Machine Learning Workflows for Chemistry: Applications in Catalysis and Ionic Liquids

Stamatia Zavitsanou
Oriel College
University of Oxford

July 2023

Author's Declaration

The work presented in this thesis was conducted under the supervision of Professor Fernanda Duarte at the Department of Chemistry, University of Oxford. I declare that all the work is my own, unless otherwise stated, and has not been submitted for any other degree at this or any other university.

Stamatia Zavitsanou

July 2023

Abstract

In today's world, data is being generated and accumulated at an astronomical rate, presenting new opportunities and challenges for the scientific community. In parallel, advancements in computer science have revolutionized the landscape of chemistry. The confluence of these two fields has given rise to a wave of sophisticated machine learning algorithms capable of building powerful predictive models. The field of computational chemistry is now finding itself navigating through this rapid evolution of technological progress.

This Thesis traces the progression from statistical models to modern machine learning techniques and is setting the stage for the intricate dance between data science and chemistry. The focus narrows down to the specific utilization of machine learning for selectivity predictions in organocatalysis and property predictions for ionic liquids. It introduces *Pythia*, a machine learning toolkit designed with accessibility in mind, aiming to democratize the application of machine learning in computational chemistry. *Pythia* employs 2D and 3D descriptors and shallow learners to predict selectivity for organocatalytic reactions. The power of *Pythia* is put to the test and its potential for predicting selectivity in catalysis is explored. This demonstrates the toolkit's practical utility in facilitating more efficient and targeted experimentation in the search for effective catalysts. Finally, we delve into the prediction of viscosity and solubility in ionic liquids, further highlighting the capabilities of machine learning in streamlining the prediction of chemical properties. This Thesis promises to accelerate the pace of discovery in computational chemistry, allowing scientists to handle the influx of data more efficiently and extract meaningful insights from it.

Acknowledgments

I would like to express my gratitude to Fernanda Duarte, for her support, supervision, and guidance throughout these past years. I am truly grateful to her for believing in me and granting me the opportunity to pursue my studies at the University of Oxford. I am sincerely thankful to the Department of Chemistry for funding my DPhil studies, and I would like to acknowledge the generous support from the Holly Synod of Greece, which funded the final year of my DPhil. Their financial assistance has been crucial in enabling me to pursue my research endeavors.

I would like to extend my special thanks to Tom Watts and Emanuele Casali for the close collaboration on this Thesis. Their dedicated partnership and contributions have been instrumental in the development of this research. Additionally, I express my deep appreciation to Alistair Sterling (loukoumaki), Tom Young (zouzounaki), and Tanya, for being with me since the beginning. Thank you to Ally, Bernie, Tomasz, Hafiz, Veronika, Henry, Aleksy, Tristan, Hanwen, Chloe, Ewa and all other members of the group, their invaluable advice and guidance have been indispensable to the progress of this work. I am also grateful to Dr. James McDonagh and Dr. Flaviu Cipcigian for their mentoring and support. Furthermore, I would like to acknowledge all my previous supervisors and teachers, particularly Dr. Zoe Cournia and Dr. Stavros Perantonis, for their significant role in motivating me to pursue this PhD.

Lastly, I would like to express my deepest appreciation to my parents, Joanna and Nick, and my siblings, Rania and Iandros, as well as my theies and theios, Andreas, Diamado, Toula, Nick, Tasia and my cousins Maria, Vaso, Dimitris, George, and especially Irene, Dina, Chris and Louis, who tutored me during my time at school enabling me to pursue and successfully gain admission to university. All my cousins (all 33 of them – even the ones not mentioned here) have been my pillars of support and have played an integral role in shaping my path to where I stand today. I am profoundly grateful for their love, guidance, and unwavering belief in me.

I also extend my gratitude to my dear friends, Konstantinos, Myrto, Andriana, Sofia, Achilleas, Alexis, Cleopatra, George, Mtzw Nav, Eleni, Haris, Stephanie, Serafeim, and last but not least the “Frenchies” (Max, Stuart, Dani, Remi, Lauriane, Dimitris, Mimi, Paul, Sarah, Alexi, Mariana, Cenk, Sylvain, Paul and Joanna). Their constant support and encouragement have been invaluable throughout my journey, and I owe much of my success to their presence in my

life. The biggest thank you goes to Jeremy, for loving me, taking care of me, and supporting me unconditionally over the past years.

