations. Finally, we added a Figure 3 which compares the Rogers-Tanimoto coefficients between the training and the test set.

Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreements #952165 and #957189. This project has received funding from the European Research Council (ERC) under the European UnionâĂŹs Horizon 2020 research and innovation program (grant agreement No. 772834). This result only reflects the author's view and the EU is not responsible for any use that may be made of the information it contains. This work was partly supported by the NCCR MARVEL, funded by the Swiss National Science Foundation.

Data Availability

The data that support the findings of this study are openly available at
http://doi.org/10.5281/zenodo.4925938.

Chapter 7

Geometry Relaxation and Transition State Search

Geometry Relaxation and Transition State Search throughout Chemical Compound Space with Quantum Machine Learning

S. Heinen, G.F. von Rudorff, O.A. von Lilienfeld, arXiv preprint arXiv:2205.02623

7.1 Abstract

We use energies and forces predicted within response operator based quantum machine learning (OQML) to perform geometry optimization and transition state search calculations with legacy optimizers. For randomly sampled initial coordinates of small organic query molecules we report systematic improvement of equilibrium and transition state geometry output as training set sizes increase. Out-of-sample S_N2 reactant complexes and transition state geometries have been predicted using the LBFGS and the QST2 algorithm with an RMSD of 0.16 and 0.4 Å — after training on up to 200 reactant complexes relaxations and transition state search trajectories from the QMrxn20 data-set, respectively. For geometry optimizations, we have also considered relaxation paths up to 5′500 constitutional isomers with sum formula $C_7H_{10}O_2$ from the QM9-database. Using the resulting OQML models with an LBFGS optimizer reproduces the minimum geometry with an RMSD of 0.14 Å. For converged equilibrium and transition state geometries subsequent vibrational normal mode frequency analysis indicates deviation from MP2 reference results by on average 14 and 26 cm^{-1}, respectively. While the numerical cost for OQML predictions is negligible in comparison to DFT or MP2, the number of steps until convergence is typically larger in either case. The success rate for reaching convergence, however, improves systematically with training set size, underscoring OQML's potential for universal applicability.

7.2 Introduction

One of the fundamental challenges in quantum chemistry is the understanding of reaction mechanisms in order to predict chemical processes. To this end, numerous neural networks (reaction predictors) have been introduced, proposing the most likely reaction path way[9–15] for a given product. These models were trained on data obtained from experimental studies[253] only containing the molecular graph (as SMILES strings[70,71]) and their corresponding yields. However, a crucial property of a chemical reaction is the activation energy (i.e. the difference between reactant and transition state energy), linked to the kinetics of the reaction. To predict activation energies with conventional electronic structure methods, both the reactant complex geometry and the transition state geometry need to be obtained. This is commonly done by iteratively following gradients of the potential energy surface (PES) towards the minimum or the saddle point, respectively. Due to the iterative nature of these schemes, imposing the repeated need to perform self-consistent field calculations to obtain updated forces, the computational burden is as large as it is predictable[2]. Furthermore, finding saddle points remain an additional challenge because often enough considerable manual work is required beforehand in order to generate reasonable initial structure guesses. Consequently, it is not surprising that so far only few reaction data-sets which contain transition state geometries as well as corresponding energies have been published in 2020[3,16], and 2021[254].

Only very recently, attempts have been made to use machine learning models to speed up transition state predictions. In 2019, Bligaard and co-workers used the nudged elastic band (NEB)[151,225] method to find transition states relying on neural network based Δ-ML model[18] together with a low level of theory as baseline[255]. More recently, Mortensen et al. contributed the 'atomistic structure learning algorithm' (ASLA)[256], enabling autonomous structure determination with much reduced need for costly first-principles total energy calculations. Lemm *et. al.*[257] introduced the graph to structure (G2S) machine learning model, predicting reactant complexes and transition state geometries for the QMrxn20[3] data-set without any account for energy considerations, solely using molecular graphs as input. Also, for 30 small organic molecules neural networks predicting energies and forces to accelerate the geometry optimization in between *ab initio* iterations was introduced by Meyer and Hauser[258], and by Born and Kästner[259]. Similar to G2S, Makós et al.[260] propose a 'transition state generative adversarial neural network' (TS-GAN) which estimates transition state geometries using information from reactants and products only. This procedure allows for better initial geometries for a transition state search reducing the number of steps towards a saddle point. Jackson *et. al.*[254] developed a neural network (TSNet) predicting transition states for a small ($\sim$ 50) S_N2 reaction dateset, as well as geometries of the QM9[67] data-set.

However, to the best of our knowledge there is no machine learning model yet using, in strict analogy to the conventional quantum chemistry based protocol, predicted energies and forces only within the conventional optimization algorithms in order to relax geometries or find transition states. To tackle this challenge, we have used for this paper the response operator based quantum machine learning (OQML)[168,169] model with the FCHL representation[60,61] and trained on energies and forces across chemical compound space in order to speed up geometry relaxations as well as transition state searches for new, out-of-sample compounds (see Fig. 1). As for any properly trained QML model, prediction errors decay systematically with training set

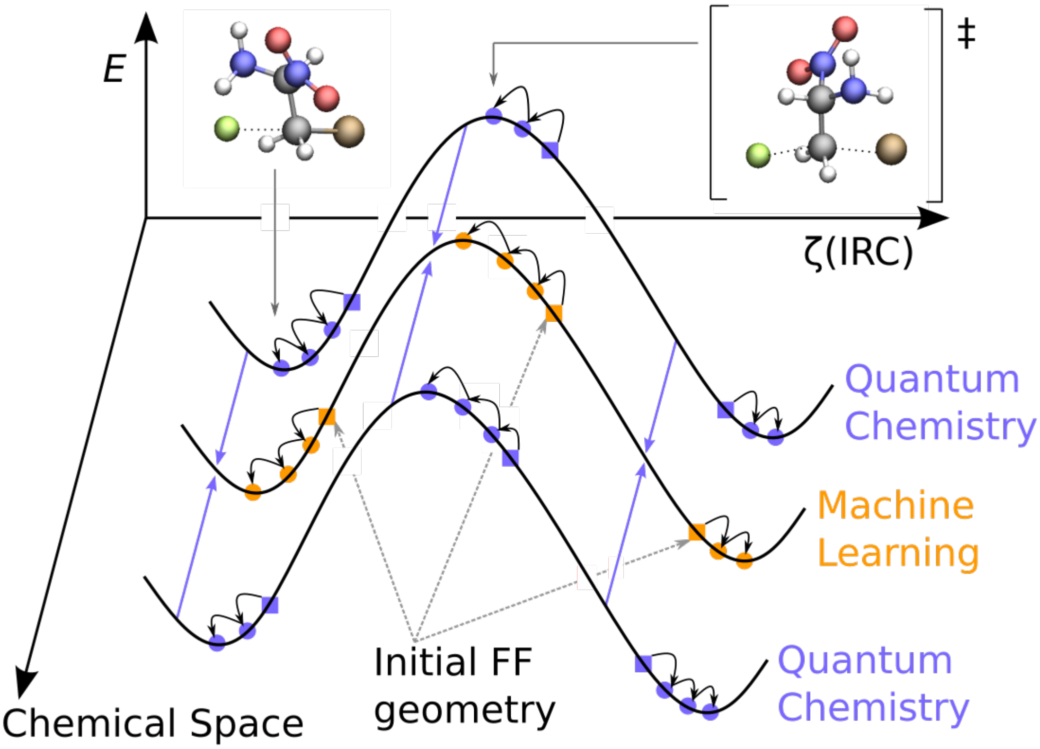


FIGURE 7.1: Schematic potential energy surfaces in chemical compound space. Arrows show the working principle of OQML based iterative structural optimization: Training on reaction profiles of different chemical systems (purple), the OQML model is able to interpolate forces and energies throughout chemical compound space enabling the relaxation of the reactant and the search of the transition state (orange). Input geometries (squares) are easily obtained, e.g. from universal force field predictions.

size, and we demonstrate for the chemistries presented that encouraging levels of accuracy can be reached.

First, we have investigated geometry optimizations for all constitutional isomers with $C_7H_{10}O_2$ sum formula drawn from QM9[67]. After training the OQML model on random geometries along the optimization path of of 5500 calculations going from a UFF minimum energy geometry to the B3LYP/6-31G(2df) minimum geometry we optimized the remaining 500 constitutional isomers resulting in a total RMSD of only 0.14 Å. To probe transition states, we have trained OQML models on the QMrxn20 data-set[3] with thousands of examples for the S_N2 text book reaction at MP2/6-311G(d) level of theory, enabling the relaxation of reactant complexes and the search of transition states which both compare well to common density functional theory (DFT) results. As shown in Figure 7.1 this means training the OQML model on quantum chemistry reference energies and forces along the optimization trajectory obtained for relaxation and transition state search runs of training systems. Starting with universal force field (UFF)[199] geometries, OQML subsequently predicts energies and forces for 200 out-of-sample query systems, thereby enabling the application of legacy relaxation and transition state search algorithms throughout chemical compound space.

7.3 Methods

We have relied on operator quantum machine learning (OQML) approach as introduced by Christensen *et. al.*[168,169] which is a kernel ridge regression (KRR) model which explicitly encodes target functions and their derivatives. A detailed derivation can be found in Christensen *et. al.*[60] section 2 (Operator quantum machine learning). To train a model the regression coefficients $\boldsymbol{\alpha}$ the following cost function is minimized:

$$J(\boldsymbol{\alpha}) = \left\| \begin{bmatrix} \mathbf{y} \\ \mathbf{f} \end{bmatrix} - \begin{bmatrix} \mathbf{K} \\ -\frac{\partial}{\partial \mathbf{r}}\mathbf{K} \end{bmatrix} \boldsymbol{\alpha} \right\|_2^2 \tag{7.1}$$

with $\mathbf{K}$ being the training kernel, $\mathbf{y}$ the energies, and $\mathbf{f}$ the forces. To predict the energies following matrix equation can be used:

$$\mathbf{y}^{\text{est}} = \mathbf{K}_s\boldsymbol{\alpha} \tag{7.2}$$

and similarly for the forces:

$$\mathbf{f}^{\text{est}} = -\frac{\partial}{\partial \mathbf{r}}\mathbf{K}_s\boldsymbol{\alpha} \tag{7.3}$$

where $\mathbf{K}_s$ being the test Kernel containing training and test instances.

The representation used throughout this work is the FCHL19 representation[60]. FCHL19 makes use of interatomic distances in its two body terms and includes interatomic angles in the three body term. FCHL19 was selected because of its remarkable performance for QM9 related data-sets[60], and due to being the best structure based representation in direct learning of activation energies in QMrxn20[4].

To find transition states, the Gaussian09 QST2[261] algorithm with loose convergence criteria was used, which allows for external energies and forces, in this case from OQML. Note that no explicit Hessian is required by this method nor is one available from our model. For both, reactant complexes and transition states, 300 out of sample reactions were chosen. For these 300 reactions also DFT geometry optimizations as well as transition state searches were performed. The three functionals used were: B3LYP[125,262], PBE0[263], and ωB97X[264] with the 6-311G(d)[265–267] basis set (same as for the reference method in[3]). More details of the training of the models can be found in the SI.

The python package rmsd[268] with the Kabsch algorithm[269] was used to obtain the RMSD's including hydrogens. For every training set size, the success rate of the geometry optimization (truncated after 50 iterations) and the transition state search (truncated after 100 iterations which is the gaussian default) was reported. Scripts and data can be found in the SI[270].

7.4 Results and Discussion

In the context of statistical learning theory, cross-validated learning curves amount to numerical proof of the robustness and applicability of a machine learning model, and they provide quantitative measures of the data-efficiency obtained. For the three OQML models studied here-within (geometries of constitutional isomers, of reactant complexes, and of transition states), Fig. 7.2(left) displays the OQML based learning curves for energies (top) and atomic forces (bottom) which indicate the systematic improvement of energy and force predictions as training set size increases.

The learning curve for the constitutional isomers are in line with the results by Christensen *et. al.*[169]. Surprisingly, although FCHL19 was optimized for small organic closed shell molecules, the learning curves for the reactant complexes and the transition states have a faster learning rate. A possible reason for this trend could be that the reactions in the QMrxn20 data-set share a common scaffold with only the substituents changing which represents a lower effective dimensionality of the problem which typically leads to faster learning. Also, relaxations for only 200 reactions were considered in the training set which implies an overall smaller subset of the chemical universe. By contrast, for the constitutional isomers, geometries from 5500 different compounds were chosen, covering a much broader chemical space.

While accurate OQML based estimates of forces and energies are necessary for subsequent relaxation and transition state search, the eventual key figure of merit, the RMSD with respect to query reference coordinates for increasing training set size, amounts to a performance curve as shown in the mid panel of Figure 7.2. We observe strong systematic improvements with increasing training set size of the RMSD for the constitutional isomers and reactant complexes. By contrast, RMSD performance curves for transition states, while also monotonically increasing with training set size, exhibit substantially smaller learning rates. Differences in learning for different data-sets while using the same representations and model architectures implies that the target function is more complicated. One can argue that the constitutional isomers are less pathological since they consist of small organic and closed shell molecules, whereas the transition states include charged compounds and non-covalent binding to leaving and attacking groups. The relatively flat progress made for the transition states might also simply due to the fact of the more complex optimization problems towards a saddle point compared to the simple downhill search of a geometry optimization. More specifically, due to the underpinning high dimensionality, the training set grows much more rapidly when adding a new reactive system including optimization steps along the way to the saddle point. This implies that the training will be less efficient. Possible ways to mitigate such a bottleneck could

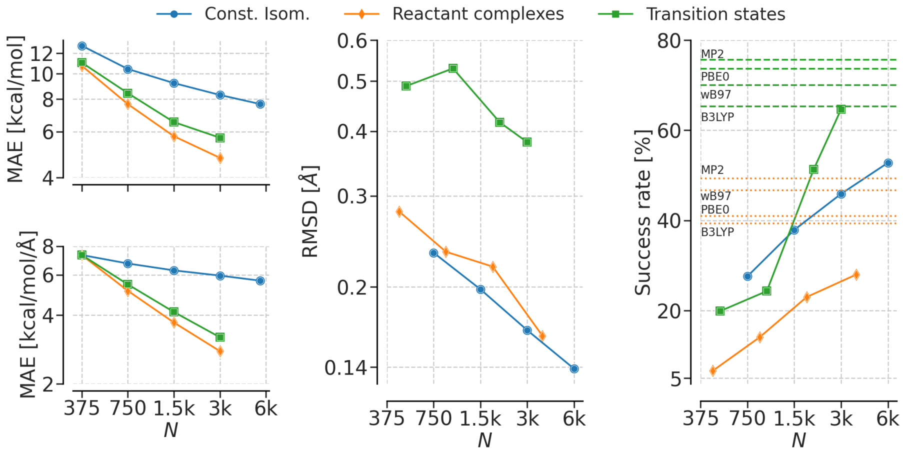


FIGURE 7.2: Learning curves for energies (top left) and atomic forces (bottom left). RMSD performance curves of geometry vs. training set size N (middle). Success rates of the optimization and transition state searches (right). Colours correspond to results based on three distinct data-sets: constitutional isomers (QM9), reactant complexes (QMrxn20), and transition states (QMrxn20). Dashed green and dotted orange horizontal lines correspond to the success rate of the reference calculations for the TS searches and the geometry optimization, respectively

include the use of the Amons approach[271] which decomposes molecules in substructures, drastically reducing the effective dimensionality of the problem. Also Δ-ML[18], multi-level grid combination techniques[272], or transfer learning[273,274] could lead to significant speed-ups and would render the models more transferable.

Regarding the performance curve for the transition states it is encouraging to note that the slope is substantially steeper than for the equilibrium geometry, also indicating that the OQML based energies and gradients also work well for locating saddle-points, which is unprecedented in literature, to the best of our knowledge. A direct one-to-one comparison to the equilibrium geometry relaxations, however, is not possible as the differences might also be due to the use of two very different optimizers (LBFGS vs. QST2).

Performance curves for success rates have also been included in Figure 7.2 right. We note that for all models and data-sets the success rate of the optimization runs systematically increases with training set size. We find that even for OQML models trained on small training set sizes resulting in relatively high RMSDs (~ 0.4 Å), the success rate increases from 7% to 28% and from 20% to 65% for the reactant complexes and the transition states, respectively, as shown in Figure 7.2, closing to the MP2 success rates (horizontal lines). Surprisingly, even though the RMSD performance curve for the constitutional isomer set is the best, the success performance curve is the worst. This could be due to the higher dimensionality in the QM9 based data-sets, where the optimizer has to locate the minimum for substantially more degrees of freedom. In any case, the systematic increase in success rate represents strong evidence in favor of the proposed model, as one can always improve it through mere addition of training instances, apparently resulting in increasingly smooth potential energy surfaces with few artifacts—an important prerequisite for successful optimization runs using algorithms such as LBFGS[275].

For further analysis of reactant complexes and transition states we used 300 out-of-sample compounds. Table 7.1 shows a summary of the predictions of the reactant complex optimization of the ground states (GS) and the transition state (TS) search, as well as the comparison to the three DFT methods (B3LYP, PBE0, ωB97X) with the 6-311G(d) basis set. RMSDs for the geometries are around 0.1Å for the DFT methods and 0.4Å for the OQML method considering the transition states. The performance of the ML model reaches the same accuracy for the reactant complexes as the DFT geometry optimization resulting in RMSD's on the order of 0.05 to 0.14 Å. Using the same model we calculated numerical frequencies and reached a mean absolute error over the 300 test transition states of 33.63 cm^{-1} and 14.09 cm^{-1} for transition states and reactant complexes, respectively, which is comparable to the DFT errors.

For the activation energy E_a, the ML model reaches slightly higher MAE of 5.851 kcal/mol compared to the MAE of ~1kcal/mol of the DFT methods. Although, the error of ~6 kcal/mol is still high, other ML models could be used to learn the activation energies e.g. the R2B model[4] which was applied on this data-set and solely uses the molecular graph as input for the ML model.

We note that a direct comparison of the OQML and DFT results in Table I would not be fair as OQML was fitted on data similar to the query compounds while DFT methods and basis sets are universal in nature and were fitted against much more diverse chemistries.

Finally, we showcase the OQML predicted results for the transition state of one randomly drawn exemplary reaction, involving [H(CN)C-C(CH_3)(NH2)] with Cl and F as leaving group and nucleophile, respectively. In Figure 7.3 the calculated transition state normal modes are shown, energies once predicted by OQML and once as obtained from MP2 for comparison. Even though, the RMSD of the predicted geometries are off by 0.4 Å, the curvature is described reasonably well by the OQML model, which is supported by the relatively small errors in frequencies, as well as by the high success rate of the transition state search.

Method	RMSD [Å]	$\Delta\nu$ [cm^{-1}]	E_a [kcal·mol^{-1}]	N
OQML (FCHL19)	0.161 0.381	14.09 26.06 (94.21)	5.851	3753/3812
B3LYP/6-311G(d)	0.053 0.134	31.37 32.37 (145.18)	1.352	116[276]
PBE0/6-311G(d)	0.046 0.096	22.93 33.89 (147.56)	1.016	106[263,277]
ωB97XD/6-311G(d)	0.033 0.096	13.94 33.63 (150.01)	0.853	1108[264]

TABLE 7.1: Summary of results for 300 out of sample test cases for geometry optimizations (left) and transition state searches (right) for OQML models and three DFT methods for comparison with MP2/6-311G(d) as reference. The table shows the difference in geometry (RMSD), in frequency (ν), and in activation energy (E_a) for each method. N corresponds to the training set size of the ML models (MP2/6-311G(d)) and to the data set size for parametrization of the DFT functionals. Geometry optimizations using the LBFGS algorithm from the ASE package were truncated after 50 iterations and the default thresh hold (*fmax* = 0.05 eV/Å) was used. The limit for the transition state search was the default iteration limit of 100 steps. Frequency values in parenthesis for the transition states are the errors of the first (imaginary frequency)

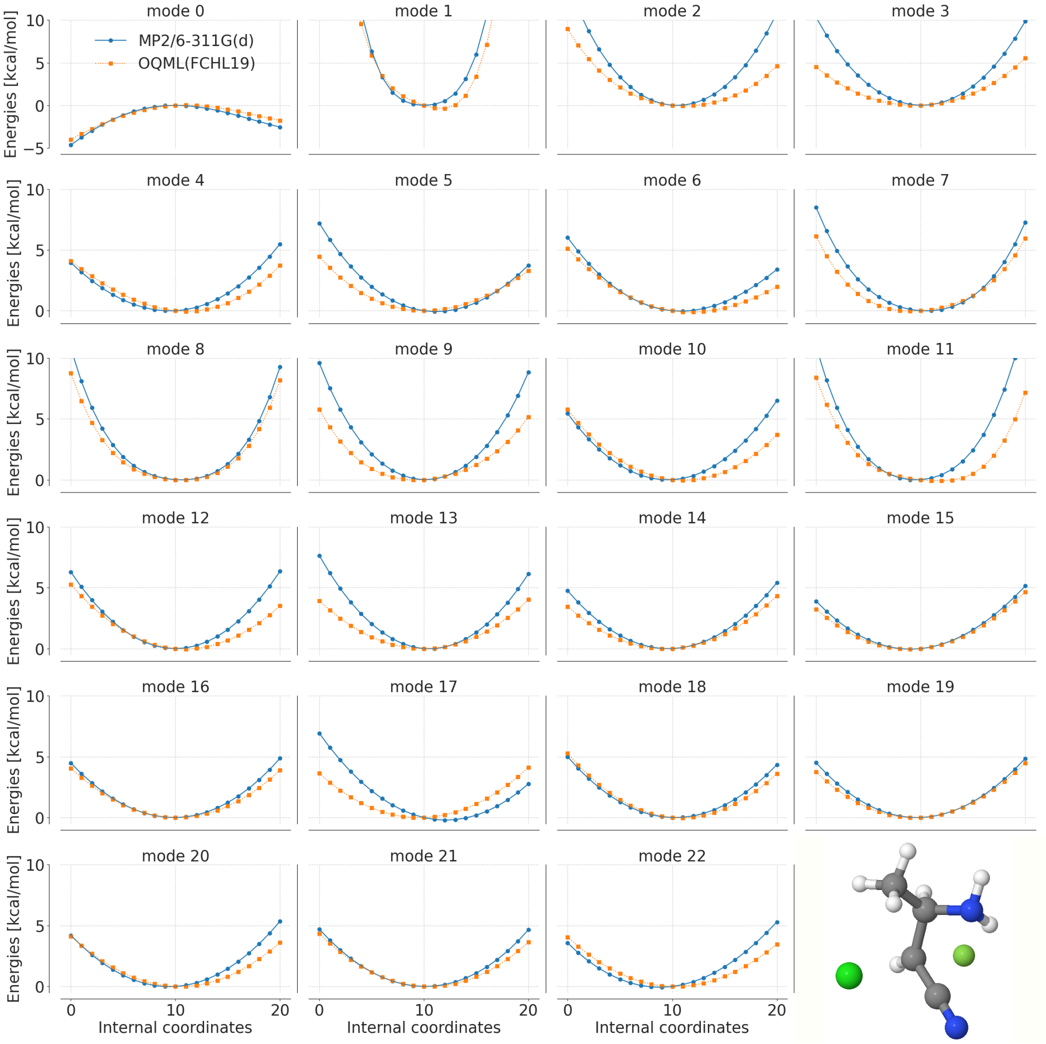


FIGURE 7.3: Example normal mode scan showing energy changes as a function of distortion along TS modes for the transition state of the S_N2 reaction of [H(CN)C-C(CH_3)(NH2)] with Cl and F as leaving group and nucleophile, respectively. Geometry of an MP2/6-311G(d) converged and validated TS was used and distorted along its normal modes. Subsequently, single point calculation (MP2), as well as ML predictions were plotted for the first 23 normal modes and their displacements. The x-axis describes the index of the distorted geometry and the y-axis describes the energy relative to the MP2 equilibrium geometry. Both energies, MP2 (blue) and ML (orange) are scaled by the equilibrium geometry (index 10).

Conclusion

Our findings indicate that OQML can be used to optimize geometries and search for saddle points (transition states) across chemical compound space. OQML is a surrogate model of conventional quantum based energies and forces, and can be successfully employed within legacy optmizers. Based on the OQML approach accuracies for RMSDs, frequencies, and activation energies improve as training set sizes increase. Similarly, the convergence success rates also improve for larger training set sizes.

Learning curves exhibit linear decay as a function of the training set size on logarithmized scales, indicating that even further improvements of the model could be reached by adding more training data. Performance curves of RMSDs suggest that the optimization process (RMSD as well as success rate) based on these models could also be further improved by increasing the training set size. Especially for the constitutional isomers (small, organic, and closed shell molecules) the description of the potential energy surfaces improves steadily by adding more training data, which improves the success rate from 27% to 52% for the lowest and the largest training set size, respectively. Vibrational frequencies obtained by OQML for our maximal training set size deviate from the reference MP2 frequencies no more than DFT.

To explore out of equilibrium geometries farther away from a local minima the QM7x data-set[278] could be investigated in the future. To make OQML more transferable and applicable also to larger reactants, an Amon based extension[271] could be implemented. OQML could also be helpful for the generation of large and consistent data-sets in quantum chemistry, especially for the study of reactions.

Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement #957189. This project has received funding from the European Research Council (ERC) under the European UnionâĂŹs Horizon 2020 research and innovation program (grant agreement No. 772834). This result only reflects the author's view and the EU is not responsible for any use that may be made of the information it contains. This work was partly supported by the NCCR MARVEL, funded by the Swiss National Science Foundation.

Chapter 8

Conclusion

In this work the reader was walked trough the process of quantum machine learning applied to chemical reaction space. The first two chapters (**Chapter 2** and **3**) gave a brief introduction to quantum chemistry and quantum machine learning, respectively.

Chapters 4 and **5** were devoted to the data generation. In **Chapter 4** the scheduling of quantum chemical calculations on small (eg. University) and large (eg. HPC) computing clusters was analyzed. Using machine learning models developed to study equilibrium properties of small organic molecules, it was possible to learn the CPU times of quantum chemical calculations. Since the CPU time of a single point calculation scales with the number of electrons, prediction errors of $\sim$ 1% were reached, which was to be expected consider the linear decay w.r.t. the training set size for the FLOPs (Figure 4.8). However, a geometry optimization or a transition state search deliver a much harder learning problem because they are a summary of single point energy and gradient calculations. Here, the CPU time also depends on the initial geometry, meaning, the closer the geometry is to the minimum or the saddle point, the fewer optimization steps needed and the shorter the calculation time. Machine learning models based on geometrical representation could learn the distance to the local minima or the saddle point and they reached prediction errors of run times of $\sim$20%. Although, the errors for GO and TS calculations were significantly higher than for the SP calculations, it was sufficient to optimized scheduling of thousands of calculations. Using these CPU time predictions allowed for better scheduling than random picking calculations by the cluster and improving the time to solution by 10 to 90%. The term Green Chemistry has its origins in the "Pollution Prevention Act" of 1990[279], where the pollution should be reduced by improved design, eg. less waste/side products in chemical reactions or cleaner/less harmful solvents. Reducing the time to solution and the number of idle cores on a super computer saves electricity and is therefore a new and innovative way, how machine learning models can be part in green chemistry.