I would like to dedicate this Thesis to my theio Andrea, who would anxiously call me every so often to inquire about my progress and ask how ‘Fernandes’ was doing (referring to Fernanda), and to Manthos, who, although may not be here to celebrate this achievement with me, I know would dance for days were he still with us.

Data Availability

The data presented in this Thesis, including Cartesian coordinates, results, and Python scripts used for data processing, has been submitted as supplementary material. The contents of the supplementary material are listed in Appendix A. This data has been submitted, in a digital format, along with the Thesis.

Publications

This Thesis is based upon the following publication:

1. Sterling, A.J.†, Zavitsanou, S.†, Ford, J. and Duarte, F.*, Selectivity in Organocatalysis – From Qualitative to Quantitative Predictive Models, *WIREs Comput. Mol. Sci.* 2021, *11*, e1518.

The following manuscripts were in preparation at the time of completion of this degree:

1. Zavitsanou, S., McDonagh, J.L., Cipcigan, F. and Duarte, F.*, Predicting Viscosity and Solubility in Ionic Liquids with Graph Neural Networks.
2. Zavitsanou, S., Bo, Z., and Duarte, F.*, *Pythia*: Explainable Machine Learning in Chemistry for Non-Experts.
3. Zavitsanou, S., Casali, E., and Duarte, F.*, Predicting enantioselectivity for the Strecker synthesis of α-amino acids.

Other publications:

1. McDonagh, J.L.*, Wunsch, B.H.*, Zavitsanou, S., Harrison, A., Elmegreen, B., Gifford, S., Van Kessel, T., Cipcigan, F., Machine Guided Discovery of Novel Carbon Capture Solvents, arXiv:2303.14223, 2023.
2. McDonagh, J.L.*, Zavitsanou, S., Harrison, A., Zubarev, D., Wunsch, B.H., Van Kessel, T., and Cipcigan, F., Chemical Space Analysis and Property Prediction for Carbon Capture Amine Molecules, ChemRxiv, 2022.
3. Zavitsanou, S., Tsengenes, A., Papadourakis, M., Amendola, G., Chatzigoulas, A., Dellis, D., Cosconati, S., and Cournia, Z.*, FEPrepare: A Web-Based Tool for Automating the Setup of Relative Binding Free Energy Calculations, *J. Chem. Inf. Model.* 2021, 61, 9, 4131-4138.
4. Rogova, T., Gabriel, P., Zavitsanou, S., Leitch, J.A., Duarte, F.* and Dixon, D.J.*, Reverse Polarity Reductive Functionalization of Tertiary Amides via a Dual Iridium-Catalyzed Hydrosilylation and Single Electron Transfer Strategy, *ACS Catal.* 2020, 10, 19, 11438–11447.

List of Abbreviations

AARD	Average Absolute Relative Deviation
AARE	Average Absolute Relative Error
AO	Atomic Orbital
AUC	Area Under the Curve
BO	Bond Order
C	Celsius
CNN	Convolutional Neural Networks
COS	Chain of Spheres
COSMO-RS	COnductor-like Screening MOdel for Realistic Solvation
DCM	DiChloroMethane
DFB	DiFluoroBenzene
DFT	Density Functional Theory
DFTB	Density-Functional Tight-Binding
DT	Decision Tree
ECFP	Extended-Connectivity FingerPrints
ee	Enantiomeric Excess
er	Enantiomeric Ratio
ET	Extra Trees
FN	False Negatives
FP	False Positives
GC	Group Contribution
GCN	Graph Convolutional Networks
GGA	Generalized Gradient Approximation
GNN	Graph Neural Network
GP	Gaussian Process
GRU	Gated Recurrent Unit
GTO	Gaussian-Type Orbital
HB	Hydrogen Bonding
HF	Hartree-Fock
HOMO	Highest Occupied Molecular Orbital
IL	Ionic Liquids
K	Kelvin
KDE	Kernel Density Estimation
kNN	k-Nearest Neighbor
LASSO	Least Absolute Shrinkage and Selection Operator
LASSOCV	Least Absolute Shrinkage and Selection Operator Cross-Validation
LDA	Linear Discriminant Analysis
LFER	Linear Free Energy Relationships
LR	Logistic Regression
LUMO	Lowest Occupied Molecular Orbital
MACCS	Molecular ACCess System
MAE	Mean Absolute Error
MCC	Matthews Correlation Coefficient