Chapter 5 dealt with the data generation itself. Machine learning models requires thousands of training data which are rare in literature for chemical reactions. Therefore, a data set for the two text book reactions E2 vs. S_N2 was generated on an MP2/6-311G(d) level of theory. Thousands of reactants and transition states were reported for these two reactions resulting 4'466 activation barriers. Using a Δ-ML approach, the data set was extended with LD-LCCSD/cc-TZVP single point energies and barriers.

Although, thousands of CPU hours were spend to generate the data set, the tricky nature of transition state calculations and the exhaustiveness of conformer searches on an MP2 level of theory prevented the completion of the data set. Therefore, in a next approach (**Chapter 6**), the R2B machine learning model was introduced, where common geometry based representation, as well as one-hot encoding were used to complete the data set. To avoid the difficult transition state calculations, solely the reactants were used as input for the ML model, which was sufficient to predict activation barriers with MAE's of 2 to 2.5 kcal/mol. Using these ML predictions, decision trees showing the largest changes in activation energy by varying the substituents were generated, which could support experimental reaction design. Furthermore, the one-hot encoding representation was used to learn key geometrical properties, such as distances, angles, and dihedrals, to study Hammond's postulate. For this data set, Hammond's postulate is only applicable to the S_N2 reaction but not to the E2 reaction.

Finally, in **Chapter 7**, the geometry optimization, as well as the transition state search using machine learned energies (as done so far) and forces (new) was investigated. Using the operator quantum machine learning approach, ML models could be used together with legacy optimizer, such as LBFGS or the Berny algorithm for geometry optimizations and transition state searches, respectively. Applying these models, similar accuracies as DFT calculations could be reached in terms of RMSD's or transition state frequencies.

Chapter 9

Appendix A

9.1 Atomic Units[22]

$$\left[-\frac{\hbar^2}{2m_e}\nabla^2 - \frac{e^2}{4\pi\epsilon_0 \mathrm{r}}\right]\phi = \mathscr{E}\phi$$

$$\left[-\frac{\hbar^2}{2m_e\lambda^2}\nabla^2 - \frac{e^2}{4\pi\epsilon_0 \lambda \mathrm{r}'}\right]\phi = \mathscr{E}\phi$$

$$\frac{\hbar^2}{m_e\lambda^2} = \frac{e^2}{4\pi\epsilon_0\lambda} = \mathscr{E}_a$$

$$\lambda = \frac{4\pi\epsilon_0\hbar^2}{m_e e^2} = a_0$$

$$\mathscr{E}_a\left[-\frac{1}{2}\nabla'^2 - \frac{1}{\mathrm{r}'}\right]\phi' = \mathscr{E}\phi'$$

$$\left(-\frac{1}{2}\nabla'^2 - \frac{1}{\mathrm{r}'}\right)\phi' = \mathscr{E}'\phi'$$

9.2 Time independent Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}\psi(t,\mathbf{r}) = \left(-\frac{\hbar^2}{2m}\Delta + V(\mathbf{r})\right) = \hat{H}\psi(t,\mathbf{r}) \tag{9.1}$$

Using the Ansatz: $\Psi(t,\mathbf{r}) = \psi(\mathbf{r}) \cdot f(t)$.

$$i\hbar\psi(\mathbf{r})\frac{\partial f(t)}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi(\mathbf{r})}{\partial \mathbf{r}^2}f(t) + V(\mathbf{r})\psi(\mathbf{r})f(t) \qquad |\cdot\frac{1}{\psi(\mathbf{r})f(t)} \tag{9.2}$$

$$i\hbar\frac{1}{f(t)}\frac{\partial f(t)}{\partial t} = -\frac{\hbar^2}{2m}\frac{1}{\psi(\mathbf{r})}\frac{\partial^2\psi(\mathbf{r})}{\partial \mathbf{r}^2} + V(\mathbf{r}) \tag{9.3}$$

Since the left side of the equation only depends on t and the right side of the equation only depends on $\mathbf{r}$, both sides must be equal to a constant, which we denote E. We can solve each side independently. For the left side of the equation we get:

$$i\hbar\frac{1}{f(t)}\frac{\partial f(t)}{\partial t} = E \tag{9.4}$$

$$i\hbar\frac{\partial f(t)}{\partial t} = f(t)E \tag{9.5}$$

$$f(t) = Ce^{-iEt/\hbar} \tag{9.6}$$

For the right side of the equation we get:

$$-\frac{\hbar^2}{2m}\frac{1}{\psi(\mathbf{r})}\frac{\partial^2\psi(\mathbf{r})}{\partial\mathbf{r}^2} + V(\mathbf{r}) = E \tag{9.7}$$

$$-\frac{\hbar^2}{2m}\frac{\partial^2\psi(\mathbf{r})}{\partial\mathbf{r}^2} + V(\mathbf{r})\psi(\mathbf{r}) = E\psi(\mathbf{r}) \tag{9.8}$$

$$\hat{H}\psi(\mathbf{r}) = E\psi(\mathbf{r}) \tag{9.9}$$

With the total wavefunction being:

$$\Psi(t,\mathbf{r}) = \psi(\mathbf{r}) \cdot Ce^{-iEt/\hbar} \tag{9.10}$$

9.3 Born Opennheimer approximation

Since the nuclei are much heavier than the electrons the wavefunction can be separated in a product Ansatz:

$$\Psi_k(\{\mathbf{r}_i\},\{\mathbf{R}_I\}) \approx \chi_{k,n}(\{\mathbf{r}_i\}) \cdot \psi_{\mathrm{el},n}^{\{\mathbf{R}_I\}}(\{\mathbf{r}_i\}) \tag{9.11}$$

inserting this Ansatz in the SDE results in:

$$\hat{H}\chi_{k,n}\psi_{\mathrm{el},n} = E\chi_{k,n}\psi_{\mathrm{el},n} \tag{9.12}$$

The total hamilton operator $\hat{H}$ can be split into an electronic ($\hat{H}_{\mathrm{el}}$) and a core hamilton operator only depending on the kinetic operator of the nuclei ($\hat{T}_{\mathrm{K}}$):

$$\hat{H} = \hat{H}_{\mathrm{el}} + \hat{T}_{\mathrm{K}} \tag{9.13}$$

inserting the hamolton operator in equation 6.18 gives:

$$\hat{T}_{\mathrm{K}}\chi_{k,n}\psi_{\mathrm{el},n} + \hat{H}_{\mathrm{el}}\chi_{k,n}\psi_{\mathrm{el},n} = E\hat{H}_{\mathrm{el}} \tag{9.14}$$

multiply from the left side with $\psi^*_{\mathrm{el},n}$:

$$\hat{T}_{\mathrm{K}}\chi_{k,n}\langle\psi_{\mathrm{el},n}|\psi_{\mathrm{el},n}\rangle + \langle\psi_{\mathrm{el},n}|\hat{H}_{\mathrm{el}}|\psi_{\mathrm{el},n}\rangle\chi_{k,n} = E_k\chi_{k,n}\langle\psi_{\mathrm{el},n}|\psi_{\mathrm{el},n}\rangle \tag{9.15}$$

Chapter 10

Appendix B

This material is available free of charge via the Internet at `http://pubs.acs.org/`. Code and raw data is available on GitHub `https://github.com/ferchault/mlscheduling`

10.1 Additional details on the data sets

Table 10.1 shows additional information regarding the used hardware.

10.1.1 $\mathbf{QMrxn^{GO}_{MP2}}$

The initial reactant geometries from the reaction data set were obtained by generating the unsubstituted molecule (hydrogen atoms instead of functional groups and Fluor as leaving group) without the nucleophile. Subsequently substituting the hydrogen atoms with functional groups span the chemical space. For every reactant a conformer search on PM6-D3 level was performed using ORCA. The lowest lying conformer geometries were then further optimized on MP2/6-31G* level of theory which resulted in the data set set $\mathbf{QMrxn^{GO}_{MP2}}$.

10.1.2 $\mathbf{QMrxn^{TS}_{MP2}}$

The starting geometries for the transition state (TS) search were obtained in a similar way as described in section 10.1.1. A transition state search was performed on the

Calculation	**SP**			**GO**			**TS**
Data set	QM9		QMspin	QM9	QMrxn	QMspin	QMrxn
# Cores	24	24	1	1	1	1	1
CPU Types	E5-2680v3	E5-2680v3	E5-2650v2 E5-2640v3 E5-2630v4	E5-2630v4	E5-2640v3 E5-2650v4	E5-2650v2 E5-2640v3 E5-2630v4	E5-2650v2 E5-2680v3 E5-2640v3 E5-2630v4 E5-2650v4
σ_{BoB}	204.8	204.8	51.2	51.2	102.4	51.2	204.8
σ_{FCHL}	12.8	12.8	51.2	409.6	51.2	51.2	51.2
λ_{BoB}	1e-7	1e-7	1e-7	1e-7	1e-7	1e-5	1e-7
λ_{FCHL}	1e-7	1e-7	1e-7	1e-7	1e-9	1e-5	1e-7

TABLE 10.1: Data sets of calculations used in this work: Software used for calculations, number of cores used per calculation, and CPU types the calculation ran on as well as details of the ML hyperparameter.

unsubstituted case and from the found TS the chemical space was spanned by exchanging the hydrogen atoms with functional groups. The following timings (using ORCA 4.0.1) and the initial geometries of the TS search form the data set $\mathbf{QMrxn}^{\mathbf{TS}}_{\mathbf{MP2}}$.

10.1.3 $\mathbf{QM9}^{\mathbf{GO}}_{\mathbf{B3LYP}}$

The QM9 data set contains geometries optimized with B3LYP/6-31G*. 5001 out of these 134k molecules were further optimized with a larger basis set (def2-TZVP) using Molpro to obtain data set $\mathbf{QM9}^{\mathbf{GO}}_{\mathbf{B3LYP}}$.

10.1.4 $\mathbf{QMspin}^{\mathbf{SP}}_{\mathbf{MRCI}}$ and $\mathbf{QMspin}^{\mathbf{GO}}_{\mathbf{CASSCF}}$

For the geometry optimization of data set $\mathbf{QMspin}^{\mathbf{GO}}_{\mathbf{CASSCF}}$ we use the CASSCF single point energy[127] and energy gradient implementation[128] in Molpro. The calculations have been run on one compute core per job and similar amounts of run time are spent for the wave function computation and the energy gradient. When performing a geometry optimization, the CASSCF wave function of a previous step is used as a starting guess for the CASSCF wave functions of the new geometry. For that reason, the first step of a geometry optimization takes significantly longer than the following steps. We take this aspect into account in our ML model as well as in our scheduling model.

10.2 First fit decreasing algorithm

The bin packing problem is NP-hard[51], i. e., the search for the optimal solution to the problem is prohibitively expensive for real-world workloads of thousands of jobs even if the time estimates were arbitrarily accurate[280–282]. First Fit Decreasing (FFD) is one of the many heuristic algorithms[283] that exists for the bin packing problem. It has been shown that for practical purposes the FFD algorithm is close to the optimal solution, as for q compute jobs as it uses at most $11/9q + 1$ jobs[284], but typically is within a few percent of the optimal solution[282]. In all cases, we calculate the total core hours and the total duration from the first to the last job. The total core hours divided by the sum of the real run times define the compute overhead. The total duration from first to last job should not be exceedingly high compared to other approaches, since this metric is about enabling science: if the calculations would take too long, a research project would not be started. The ideal approach therefore reduces the overhead while keeping the total wall time at least comparable to established approaches.

10.3 Learning curves

In the following we compare models with respect to different training inputs. We trained models on CPU, normalized (by number of electrons) CPU, wall, and normalized wall times. For CPU times only calculations done with Molpro could be considered because ORCA output files only contain total (wall) times. Best performance was reached with models trained on normalized CPU times. The differences are small (around 1% to 4%). For predictions normalized wall times were used because of their application to the scheduling. For the test sets **QMspin$^{\mathbf{SP}}_{\mathbf{MRCI}}$** and **QMspin$^{\mathbf{GO}}_{\mathbf{CASSCF}}$** we also training on CPU times normalized by the formal scaling of the method, this did neither lead to significant changes in the training model (results not shown).

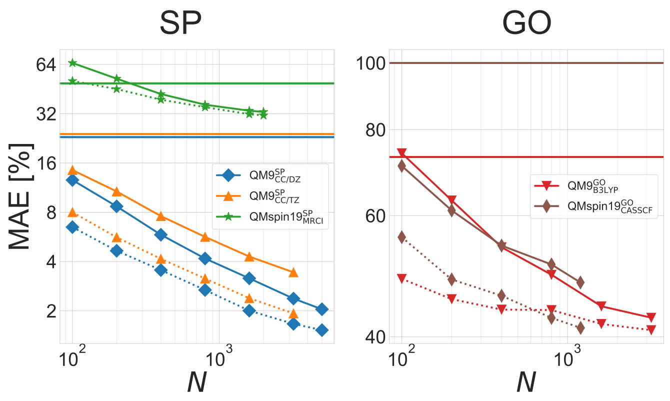


FIGURE 10.1: Learning curves showing normalized test errors (cross validated MAE divided by median of test set) using BoB and FCHL as representations. The model was trained on CPU times. Horizontal lines correspond to the performance assuming all calculations have mean run time (standard deviation divided by the mean wall time of the data set.