MDEA	Dimethylethanolamine
MEA	Ethanolamine
ML	Machine Learning
MLP	Multi-Layer Perceptron
MLR	Multivariable Linear Regression
MO	Molecular Orbital
MPNN	Message Passing Neural Networks
MSE	Mean Squared Error
MTBE	Methyl Tert-Butyl Ether
NBO	Natural Bond Orbital
NMR	Nuclear Magnetic Resonance
NN	Neural Networks
OPERA	OPEn (q)saR App
PC	Principal Components
PCA	Principal Component Analysis
PET	Positron Emission Tomography
PTC	Phase Transfer Catalysis
QM	Quantum Mechanics
QSAR	Quantitative Structure-Activity Relationships
QSPR	Quantitative Structure-Property Relationships
RF	Random Forests
RI	Resolution of Identity
RI-JCOSX	Resolution of Identity Chain of spheres Exchange
RMSD	Root-Mean-Square Deviation
RMSE	Root-Mean-Square Error
RNN	Recurrent Neural Networks
ROC	Receiver Operating Characteristic (curve)
RSS	Residual Sum of Squares
SELFIES	SELF-referencIng Embedded Strings
SGD	Stochastic Gradient Descent
SHAP	SHapley Additive exPlanations
SMD	Solvation Model Density
SMILES	Simplified Molecular-Input Line-Entry System
SMOTE	Synthetic Minority Over-sampling TEchnique
SMOTEN	Synthetic Minority Over-sampling TEchnique for Nominal
SMOTENC	Synthetic Minority Over-sampling TEchnique for Nominal and Continuous
SOAP	Smooth Overlap of Atomic Positions
STO	Slater-Type Orbital
SV	Support Vector
SVM	Support Vector Machine
SVR	Support Vector Regression
TB	Tight Binding
TN	True Negatives
TP	True Positives
TPSA	Topological Polar Surface Area
TS	Transition State

t-SNE	t-distributed Stochastic Neighbor Embedding
UNIFAC	UNIversal Functional Activity Coefficient
UNIQUAC	UNiversal QUAsiChemical
vdW	Van der Waals
VFT	Vogel-Fulcher-Tammann
VIF	Variance Inflation Factor
WFT	Wave Function Theory

Table of Contents

1. Introduction

Parts of this chapter are based on the published review "*Selectivity in organocatalysis-From qualitative to quantitative predictive models*" at the WIREs computational Molecular Science.[1]

1.1. A brief history of machine learning

In recent years the use of Machine learning (ML) has witnessed explosive growth in all areas of science. This surge can be attributed to the abundance of available data and the advancements in computing resources, enabling researchers to delve into complex analyses. ML leverages and expands upon the foundations of statistics to develop and enhance its algorithms and offer insights into the underlying mechanisms while generating accurate predictions.[2–6]

The history of statistics can be traced back to the 17th century and dice gambling. During this time, Fermat and Pascal began developing the concept of probability, which continued during the 18th century with Bayes, who made significant contributions to the field through his work on conditional probability, which provided the basis of Bayesian inference. In the 19th century, Laplace further expanded upon Bayes' ideas and introduced additional concepts to Bayesian statistics. In this period, Laplace, Gauss, and Legendre also introduced the method of least squares and formulated an early version of the standard linear model. Additionally, Quetelet organized the first international statistics conference, where the application of statistics to biology was discussed.[7–11]

The foundation of statistical theory was laid by Galton and Pearson, who are credited as its principal founders. They introduced concepts such as standard deviation, median, correlation, regression, as well as their practical applications.[12–15] Pearson also made significant contributions to statistical hypothesis testing, introducing Pearson's chi-squared test and principal component analysis (PCA).[16–19] In 1922, Fisher introduced the method of maximum likelihood estimation.[20]