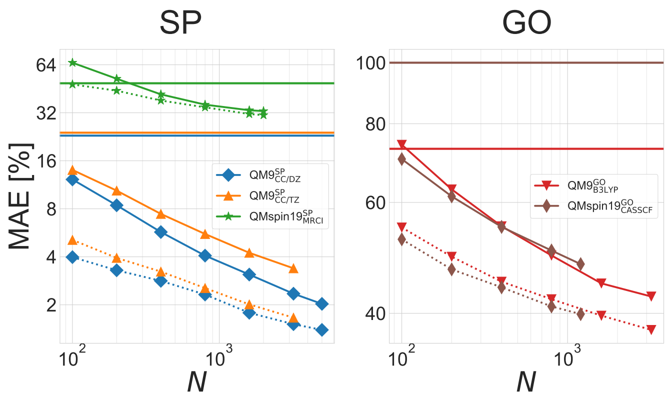


FIGURE 10.2: Learning curves showing normalized test errors (cross validated MAE divided by median of test set) using BoB and FCHL as representations. The model was trained on normalized (by number of electrons) CPU times. Horizontal lines correspond to the performance assuming all calculations have mean run time (standard deviation divided by the mean wall time of the data set.

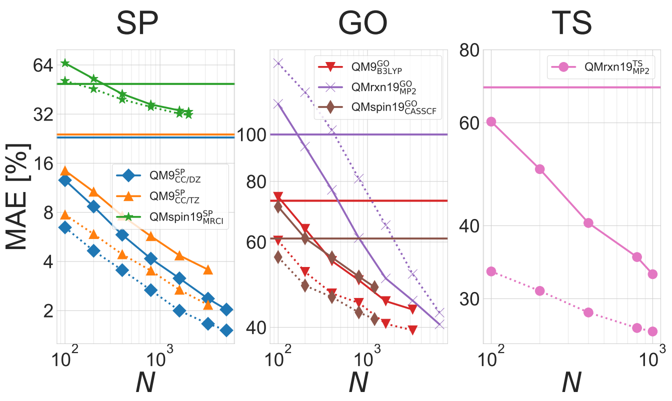


FIGURE 10.3: Learning curves showing normalized test errors (cross validated MAE divided by median of test set) using BoB and FCHL as representations. The model was trained on wall times. Horizontal lines correspond to the performance assuming all calculations have mean run time (standard deviation divided by the mean wall time of the data set.

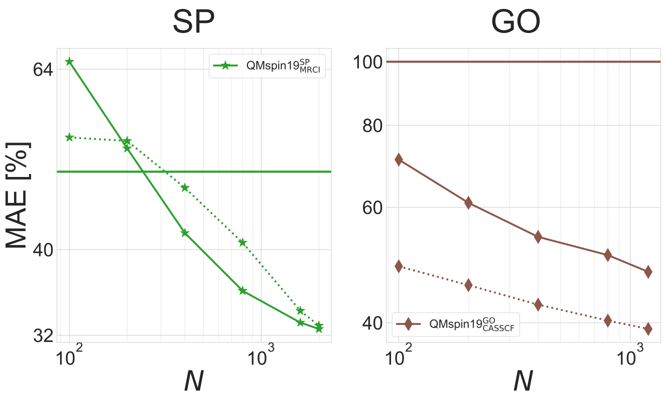


FIGURE 10.4: Learning curves showing normalized test errors (cross validated MAE divided by median of test set) using BoB and FCHL as representations. The model was trained on normalized (number of occupied orbitals to the power of 2 times number of basis functions to the power of 4) wall times. Horizontal lines correspond to the performance assuming all calculations have mean run time (standard deviation divided by the mean wall time of the data set.

Chapter 11

Appendix C

11.1 Learning Barriers

To predict the activation barriers (E_a), we optimized the hyperparameters using a five fold cross validation on the training set for different combinations of σ and λ as described in the methods section of the manuscript. Figure 11.1 shows the results of the hyperparameter optimization for learning the barriers as a heat map of the different σ and λ combinations with the MAE encoded in the color map. Table 11.1 and 11.2 contain the values used to generate the learning curves for the E2 and the S_N2 reaction, respectively. Figure 11.2 shows learning curves for all models using the reactants (solid lines) and the reactant complexes (dashed lines) as input for the QML model. Using the reactant complexes as input yields slightly better results, but the difference in learning is negligible. Reactant complexes as well as reactants were canonicalized according to the lowest lying geometry to make the learning problem unique.

11.2 Learning Geometries

Table 11.3 and 11.5 contain the data used to generate the learning curves for the S_N2 and the E2 reaction, respectively.

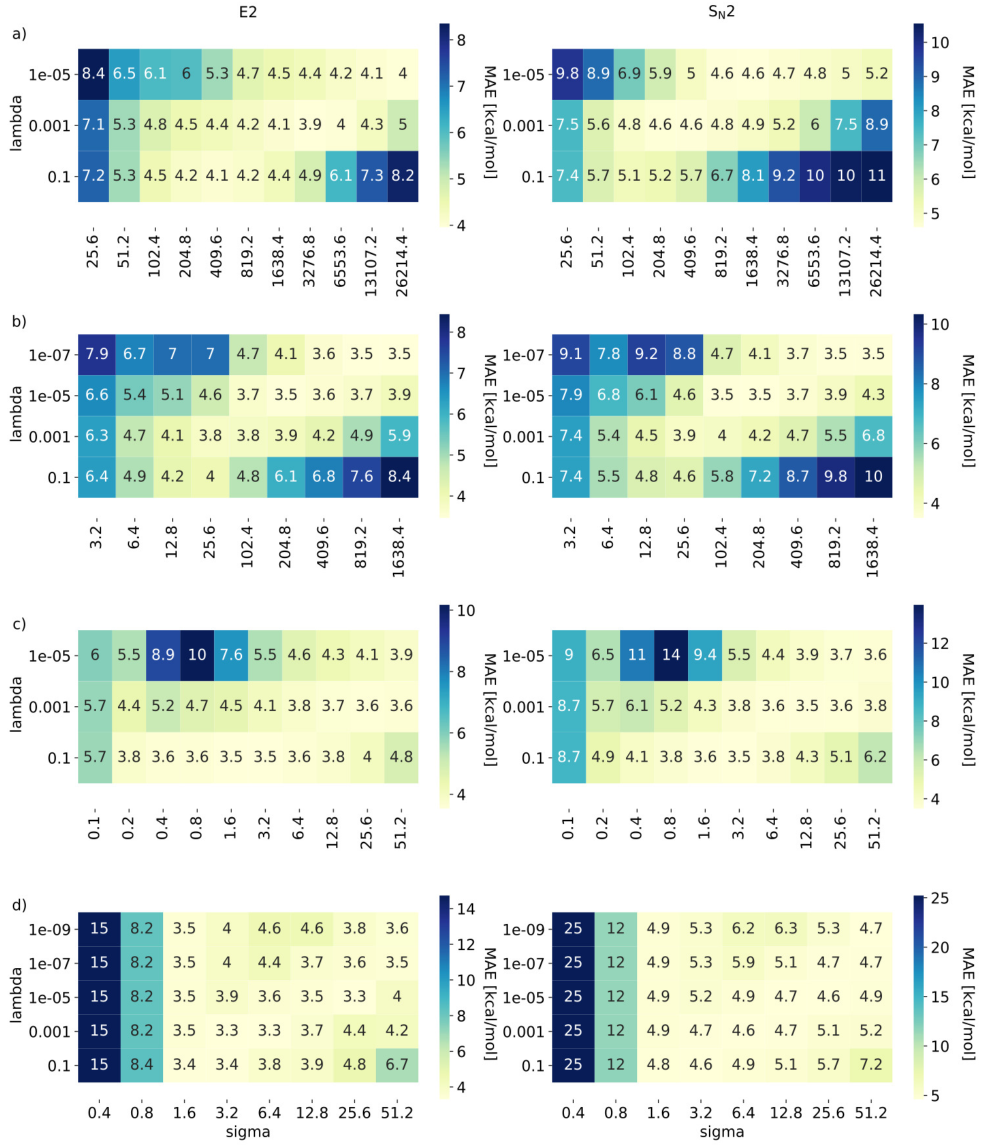


FIGURE 11.1: Hyperparameter optimization of R2B models: a) BoB, b) SLATM, c) FCHL19, and d) one-hot encoding for E2 (left) and S$_N$2 (right) reactions.

Representation (λ/σ)	N	MAE [kcal/mol]
BoB (1-05/1638.4)	50	5.48
	100	4.90
	200	4.40
	400	3.86
	800	3.53
SLATM (1e-05/204.8)	50	5.42
	100	4.32
	200	3.97
	400	3.36
	800	3.06
FCHL19 (0.1/1.6)	50	5.28
	100	4.65
	200	3.87
	400	3.32
	800	2.95
one-hot encoding (0.001/6.4)	50	4.29
	100	3.55
	200	3.32
	400	2.86
	800	2.53

TABLE 11.1: Results from R2B models used to generate learning curves for the E2 reaction.

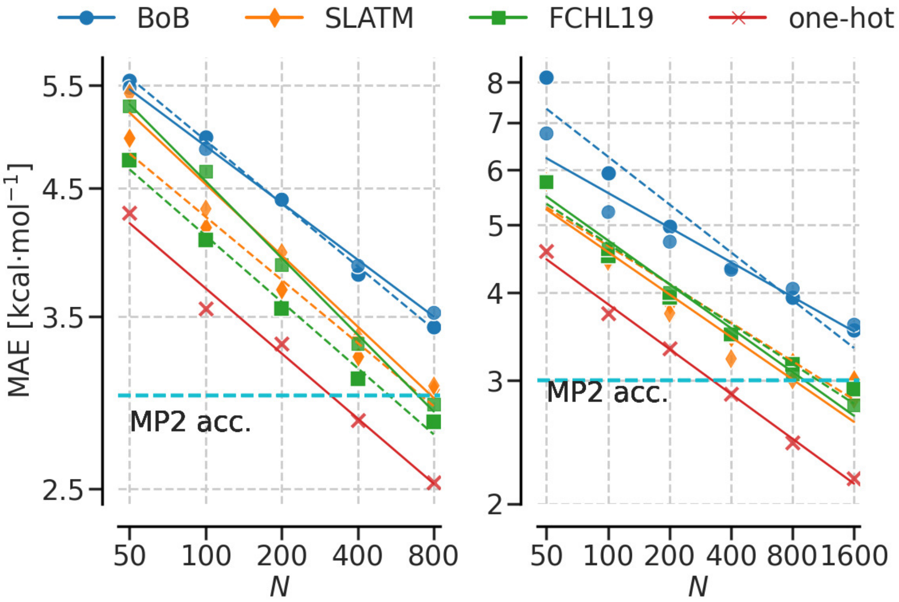


FIGURE 11.2: **Learning curves** Activation energy prediction errors (out-of-sample) as a function of training set size N for activation barriers (E_a) of E2 a) and S_N2 b) using reactant complex geometries (dashed lines) and reactants (solid lines) as inputs only. Results are shown for four representations (BoB, SLATM, FCHL19, one-hot) used within KRR models. Training data reference level of theory corresponds to DF-LCCSD/cc-pVTZ//MP2/6-311G(d), and estimated MP2 error is denoted as a green dashed horizontal line.

Representation (λ/σ)	N	MAE [kcal/mol]
BoB (1e-05/1638.4)	50	6.75
	100	5.22
	200	4.73
	400	4.32
	800	4.06
	1600	3.60
SLATM (1e-05/204.8)	50	5.76
	100	4.43
	200	3.74
	400	3.22
	800	3.00
	1600	2.82
FCHL19 (0.1/1.6)	50	5.75
	100	5.63
	200	4.00
	400	3.49
	800	3.06
	1600	2.76
one-hot encoding (0.01/6.4)	50	4.58
	100	3.74
	200	3.33
	400	2.87
	800	2.44
	1600	2.17

TABLE 11.2: Results from R2B models used to generate learning curves for the S_N2 reaction.

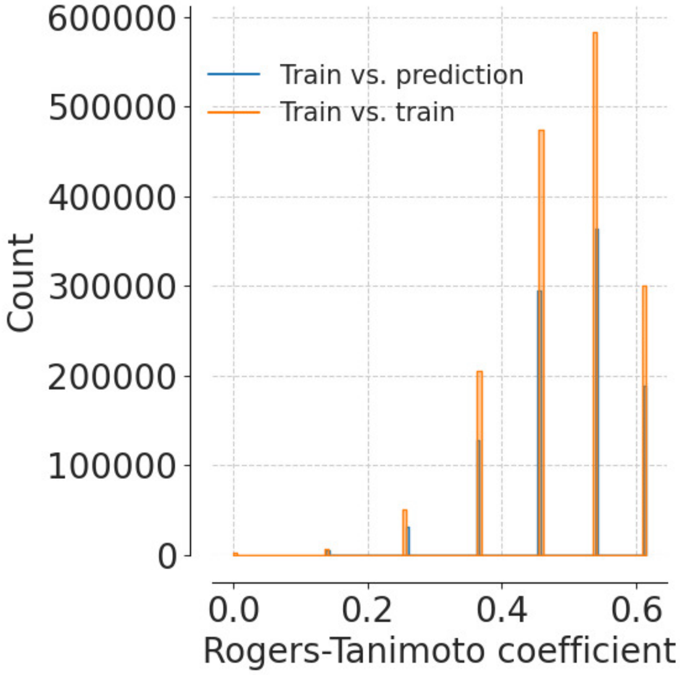


FIGURE 11.3: **Rogers-Tanimoto coefficients** Similarity check for the one hot encoding between the training and test molecules using the Rogers-Tanimoto coefficient. Two binary representations are equal and inverse if their coefficient is 0 and 1, respectively.

Parameter (λ & σ)	N	MAE [kcal/mol]
Reactant d_{x} (0.1/3.2)	225	0.077
	450	0.036
	900	0.019
	1800	0.012
Reactant d_{y} (0.1/1.6)	225	1.04
	450	0.61
	900	0.36
	1800	0.25
Reactant α (0.5/3.2)	225	7.47
	450	4.71
	900	3.63
	1800	3.11
TS d_{x} (0.1/3.2)	225	0.010
	450	0.053
	900	0.035
	1800	0.026
TS d_{y} (0.1/3.2)	225	0.146
	450	0.098
	900	0.079
	1800	0.070
TS α (0.1/3.2)	225	6.73
	450	4.27
	900	3.26
	1800	2.89

TABLE 11.3: Results from R2B models used to generate learning curves for the S_N2 reaction. Learning of the reactant complex and transition state geometries.

Parameter (λ & σ)	N	MAE [kcal/mol]
Reactant d_x (0.1/3.2)	125	0.136
	250	0.073
	500	0.044
	1000	0.032
Reactant d_y (0.1/3.2)	125	0.226
	250	0.134
	500	0.078
	1000	0.052
Reactant $d\alpha$ (0.1/3.2)	125	8.10
	250	4.36
	500	2.63
	1000	1.87
Reactant β (0.1/3.2)	125	8.45
	250	5.62
	500	4.39
	1000	3.72
Reactant θ (0.1/3.2)	125	85.67
	250	83.48
	500	82.05
	1000	74.79

TABLE 11.4: Results from R2B models used to generate learning curves for the E2 reaction. Learning of reactant complexes.

Parameter (λ & σ)	N	MAE [kcal/mol]
TS d_x (0.1/3.2)	125	0.158
	250	0.094
	500	0.056
	1000	0.036
TS d_y (0.1/1.6)	125	0.203
	250	0.120
	500	0.071
	1000	0.046
TS α (0.1/3.2)	125	8.26
	250	4.25
	500	2.47
	1000	1.59
TS β (0.1/3.2)	125	9.92
	250	7.27
	500	6.04
	1000	5.13
TS θ (0.1/3.2)	125	16.66
	250	11.96
	500	9.73
	1000	8.79

TABLE 11.5: Results from R2B models used to generate learning curves for the E2 reaction. Learning of transition state geometries.

Chapter 12

Appendix D

12.1 Data Sets

12.1.1 Constitutional Isomers

The constitutional isomers are part of the QM9 data set[67] with the sum formula $C_7O_2H_{10}$. The data set contains 6095 compounds at the B3LYP/6-31G(p,2df) level of theory (Gaussian09[205]).

For this work, all geometries were optimized with OpenBabel[198] using the UFF force field[199] (truncated after 200 steps). Subsequently, the geometries were re-optimized using the ORCA 4.0[130] electronic structure code at the B3LYP/6-31G(2df) level of theory.

12.1.2 Nucleophilic Substitution Reaction (S_N2)

The nucleophilic substitution reactions (S_N2) are a subset from the QMrxn20[3,240] data set. As described in Figure 1, the scaffold of the molecules is ethane which was substituted with leaving groups X, nucleophiles Y^-, and functional groups R1-4. The data set contains 1807 reactions consisting of reactant complexes and transition states on an MP2/6-311G(d) level of theory. Out of these reactions, 200 were randomly chosen for training and the test set contains 300 out of sample compounds (reactant complexes and transition states).

12.2 Model Training

To obtain the hyperparameters σ and λ a five fold cross validation over a range of both hyperparameters within the training set was performed. The hyperparameter were not optimized on the out of sample test set. The best performing model, lowest MAE of energies and forces was chosen for the training of the model.

12.2.1 Constitutional Isomers

The 6095 compounds were split into 5595 training and 500 testing molecules. Then, randomly picked geometries from the optimization steps were displaced using their normal modes yielding 6000 training instances with energies and forces. Normal modes were calculated using the *freq=hpmodes* keyword from the Gaussian09 code. Energies and gradients were calculated using the ORCA 4.0 code (*engrad* keyword).

For the geometry relaxation UFF geometries as starting points were used and the LBFGS algorithm as implemented in the ASE[285] python package was used together

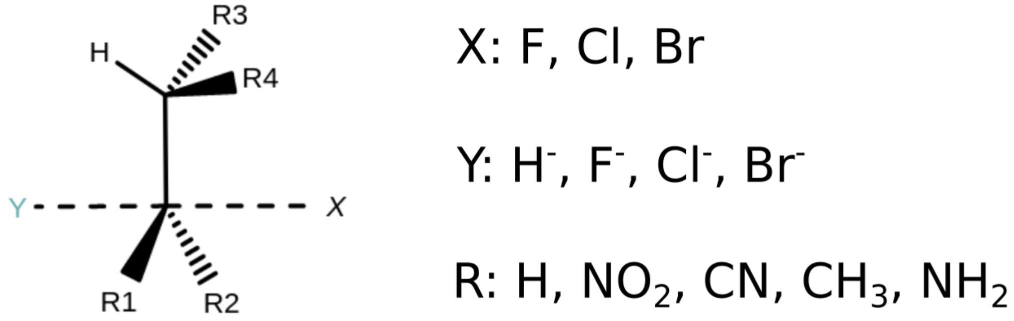

R3
H
R4
Y
X
R1
R2
X: F, Cl, Br
Y: H$^-$, F$^-$, Cl$^-$, Br$^-$
R: H, NO_2, CN, CH_3, NH_2

FIGURE 12.1: Example ethane scaffold of an S_N2 reactions (left) with leaving groups X, nucleophiles Y, and functional groups R1-4 (right).

with the ORCA 4.0 calculator implemented in ASE. For the convergence thresh hold (*fmax*, maximum force of all atoms), the default value of the ASE package was used (0.05 eV/Å) and the geometry optimization was truncated after 50 iterations. Example scripts can be found in[270].

12.2.2 Reactant Complexes

For the reaction data set in QMrxn20, the reactant complexes (training set) were optimized with the UFF force field (truncated after 200 steps) and subsequent geometry optimizations on the MP2/6-311G(d)[188–192] level of theory were performed using the ORCA 4.0 code. For the training set, 200 reactions were randomly chosen with their optimization steps. For these reactant complexes, random geometries along the optimization paths were chosen and displaced along their normal modes (using Gaussian09 *freq=hpmodes*), yielding 3753 training instances for the reactant complexes. Energies and gradients were calculated using the *engrad* keyword from the ORCA 4.0 package.

12.2.3 Transition States

For the transition state training data, reactant and product complexes from the QMrxn20[3] data set were optimized using Openbabel's UFF force field (truncated after 200 steps). Then, a transition sate search for the 200 training instances was performed using Gaussian09 *QST2* keyword (berny algorithm) and the *loose* keyword. On the geometries along the transition state search path normal mode calculations using the *freq=hpmodes* from the Gaussian09 code were performed yielding 3812 training instances for the transition states. Energies and gradients were obtained using ORCA 4.0 *engrad* keyword.

12.3 Optimization

12.3.1 Geometry Relaxation

For the geometry relaxation (constitutional isomers and reactant complexes), the ASE code and the LBFGS algorithm with the ORCA 4.0 calculator were used. For the OQML models a machine learning calculator yielding forces and energies when given a geometry was implemented in ASE. Example scripts can be found in[270]

12.3.2 Transition State Search

For the transition state search the Gaussian09 package was used which allows for external energy and force calculations with the keywords *QST2* and *loose*. Example scripts can be found in[270].

12.4 Validation

12.4.1 Constitutional Isomers

The validation criterion for the constitutional isomers was the convergence after 50 LBFGS (ASE) steps with the thresh hold of *fmax*=0.05 eV/Å.

12.4.2 Reactant Complexes

In addition to the convergence criteria for the LBFGS optimization, the fragments for the reactant complexes were analyzed. For every reactant complex there should be two fragments, the main molecule and the nucleophile Y^- containing only one atom.

12.4.3 Transition States

The transition state validation contains multiple tests:

1. *normal termination* of the Gaussian09 code
2. 1 imaginary frequency < 100 cm^{-1} (value derived from[3])
3. Y–C–X (nucleophile–reaction center–leaving group) angle > 155° (see Figure 2 a)
4. minimal distance of 0.9 Å between atoms (see Figure 2 b)
5. Correct movement of the reaction center (see Figure 2 c) for the first normal mode
6. Movement of remaining normal modes < 0.5Å (see Figure 2 d)

Displaced geometries for 5 and 6 were obtained using the vibration package from the ASE code (example script can be found in[270]). Only if a compound passes all tests, it is considered in the subsequent analysis of RMSD's and frequencies.

12.4.4 Success Rate

Success rates for the ML models and the DFT calculations are shown in Table 1. The success rate is calculated by 100 divided by the total number of optimizations (300) times the validated compounds.

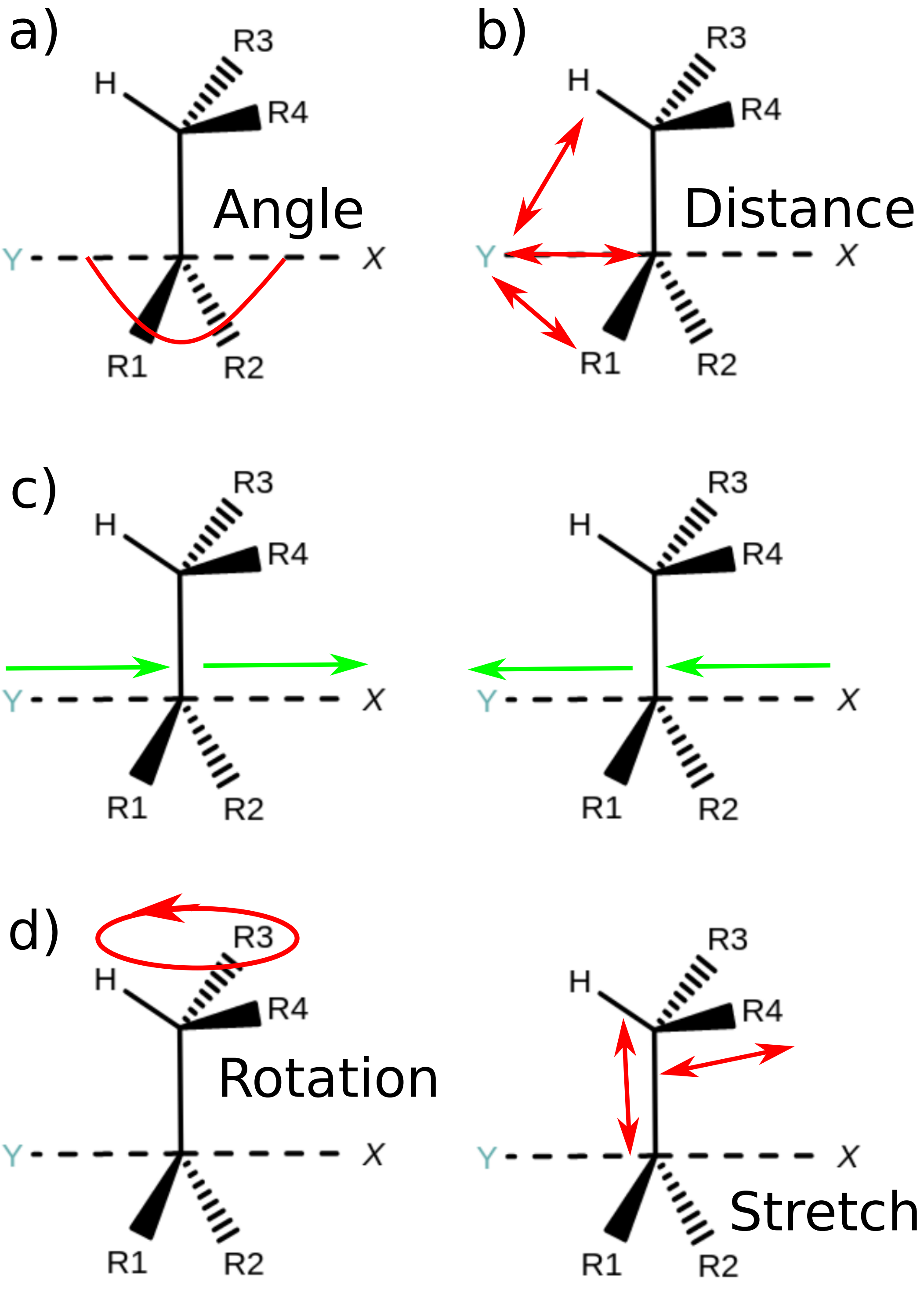


FIGURE 12.2: Validation for transition states a) Angle of reaction center, b) minimal distance of 0.9Å between the atoms, c) movement of atoms in reaction center using the first normal mode, and d) movement of all other atoms and normal modes.

Method	Success rate (GS)	Success rate (TS)
OQML (FCHL19)	28%	64.66%
B3LYP/6-311G(d)	39.33%	65.33%
PBE0/6-311G(d)	41%	73.66%
ωB97X/6-311G(d)	46.66%	70%

TABLE 12.1: Success rates for geometry optimizations of ground states (GS) and transition state searches (TS) for the OQML model and the three DFT methods.

12.4.5 RMSD's

For every validated constitutional isomer, reactant complex, and transition state RMSD's w.r.t. the MP2 reference compounds were calculated using the python RMSD code[268] with the Kabsch algorithm including hydrogens (except for the constitutional isomers).

12.4.6 Frequencies

The vibrational frequencies of the transition state and the reactant complexes were obtained using the vibration package from the ASE code with the respective method using the ORCA5[286] calculator.