Other important contributions at this time included Spearman's rank correlation coefficient, which was a useful extension of the Pearson correlation coefficient.[21] Gosset introduced the student's t-distribution,[22] a continuous probability distribution used in situations where the sample size is small, and the population standard deviation unknown. Additionally, Neyman

demonstrated that stratified random sampling was a better method of estimation than purposive (quota) sampling.[23]

By the 1960s, when computers became more widely available, much of the theoretical work in statistics had already been established, providing a solid base for integrating statistical models into ML framework.

Alan Turing, widely regarded as the father of computer science, made foundational contributions to the field and demonstrated the potential of machines to perform complex computations ("On computable numbers").[24] Building on this work the history of ML started in 1943 with the first mathematical model of neural networks (NN) presented by Pitts and McCulloch,[25] which showed the immense computational power of simple elements connected in a neural network. Their work received little attention until these ideas were later applied by John von Neumann, Norbert Wiener, and others.[26] Turing's proposal of the Turing test in 1950, sparked discussions about artificial intelligence and set the stage for the subsequent advancements in ML.[27,28]

In 1952, Samuel is credited with creating the first computer program to play championship-level checkers, and with developing the minimax algorithm, a technique for minimizing losses in games,[29,30] still widely used today. In 1957, Rosenblatt[31] developed the perceptron, one of the first algorithms to use artificial NNs. It was designed to improve the accuracy of computer predictions by adjusting its parameters until it reached an optimal solution. In 1965, Ivankhnenko and Lapa developed the hierarchical representation of NNs that uses polynomial activation function and is trained using the group method of data handling, making it the first ever multi-layer perceptron (MLP).[32] In 1967, Cover and Hart[33] published on nearest neighbors, an algorithm that is still used today to classify an input object into one of two categories. In 1974, in his thesis, Werbos[34] laid the foundations for backpropagation as a method to improve the accuracy of a model by adjusting its weights so that it can more accurately predict future outputs.

In 1995, Ho[35] introduced random decision forests, which involved the construction of multiple trees through the pseudorandom selection of subsets from the feature vector. This approach aimed to mitigate over-fitting and enhance generalization in decision tree (DT) models. This early variation of ensemble learning later evolved into what is now known as random forests (RF). In 2000, Leo Breiman and Adele Cutler made significant contributions to further refine and popularize the RF algorithm, shaping it into the widely used and influential technique it is

today.[36] By 2006, Hinton *et al*.[37] had introduced deep learning, describing the first algorithm that could achieve human-level performance on difficult and complex pattern recognition tasks.

Over the years, all these advancements found practical applications illustrating the remarkable evolution of ML and its impact in various domains (Figure 1).[38] Notable examples include the Stanford Cart (1979),[39] a remote-controlled robot that equipped with computer vision and pattern recognition algorithms successfully crossed a room filled with chairs without human intervention in a few hours. Deep Blue (1997), which defeated chess grandmaster Garry Kasparov.[29,40] IBM's Watson (2011), one of the best AI engines used in healthcare, finance, cybersecurity, law even retail and sports.[41,42] Eugene Goostman (2014), which is the first chatbot regarded to have passed the Turing test. Google's AI algorithms, that beat humans in the game of Go (2016). Waymo's autonomous taxis (2017), and OpenAI's language model GPT-3 (2020).