12.4.7 Activation Energies

For the activation energies, the validated geometries of reactant complexes and transition states were taken and single point calculations using the ORCA5[286] code were performed. Similarly, the same OQML models used for the optimization, were used to calculate the energies of reactant complexes and transition states.

Bibliography

[1] B. Meyer, B. Sawatlon, S. Heinen, O. A. von Lilienfeld, C. Corminboeuf, *Chem. Sci.* **2018**, *35*, 7069 –7077.

[2] S. Heinen, M. Schwilk, G. F. von Rudorff, O. A. von Lilienfeld, *Machine Learning: Science and Technology* **2020**, *1*, 025002.

[3] G. F. von Rudorff, S. Heinen, M. Bragato, A. von Lilienfeld, *Machine Learning: Science and Technology* **2020**, DOI `10.1088/2632-2153/aba822`.

[4] S. Heinen, G. F. von Rudorff, O. A. von Lilienfeld, *The Journal of Chemical Physics* **2021**, *155*, 064105.

[5] A. Merkel, Z. Havlas, R. Zahradnik, *Journal of the American Chemical Society* **1988**, *110*, 8355–8359.

[6] E. Track, N. Forbes, G. Strawn, *Comput. Sci. Eng.* **2017**, *19*, 4–6.

[7] O. A. von Lilienfeld, K.-R. Müller, A. Tkatchenko, *Nature Reviews Chemistry* **2020**, 1–12.

[8] List of top 500 super computers `http://www.top500.org`, Accessed: 24/04/19.

[9] F. Strieth-Kalthoff, F. Sandfort, M. H. S. Segler, F. Glorius, *Chemical Society Reviews* **2020**, *49*, 6154–6168.

[10] M. A. Kayala, C.-A. Azencott, J. H. Chen, P. Baldi, *Journal of Chemical Information and Modeling* **2011**, *51*, 2209–2222.

[11] J. N. Wei, D. Duvenaud, A. Aspuru-Guzik, *ACS Central Science* **2016**, *2*, 725–732.

[12] W. Jin, C. Coley, R. Barzilay, T. Jaakkola in *Advances in Neural Information Processing Systems 30*, (Eds.: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett), Curran Associates, Inc., **2017**, pp. 2607–2616.

[13] D. Fooshee, A. Mood, E. Gutman, M. Tavakoli, G. Urban, F. Liu, N. Huynh, D. V. Vranken, P. Baldi, *Molecular Systems Design & Engineering* **2018**, *3*, 442–452.

[14] M. H. S. Segler, M. Preuss, M. P. Waller, *Nature* **2018**, *555*, 604–610.

[15] P. Schwaller, T. Gaudin, D. Lányi, C. Bekas, T. Laino, *Chemical Science* **2018**, *9*, 6091–6098.

[16] C. A. Grambow, L. Pattanaik, W. H. Green, *Scientific Data* **2020**, *7*, 1–8.

[17] O. T. Unke, M. Meuwly, SN2 reactions, **2019**.

[18] R. Ramakrishnan, P. O. Dral, M. Rupp, O. A. von Lilienfeld, *Journal of Chemical Theory and Computation* **2015**, *11*, 2087–2096.

[19] A. S. Christensen, F. A. Faber, O. A. von Lilienfeld, *The Journal of Chemical Physics* **2019**, *150*, 064105.

[20] A. S. Christensen, O. A. von Lilienfeld, *Machine Learning: Science and Technology* **2020**, *1*, 045018.

[21] I. N. Levine, *Quantum Chemistry*, Prentice-Hall Inc., New Jersey (USA), **2000**.

[22] A. Szabo, N. S. Ostlund, *Modern Quantum Chemistry*, Dover Publications, Inc., Mineola, First edition, **1969**.

[23] M. Steinhauser, *Quantenmechanik für Naturwissenschaftler*, Springer Berlin, **2016**.

[24] W. Wien, *The London Edinburgh and Dublin Philosophical Magazine and Journal of Science* **1897**, *43*, 214–220.

[25] A. Einstein, *Annalen der Physik* **1905**, *322*, 132–148.

[26] M. Born, *Zeitschrift für Physik* **1924**, *26*, 379–395.

[27] W. A. Fedak, J. J. Prentis, *American Journal of Physics* **2009**, *77*, 128–139.

[28] L. D. BROGLIE, *Nature* **1923**, *112*, 540–540.

[29] W. Heisenberg, *Monatshefte für Mathematik und Physik* **1931**, *38-38*, 365–372.

[30] I. N. Levine, *Quantum Chemistry*, Prentice-Hall Inc., New Jersey (USA), **2000**, pp. 2–13.

[31] E. Schrödinger, *Annalen der Physik* **1926**, *384*, 361–376.

[32] W. Pauli, *Zeitschrift für Physik* **1925**, *31*, 765–783.

[33] I. N. Levine, *Quantum Chemistry*, Prentice-Hall Inc., New Jersey (USA), **2000**, p. 197.

[34] E. R. Davidson, D. Feller, *Chemical Reviews* **1986**, *86*, 681–696.

[35] I. N. Levine, *Quantum Chemistry*, Prentice-Hall Inc., New Jersey (USA), **2000**, p. 541.

[36] I. N. Levine, *Quantum Chemistry*, Prentice-Hall Inc., New Jersey (USA), **2000**, p. 526.

[37] E. R. Davidson, *J. Comp. Phys.* **1975**, *17*, 87 –94.

[38] I. N. Levine, *Quantum Chemistry*, Prentice-Hall Inc., New Jersey (USA), **2000**, pp. 552–554.

[39] **2009**, DOI `10.1002/0470862106.ia615`.

[40] S. H. Vosko, L. Wilk, M. Nusair, *Canadian Journal of Physics* **1980**, *58*, 1200–1211.

[41] V. Vapnik, *The nature of statistical learning theory*, Springer science & business media, **2013**.

[42] F. Rosenblatt, **1957**.

[43] F. Rosenblatt, *Proceedings of the IRE* **1960**, *48*, 301–309.

[44] W. McCulloch W.S., Pitts, *Bulletin of Mathematical Biophysics* **1943**, *5*, 115–133.

[45] H. G. W. R. Rumelhart D., *Nature* **1986**, *323*, 533–536.

[46] M. Campbell, A. Hoane, F. hsiung Hsu, *Artificial Intelligence* **2002**, *134*, 57–83.

[47] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, *Nature* **2016**, *529*, 484–489.

[48] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, D. Hassabis, *Science* **2018**, *362*, 1140–1144.

[49] D. Garisto, *Nature* **2019**, DOI `10.1038/d41586-019-03298-6`.

[50] D. Garisto, *Nature* **2019**, DOI `10.1038/d41586-019-03298-6`.

[51] M. R. Garey, D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA, **1990**.

[52] M. Rupp, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, *Physical Review Letters* **2012**, *108*, DOI `10.1103/physrevlett.108.058301`.

[53] F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley, O. A. von Lilienfeld, *Journal of Chemical Theory and Computation* **2017**, *13*, 5255–5264.

[54] O. A. von Lilienfeld, K.-R. Müller, A. Tkatchenko, *Nature Reviews Chemistry* **2020**, *4*, 347–358.

[55] M. Rupp, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, *Physical Review Letters* **2012**, *108*, DOI `10.1103/physrevlett.108.058301`.

[56] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld, K.-R. Müller, A. Tkatchenko, *The Journal of Physical Chemistry Letters* **2015**, *6*, 2326–2331.

[57] B. Huang, O. A. von Lilienfeld, *Nature Chemistry* **2020**, *12*, 945–951.

[58] B. M. Axilrod, E. Teller, *The Journal of Chemical Physics* **1943**, *11*, 299–300.

[59] M. Y., *J. Phys. Soc. Jpn.* **1943**, *17*, 629.

[60] A. S. Christensen, L. A. Bratholm, F. A. Faber, O. A. von Lilienfeld, *The Journal of Chemical Physics* **2020**, *152*, 044107.

[61] F. A. Faber, A. S. Christensen, B. Huang, O. A. von Lilienfeld, *The Journal of Chemical Physics* **2018**, *148*, 241717.

[62] A. T. Müller, J. A. Hiss, G. Schneider, *Journal of Chemical Information and Modeling* **2018**, *58*, 472–479.

[63] S. Spänig, D. Heider, *BioData Mining* **2019**, *12*, DOI `10.1186/s13040-019-0196-x`.

[64] F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi, M. Ceriotti, *Chemical Reviews* **2021**, *121*, 9759–9815.

[65] A. S. Christensen, F. A. aber, B. Huang, L. A. Bratholm, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, QML: A Python Toolkit for Quantum Machine Learning, https://github.com/qmlcode/qml, **2017**.

[66] R. Ramakrishnan, P. Dral, M. Rupp, O. A. von Lilienfeld, **2014**, *1*, 140022.

[67] ML scripts for the constitutional isomers of QM9, `https://github.com/heini-phys-chem/const`$_{i}$*so*.

[68] K. P. Murphy, *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning)*, The MIT Press, **2012**.

[69] L. Ruddigkeit, R. van Deursen, L. Blum, J.-L. Reymond, **2012**, *52*, 2684.

[70] D. Weiniger, **1988**, *28*, 31–36.

[71] J. L. W. D. Weiniger, A. Weiniger, **1989**, *29*, 97–101.

[72] M. Schwilk, D. N. Tahchieva, O. A. von Lilienfeld.

[73] Argone Leadership Computing Facility, `https://www.alcf.anl.gov/`, Accessed: 05.06.2019.

[74] Swiss National Supercomputing Center, Annual Report 2017, `https://www.cscs.ch.`, Accessed: 26.04.2019.

[75] National Energy Research Scientigic Computing Center, `https://www.nersc.gov/.`, Accessed: 02.06.2019.

[76] Archer, `http://www.archer.ac.uk/`, Accessed: 05.06.2019.

[77] NVIDIA enabling new path to exascale supercomputing, `https://nvidianews.nvidia.com`, Accessed: 24.06.2019.

[78] C. D. Sherrill, Computational Scaling of the Configuration Interaction Method with System Size, Accessed: 03/06/19, **1996**.

[79] K. Singh, E. Ipek, S. A. McKee, B. R. de Supinski, M. Schulz, R. Caruana, *Concurrency and Computation: Practice and Experience* **2007**, *19*, 2219–2235.

[80] P. Malakar, P. Balaprakash, V. Vishwanath, V. Morozov, K. Kumaran in 2018 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), IEEE, **2018**, pp. 33–44.

[81] Y. Wang, J. Qiao, S. Lin, T. Zhao, *Comput. Sci. Eng.* **2018**, *20*, 63–76.

[82] E. R. Rodrigues, R. L. Cunha, M. A. Netto, M. Spriggs in 2016 Third International Workshop on HPC User Support Tools (HUST), IEEE, **2016**, pp. 6–13.

[83] S. K. Garg, C. S. Yeo, A. Anandasivam, R. Buyya, *J. Parallel Distrib. Comput.* **2011**, *71*, 732–749.

[84] D. Nemirovsky, T. Arkose, N. Markovic, M. Nemirovsky, O. Unsal, A. Cristal, M. Valero in Latin American High Performance Computing Conference, Springer, **2017**, pp. 3–20.

[85] G Kousalya, P Balakrishnan, C. P. Raj in *Automated Workflow Scheduling in Self-Adaptive Clouds*, Springer, **2017**, pp. 119–135.

[86] J. Sahni, D. P. Vidyarthi, *IEEE Transactions on Cloud Computing* **2018**, *6*, 2–18.

[87] H. Liu, R. Zhao, K. Nie, *IEEE Access* **2018**, *6*, 47112–47124.

[88] J. Antony, A. P. Rendell, R. Yang, G. Trucks, M. J. Frisch, *Procedia Computer Science* **2011**, *4*, 281–291.

[89] J. Papay, T. J. Atherton, M. J. Zemerly, G. R. Nudd, *Parallel Algorithms and Applications* **1996**, *10*, 127–143.

[90] S. M. Mniszewski, C. Junghans, A. F. Voter, D. Perez, S. J. Eidenbenz, *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **2015**, *25*, 15.

[91] C. Duan, J. P. Janet, F. Liu, A. Nandy, H. J. Kulik, **2019**, DOI `10.1021/acs.jctc.9b00057`.

[92] O. A. von Lilienfeld, *International Journal of Quantum Chemistry* **2013**, *113*, 1676–1689.

[93] O. A. von Lilienfeld, *Angewandte Chemie International Edition* **2018**, *57*, 4164–4169.

[94] M. Rupp, O. A. von Lilienfeld, K. Burke, *jcp* **2018**, *148*, 241401.

[95] M. Rupp, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, *prl* **2012**, *108*, 058301.

[96] K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. von Lilienfeld, A. Tkatchenko, K.-R. Müller, *jctc* **2013**, *9*, 3404–3419.

[97] R. Ramakrishnan, O. A. von Lilienfeld, *CHIMIA* **2015**, *69*, 182.

[98] B. Huang, O. A. von Lilienfeld, *jcp* **2016**, *145*, DOI `http://dx.doi.org/10.1063/1.4964627`.

[99] R. Ramakrishnan, O. A. von Lilienfeld in *Reviews in Computational Chemistry, Vol. 30*, John Wiley & Sons, Inc., **2017**, pp. 225–256.

[100] C. E. Rasmussen, C. K. I. Williams, *Gaussian Processes for Machine Learning*, see http://www.gaussianprocess.org, The MIT Press, **2006**.

[101] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, *New Journal of Physics* **2013**, *15*, 095003.

[102] J. S. Smith, O. Isayev, A. E. Roitberg, *Chem. Sci.* **2017**, *8*, 3192–3203.

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

[104] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, K.-R. Müller, *The Journal of Chemical Physics* **2018**, *148*, 241722.

[105] O. T. Unke, M. Meuwly, *The Journal of Chemical Physics* **2018**, *148*, 241708.

[106] G. N. Simm, M. Reiher, **2018**, *14*, 5238–5248.

[107] J. S. Smith, O. Isayev, A. E. Roitberg, *Scientific data* **2017**, *4*, 170193.

[108] J. P. Janet, H. J. Kulik, *Chem. Sci.* **2017**, *8*, 5137–5152.

[109] Z. Li, N. Omidvar, W. S. Chin, E. Robb, A. Morris, L. Achenie, H. Xin, *J. Phys. Chem. A* **2018**, *122*, PMID: 29688014, 4571–4578.

[110] H. H. Rosenbrock, *The Computer Journal* **1960**, *3*, 175–184.

[111] D. M. Himmelblau, *Applied Nonlinear Programming*, ISBN-13: 978-0070289215, McGraw-Hill, **1972**.

[112] E. Jones, T. Oliphant, P. Peterson, SciPy: Open source scientific tools for Python, [Version: 1.3.1], **2001–**.

[113] J. A. Nelder, R. Mead, *Comput. J.* **1965**, *7*, 308–313.

[114] R. H. Byrd, P. Lu, J. Nocedal, C. Zhu, *SIAM Journal on Scientific Computing* **1995**, *16*, 1190–1208.

[115] S. G. Nash, *SIAM Journal on Numerical Analysis* **1984**, *21*, 770–788.

[116] M. Schwilk, Q. Ma, C. Köppl, H.-J. Werner, *J. Chem. Theory Comput.* **2017**, *13*, 3650–3675.

[117] Q. Ma, M. Schwilk, C. Köppl, H.-J. Werner, *J. Chem. Theory Comput.* **2017**, *13*, 4871–4896.

[118] Q. Ma, H.-J. Werner, *J. Chem. Theory Comput.* **2018**, *14*, 198–215.

[119] M. Schwilk, P. Zaspel, O. A. von Lilienfeld, H. Harbrecht, *to be published* **2019**.

[120] P. J. Knowles, H.-J. Werner, *Chem. Phys. Lett.* **1988**, *145*, 514 –522.

[121] H. Werner, P. J. Knowles, *J. Chem. Phys.* **1988**, *89*, 5803–5814.

[122] T. Shiozaki, G. Knizia, H.-J. Werner, *J. Chem. Phys.* **2011**, *134*, 034113.

[123] T. Shiozaki, H.-J. Werner, *Mol. Phys.* **2013**, *111*, 607–630.

[124] D. N. Tahchieva, M. Schwilk, O. A. von Lilienfeld, *to be published* **2019**.

[125] A. D. Becke, *J. Chem. Phys* **1993**, *98*, 5648.

[126] C. Lee, W. Yang, R. G. Parr, *Phys. Rev. B* **1988**, *37*, 785–789.

[127] H. Werner, P. J. Knowles, *J. Chem. Phys.* **1985**, *82*, 5053–5063.

[128] T. Busch, A. D. Esposti, H. Werner, *J. Chem. Phys.* **1991**, *94*, 6708–6715.

[129] H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, M. Schütz, and others, MOLPRO, a package of ab initio programs, see http://www.molpro.net, Stuttgart, Germany.

[130] F. Neese, **2006**, ORCA 2.8, *An ab initio, density functional and semiempirical program package*, University of Bonn, Germany.

[131] *Perf* of the linux kernel version 3.10.0-327.el7.x86_64 tools was used. *Perf* measures the number of retired FLOP (as a certain amount of speculative executions may be negated, given that logical branches cannot be evaluated between instructions within a clock cycle).

[132] Q. Ma, H.-J. Werner, *WIRES COMPUT MOL SCI*, *8*, e1371.

[133] D. G. Krige, *Journal of the Chemical Metallurgical and Mining Society of South Africa* **1951**, *52*, 119–139.

[134] O. A. von Lilienfeld, R. Ramakrishnan, M. Rupp, A. Knoll, **2015**, *115*, http://arxiv.org/abs/1307.2918, 1084.

[135] K. R. Müller, M. Finke, N. Murata, K. Schulten, S. Amari, **1996**, *8*, 1085.

[136] B. Huang, O. A. von Lilienfeld, *arXiv preprint arXiv:1707.04146* **2017**, submitted to Nature.

[137] R. Ramakrishnan, P. Dral, M. Rupp, O. A. von Lilienfeld, *jctc* **2015**, *11*, 2087.

[138] Intel Corporation, Intel Math Kernel Library, Accessed: 11/20/2018, **2018**.

[139] Z. Xianyi, W. Qian, W. Saar, OpenBLAS, An optimized BLAS library, Accessed: 11/16/18, **2017**.

[140] Open MPI: Open Source High Performance Computing, Accessed: 11/15/18, **2018**.

[141] Global Arrays Programming Models, Accessed: 11/16/18, **2018**.

[142] J. Nieplocha, B. Palmer, V. Tipparaju, M. Krishnan, H. Trease, E. Apra, *Int. J. High Perf. Comp. Appl.* **2006**, *20*, 203–231.

[143] Due to non-deterministic run time behaviour of the CPU, as well as measurement errors of *perf*, the FLOP count varies within a few tenth of percent for consecutive runs of the same calculation.

[144] Raw data for this work and sample implementations for easy use can be found on `https://github.com/ferchault/mlscheduling`.

[145] W. A. Warr, *Molecular Informatics* **2014**, *33*, 469–476.

[146] N. Schneider, D. M. Lowe, R. A. Sayle, G. A. Landrum, *Journal of Chemical Information and Modeling* **2015**, *55*, 39–53.

[147] J. L. Baylon, N. A. Cilfone, J. R. Gulcher, T. W. Chittenden, *Journal of Chemical Information and Modeling* **2019**, *59*, 673–688.

[148] H. Gao, T. J. Struble, C. W. Coley, Y. Wang, W. H. Green, K. F. Jensen, *ACS Central Science* **2018**, *4*, 1465–1476.

[149] G. Henkelman, G. Jóhannesson, H. Jónsson in *Theoretical methods in condensed phase chemistry*, Springer, **2002**, pp. 269–302.

[150] G. Henkelman, B. P. Uberuaga, H. Jónsson, *The Journal of chemical physics* **2000**, *113*, 9901–9904.

[151] G. Henkelman, H. Jónsson, *The Journal of chemical physics* **2000**, *113*, 9978–9985.

[152] J. Zheng, Y. Zhao, D. G. Truhlar, *Journal of Chemical Theory and Computation* **2009**, *5*, 808–821.

[153] A. P. Bento, F. M. Bickelhaupt, *The Journal of Organic Chemistry* **2008**, *73*, 7290–7299.

[154] R. Yi, H. Basch, S. Hoz, *The Journal of Organic Chemistry* **2002**, *67*, 5891–5895.

[155] S. Liu, H. Hu, L. G. Pedersen, *The Journal of Physical Chemistry A* **2010**, *114*, 5913–5918.

[156] F. M. Bickelhaupt, *Journal of computational chemistry* **1999**, *20*, 114–128.

[157] X.-P. Wu, X.-M. Sun, X.-G. Wei, Y. Ren, N.-B. Wong, W.-K. Li, *Journal of Chemical Theory and Computation* **2009**, *5*, PMID: 26609852, 1597–1606.

[158] Y. Zhao, D. G. Truhlar, *Journal of Chemical Theory and Computation* **2010**, *6*, 1104–1108.

[159] S. M. Villano, N. Eyet, W. C. Lineberger, V. M. Bierbaum, *Journal of the American Chemical Society* **2009**, *131*, PMID: 19456156, 8227–8233.

[160] B Safi, K Choho, P Geerlings, *The Journal of Physical Chemistry A* **2001**, *105*, 591–601.

[161] S. Gronert, A. E. Fagin, K. Okamoto, S. Mogali, L. M. Pratt, *Journal of the American Chemical Society* **2004**, *126*, 12977–12983.

[162] S. Grimme, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2011**, *1*, 211–228.

[163] T. Schwabe, S. Grimme, *Accounts of chemical research* **2008**, *41*, 569–579.

[164] A. Laio, M. Parrinello, *Proceedings of the National Academy of Sciences* **2002**, *99*, 12562–12566.

[165] S. Gronert, L. M. Pratt, S. Mogali, *Journal of the American Chemical Society* **2001**, *123*, 3081–3091.

[166] S. M. Villano, S. Kato, V. M. Bierbaum, *Journal of the American Chemical Society* **2006**, *128*, 736–737.

[167] P. D. Mezei, O. A. von Lilienfeld, *Journal of Chemical Theory and Computation* **2020**, *16*, 2647–2653.

[168] A. S. Christensen, O. A. von Lilienfeld, *CHIMIA International Journal for Chemistry* **2019**, *73*, 1028–1031.

[169] A. S. Christensen, L. A. Bratholm, F. A. Faber, O. A. von Lilienfeld, *The Journal of Chemical Physics* **2020**, *152*, 044107.

[170] C. W. Coley, R. Barzilay, T. S. Jaakkola, W. H. Green, K. F. Jensen, *ACS Central Science* **2017**, *3*, 434–443.

[171] B. Sanchez-Lengeling, A. Aspuru-Guzik, *Science* **2018**, *361*, 360–365.

[172] M. H. S. Segler, M. P. Waller, *Chemistry - A European Journal* **2017**, *23*, 5966–5971.

[173] A. Fabrizio, B. Meyer, R. Fabregat, C. Corminboeuf, *CHIMIA International Journal for Chemistry* **2019**, *73*, 983–989.

[174] C. W. Coley, W. Jin, L. Rogers, T. F. Jamison, T. S. Jaakkola, W. H. Green, R. Barzilay, K. F. Jensen, *Chemical science* **2019**, *10*, 370–377.

[175] J. A. Kammeraad, J. Goetz, E. A. Walker, A. Tewari, P. M. Zimmerman, *Journal of Chemical Information and Modeling* **2020**, *60*, PMID: 32091880, 1290–1301.

[176] P. Sadowski, D. Fooshee, N. Subrahmanya, P. Baldi, *Journal of Chemical Information and Modeling* **2016**, *56*, PMID: 27749058, 2125–2128.

[177] S. Brandt, F. Sittel, M. Ernst, G. Stock, *The Journal of Physical Chemistry Letters* **2018**, *9*, PMID: 29630378, 2144–2150.

[178] A. R. Singh, B. A. Rohr, J. A. Gauthier, J. K. Nørskov, *Catalysis Letters* **2019**, *149*, 2347–2354.

[179] P. Zaspel, B. Huang, H. Harbrecht, O. A. von Lilienfeld, *Journal of Chemical Theory and Computation* **2018**, *15*, 1546–1559.

[180] J. S. Smith, B. T. Nebgen, R. Zubatyuk, N. Lubbers, C. Devereux, K. Barros, S. Tretiak, O. Isayev, A. E. Roitberg, *Nature Communications* **2019**, *10*, DOI `10.1038/s41467-019-10827-4`.

[181] L. Ruddigkeit, R. van Deursen, L. C. Blum, J.-L. Reymond, *Journal of Chemical Information and Modeling* **2012**, *52*, 2864–2875.

[182] J. M. Gonzales, R. S. Cox, S. T. Brown, W. D. Allen, H. F. Schaefer, *The Journal of Physical Chemistry A* **2001**, *105*, 11327–11346.

[183] Y. Zhao, N. González-García, D. G. Truhlar, *The Journal of Physical Chemistry A* **2005**, *109*, 2012–2018.

[184] M. Stei, E. Carrascosa, M. A. Kainz, A. H. Kelkar, J. Meyer, I. Szabó, G. Czakó, R. Wester, *Nature Chemistry* **2015**, *8*, 151–156.

[185] T. A. Hamlin, M. Swart, F. M. Bickelhaupt, *ChemPhysChem* **2018**, *19*, 1315–1330.

[186] O. T. Unke, M. Meuwly, *Journal of Chemical Theory and Computation* **2019**, *15*, 3678–3693.

[187] S. Gronert, *Journal of the American Chemical Society* **1991**, *113*, 6041–6048.

[188] R. Krishnan, J. S. Binkley, R. Seeger, J. A. Pople, *The Journal of Chemical Physics* **1980**, *72*, 650–654.

[189] L. A. Curtiss, M. P. McGrath, J.-P. Blaudeau, N. E. Davis, R. C. Binning, L. Radom, *The Journal of Chemical Physics* **1995**, *103*, 6104–6113.

[190] A. D. McLean, G. S. Chandler, *The Journal of Chemical Physics* **1980**, *72*, 5639–5648.

[191] M. J. Frisch, J. A. Pople, J. S. Binkley, *The Journal of Chemical Physics* **1984**, *80*, 3265–3269.

[192] T. Clark, J. Chandrasekhar, G. W. Spitznagel, P. V. R. Schleyer, *Journal of Computational Chemistry* **1983**, *4*, 294–301.

[193] P. L. Fast, D. G. Truhlar, *The Journal of Physical Chemistry A* **2000**, *104*, 6111–6116.

[194] J. Baker, P. Pulay, *The Journal of Chemical Physics* **2002**, *117*, 1441–1449.

[195] S. Schenker, C. Schneider, S. B. Tsogoeva, T. Clark, *Journal of Chemical Theory and Computation* **2011**, *7*, 3586–3595.

[196] B. J. Lynch, D. G. Truhlar, *The Journal of Physical Chemistry A* **2001**, *105*, 2936–2941.

[197] X. Xu, I. M. Alecu, D. G. Truhlar, *Journal of Chemical Theory and Computation* **2011**, *7*, 1667–1676.

[198] N. M. OBoyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, G. R. Hutchison, *Journal of Cheminformatics* **2011**, *3*, 33.

[199] A. K. Rappe, C. J. Casewit, K. S. Colwell, W. A. Goddard, W. M. Skiff, *Journal of the American Chemical Society* **1992**, *114*, 10024–10035.

[200] S. Riniker, G. A. Landrum, *Journal of Chemical Information and Modeling* **2015**, *55*, 2562–2574.

[201] F. Neese, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2011**, *2*, 73–78.

[202] M. M. Francl, W. J. Pietro, W. J. Hehre, J. S. Binkley, M. S. Gordon, D. J. DeFrees, J. A. Pople, *The Journal of Chemical Physics* **1982**, *77*, 3654–3665.

[203] E. F. Valeev, Libint: A library for the evaluation of molecular integrals of many-body operators over Gaussian functions, http://libint.valeyev.net/, version 2.7.0-beta.5, **2020**.

[204] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, O. Beckstein, *Journal of Computational Chemistry* **2011**, *32*, 2319–2327.

[205] Gaussian09 quantum chemistry code, `http://gaussian.com/`, Accessed: 08.09.2021.

[206] P. J. Stephens, F. J. Devlin, C. F. Chabalowski, M. J. Frisch, *The Journal of Physical Chemistry* **1994**, *98*, 11623–11627.

[207] R. Ditchfield, W. J. Hehre, J. A. Pople, *The Journal of Chemical Physics* **1971**, *54*, 724–728.

[208] W. J. Hehre, R. Ditchfield, J. A. Pople, *The Journal of Chemical Physics* **1972**, *56*, 2257–2261.

[209] P. C. Hariharan, J. A. Pople, *Theoretica Chimica Acta* **1973**, *28*, 213–222.

[210] K. Fukui, *Accounts of Chemical Research* **1981**, *14*, 363–368.

[211] H.-J. Werner, M. Schütz, *The Journal of Chemical Physics* **2011**, *135*, 144116.

[212] C. Hampel, K. A. Peterson, H.-J. Werner, *Chemical Physics Letters* **1992**, *190*, 1–12.

[213] M. Schütz, F. R. Manby, *Phys. Chem. Chem. Phys.* **2003**, *5*, 3349–3358.

[214] T. H. Dunning, *The Journal of Chemical Physics* **1989**, *90*, 1007–1023.

[215] R. A. Kendall, T. H. Dunning, R. J. Harrison, *The Journal of Chemical Physics* **1992**, *96*, 6796–6806.

[216] A. K. Wilson, D. E. Woon, K. A. Peterson, T. H. Dunning, *The Journal of Chemical Physics* **1999**, *110*, 7667–7676.

[217] D. E. Woon, T. H. Dunning, *The Journal of Chemical Physics* **1993**, *98*, 1358–1371.

[218] G. Vayner, K. N. Houk, W. L. Jorgensen, J. I. Brauman, *Journal of the American Chemical Society* **2004**, *126*, 9054–9058.

[219] J. M. Granda, L. Donina, V. Dragone, D.-L. Long, L. Cronin, *Nature* **2018**, *559*, 377–381.

[220] M. M. Flores-Leonar, L. M. Mejía-Mendoza, A. Aguilar-Granda, B. Sanchez-Lengeling, H. Tribukait, C. Amador-Bedolla, A. Aspuru-Guzik, *Current Opinion in Green and Sustainable Chemistry* **2020**, *25*, 100370.

[221] T. A. Young, J. J. Silcock, A. J. Sterling, F. Duarte, *Angew. Chem. Int. Ed.* **2020**, DOI `10.1002/anie.202011941`.

[222] A. L. Dewyer, A. J. Argüelles, P. M. Zimmerman, *WIREs Comput Mol Sci* **2017**, *8*, e1354.

[223] C. W. Coley, N. S. Eyke, K. F. Jensen, *Angew. Chem. Int. Ed.* **2020**, DOI `10.1002/anie.201909987`.

[224] P. M. Zimmerman, *J. Comput. Chem.* **2015**, *36*, 601–611.

[225] G. Henkelman, B. P. Uberuaga, H. Jónsson, *The Journal of Chemical Physics* **2000**, *113*, 9901–9904.

[226] P. M. Zimmerman, *The Journal of Chemical Physics* **2013**, *138*, 184102.

[227] S. Szymkuć, E. P. Gajewska, T. Klucznik, K. Molga, P. Dittwald, M. Startek, M. Bajczyk, B. A. Grzybowski, *Angew. Chem. Int. Ed.* **2016**, *55*, 5904–5937.

[228] P. M. Pflüger, F. Glorius, *Angew. Chem. Int. Ed.* **2020**, DOI `10.1002/anie.202008366`.

[229] F. Sandfort, F. Strieth-Kalthoff, M. Kühnemund, C. Beecks, F. Glorius, *Chem* **2020**, *6*, 1379–1390.

[230] B. Huang, O. A. von Lilienfeld, Ab initio machine learning in chemical compound space, **2020**.

[231] P. Friederich, G. dos Passos Gomes, R. D. Bin, A. Aspuru-Guzik, D. Balcells, *Chemical Science* **2020**, *11*, 4584–4601.

[232] K. Jorner, T. Brinck, P.-O. Norrby, D. Buttar, **2020**, DOI 10.26434/chemrxiv.12758498.v1.

[233] E. Komp, S. Valleau, *The Journal of Physical Chemistry A* **2020**, *124*, 8607–8613.

[234] S. Gallarati, R. Fabregat, R. Laplaza, S. Bhattacharjee, M. D. Wodrich, C. Corminboeuf, *Chemical Science* **2021**, *12*, 6879–6889.

[235] N. E. S. K. P. C. Vollhardt, *Organische Chemie*, John Wiley & Sons, 2011.

[236] R. Ramakrishnan, O. A. von Lilienfeld, *CHIMIA International Journal for Chemistry* **2015**, *69*, 182–186.

[237] *Proceedings of the Royal Society of London. Series A Containing Papers of a Mathematical and Physical Character* **1924**, *106*, 463–477.

[238] A. T. Müller, J. A. Hiss, G. Schneider, *Journal of Chemical Information and Modeling* **2018**, *58*, 472–479.

[239] S. Spänig, D. Heider, *BioData Mining* **2019**, *12*, DOI 10.1186/s13040-019-0196-x.

[240] S. Heinen, G. F. von Rudorff, A. von Lilienfeld, version v1 **2021**, DOI 10.5281/zenodo.4925938.

[241] O. A. von Lilienfeld, R. Ramakrishnan, M. Rupp, A. Knoll, *International Journal of Quantum Chemistry* **2015**, *115*, 1084–1093.

[242] B. Parsaeifard, D. S. De, A. S. Christensen, F. A. Faber, E. Kocer, S. De, J. Behler, A. von Lilienfeld, S. Goedecker, *Machine Learning: Science and Technology* **2020**, DOI 10.1088/2632-2153/abb212.

[243] A. K. Rappe, C. J. Casewit, K. S. Colwell, W. A. Goddard, W. M. Skiff, *Journal of the American Chemical Society* **1992**, *114*, 10024–10035.

[244] M. R. Marcel F. Langer, Alex Goeßmann, **2021**.

[245] F. A. Faber, A. Lindmaa, O. A. von Lilienfeld, R. Armiento, *Physical Review Letters* **2016**, *117*, DOI 10.1103/physrevlett.117.135502.

[246] J. Schmidt, J. Shi, P. Borlido, L. Chen, S. Botti, M. A. L. Marques, *Chemistry of Materials* **2017**, *29*, 5090–5103.

[247] M. Bragato, G. F. von Rudorff, O. A. von Lilienfeld, *Chemical Science* **2020**, *11*, 11859–11868.

[248] D. Lemm, G. F. von Rudorff, O. A. von Lilienfeld, *Nature Communications* **2021**, *12*, DOI 10.1038/s41467-021-24525-7.

[249] J. Peiró-García, I. Nebot-Gil, *ChemPhysChem* **2003**, *4*, 843–847.

[250] R. Q. Zhang, W. C. Lu, Y. L. Zhao, S. T. Lee, *The Journal of Physical Chemistry B* **2004**, *108*, 1967–1973.

[251] A. Shiroudi, M. S. Deleuze, S. Canneaux, *Physical Chemistry Chemical Physics* **2015**, *17*, 13719–13732.

[252] J. Weinreich, N. J. Browning, O. A. von Lilienfeld, *The Journal of Chemical Physics* **2021**, *154*, 134113.

[253] D. M. Lowe, **2012**, DOI 10.17863/CAM.16293.

[254] R. Jackson, W. Zhang, J. Pearson, *Chemical Science* **2021**, *12*, 10022–10040.

[255] J. A. G. Torres, P. C. Jennings, M. H. Hansen, J. R. Boes, T. Bligaard, *Physical Review Letters* **2019**, *122*, DOI 10.1103/physrevlett.122.156001.

[256] H. L. Mortensen, S. A. Meldgaard, M. K. Bisbo, M.-P. V. Christiansen, B. Hammer, Atomistic Structure Learning Algorithm with surrogate energy model relaxation, **2020**.

[257] D. Lemm, G. F. von Rudorff, O. A. von Lilienfeld, *Nature Communications* **2021**, *12*, DOI `10.1038/s41467-021-24525-7`.

[258] R. Meyer, A. W. Hauser, *The Journal of Chemical Physics* **2020**, *152*, 084112.

[259] D. Born, J. Kästner, *Journal of Chemical Theory and Computation* **2021**, DOI `10.1021/acs.jctc.1c00517`.

[260] M. Z. Makoś, N. Verma, E. C. Larson, M. Freindorf, E. Kraka, *The Journal of Chemical Physics* **2021**, *155*, 024116.

[261] C. Peng, H. B. Schlegel, *Israel Journal of Chemistry* **1993**, *33*, 449–454.

[262] P. J. Stevens, F. J. Devlin, C. F. Chabalowski, M. J. Frisch, **1993**, *98*, 11623.

[263] C. Adamo, V. Barone, *The Journal of Chemical Physics* **1999**, *110*, 6158–6170.

[264] N. Mardirossian, M. Head-Gordon, *Physical Chemistry Chemical Physics* **2014**, *16*, 9904.

[265] J. S. Binkley, J. A. Pople, W. J. Hehre, *J. Am. Chem. Soc.* **1980**, *102*, 939–947.

[266] G. A. Petersson, A. Bennett, T. G. Tensfeldt, M. A. Al-Laham, W. A. Shirley, J. Mantzaris, *The Journal of Chemical Physics* **1988**, *89*, 2193–2218.

[267] G. A. Petersson, M. A. Al-Laham, *The Journal of Chemical Physics* **1991**, *94*, 6081–6090.

[268] Python library to calculate the RMSD, `https://github.com/charnley/rmsd`.

[269] W. Kabsch, *Acta Crystallographica Section A* **1976**, *32*, 922–923.

[270] S. Heinen, G. F. von Rudorff, O. A. von Lilienfeld, **2022**, DOI `10.5281/zenodo.6823150`.

[271] B. Huang, O. A. von Lilienfeld, *Nature Chemistry* **2020**.

[272] P. Zaspel, B. Huang, H. Harbrecht, O. A. von Lilienfeld, *Journal of Chemical Theory and Computation* **2018**, *15*, 1546–1559.

[273] J. S. Smith, B. T. Nebgen, R. Zubatyuk, N. Lubbers, C. Devereux, K. Barros, S. Tretiak, O. Isayev, A. Roitberg, **2018**, DOI `10.26434/chemrxiv.6744440.v1`.

[274] J. S. Smith, B. T. Nebgen, R. Zubatyuk, N. Lubbers, C. Devereux, K. Barros, S. Tretiak, O. Isayev, A. E. Roitberg, *Nature Communications* **2019**, *10*, DOI `10.1038/s41467-019-10827-4`.

[275] D. C. Liu, J. Nocedal, *Mathematical Programming* **1989**, *45*, 503–528.

[276] L. Lu, *Int. J. Quantum Chem.* **2015**, *115*, 502–509.

[277] M. Ernzerhof, G. E. Scuseria, **1999**, *110*, 5029.

[278] J. Hoja, L. M. Sandonas, B. G. Ernst, A. Vazquez-Mayagoitia, R. A. DiStasio, A. Tkatchenko, *Scientific Data* **2021**, *8*, DOI `10.1038/s41597-021-00812-2`.

[279] M. A. Murphy, *Foundations of Chemistry* **2017**, *20*, 121–165.

[280] S. Martello, P. Toth, *Discrete Applied Mathematics* **1990**, *28*, 59–70.

[281] S. Martello, P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley & Sons, Inc., New York, NY, USA, **1990**.

[282] R. E. Korf in Eighteenth National Conference on Artificial Intelligence, American Association for Artificial Intelligence, Edmonton, Alberta, Canada, **2002**, pp. 731–736.

[283] E. G. Coffman, M. R. Garey, D. S. Johnson in *Algorithm Design for Computer System Design*, (Eds.: G. Ausiello, M. Lucertini, P. Serafini), Springer Vienna, Vienna, **1984**, pp. 49–106.

[284] M. Yue, *Acta Mathematicae Applicatae Sinica* **1991**, *7*, 321–331.

[285] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K.

Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, K. W. Jacobsen, *Journal of Physics: Condensed Matter* **2017**, *29*, 273002.

[286] F. Neese, *WIREs Computational Molecular Science* **2022**, DOI `10.1002/wcms.1606`.