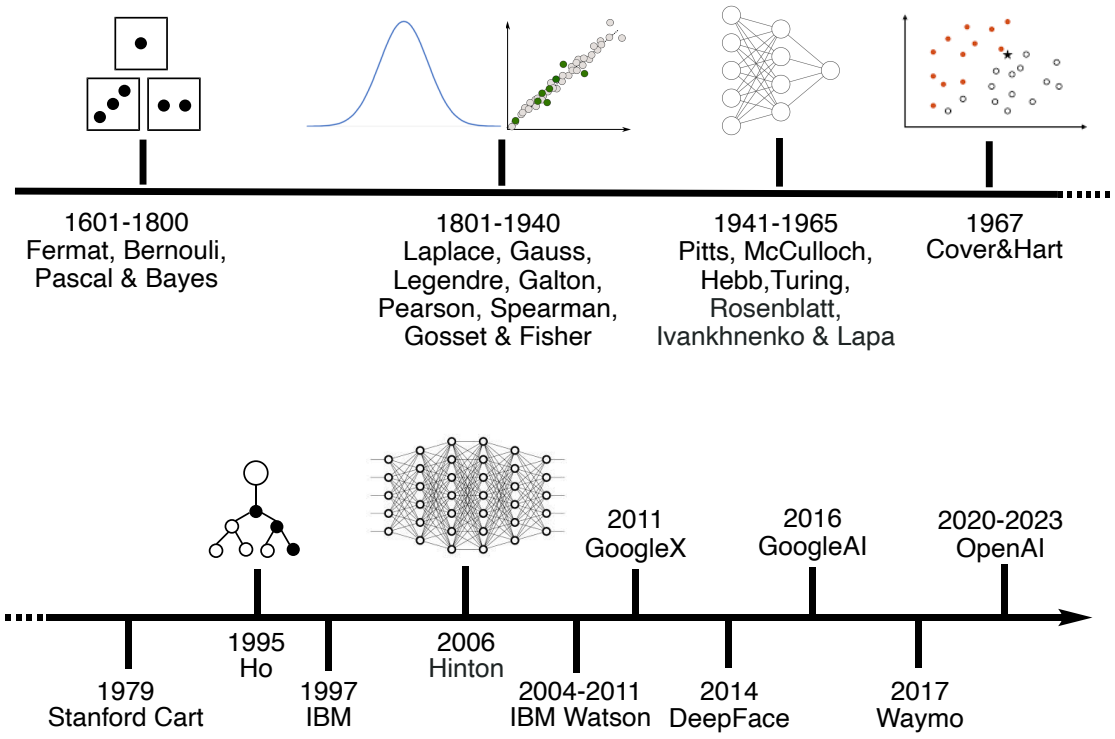


Figure 1: A timeline with the history of statistics and ML applications.

ML has demonstrated its versatility, positioning it as one of the most captivating technologies of our era. At present, almost every common domain is powered by ML, including healthcare,[41,42] the automotive industry, robotics, and computer vision.[38] The popular use of ML algorithms has been facilitated by the availability of automated ML or AutoML tools, which have enabled non-experts to use complex ML algorithms. For example, tasks such as data pre-processing, feature engineering, feature extraction, feature selection, algorithm

selection and hyperparameter optimization can now be performed easily by AutoML tools. However, there are some critical problems to be solved before AutoML has the potential to become a dominant force in the future. They include acquisition, transformation, privacy and security of data, as well as, explainability of ML models, ethical and biases concerns.[43–45]

1.2. Overview of machine learning applications in chemistry

Over the past decades, ML has been applied in various subfields of chemistry,[46–52] including quantum,[53–55] environmental,[56] synthetic,[57–61] analytical,[62] and industrial chemistry,[63] drug discovery and biochemistry,[64–68] and material science.[69–71] In the past 20 years, more than 70,000 journal publications and 17,500 patents have been produced related to ML in chemistry.[72,73]

Some thunderous achievements of ML in Chemistry include: Alpha fold,[74] an AI system developed by DeepMind that predicts protein 3D structures from its amino acid sequence using advanced NN architectures and vast amounts of structural data. DeepChem,[75] an open-source toolkit for deep learning that has been used for molecular property prediction, chemical synthesis planning, and virtual screening. RXN, a tool created by IBM, using deep learning models, which has been used to predict chemical reactions,[76,77] retrosynthesis pathways,[78,79] and experimental procedures.[80,81] Exscientia's AI platform, which utilizes deep and reinforcement learning, and has now designed two drugs that are in Phase 1 of human clinical trials.[82] These notable advancements showcase that ML techniques push the boundaries of scientific discovery and accelerate innovation in the field.

1.2.1. From quantitative-structure-activity-relationships to machine learning

The use of ML techniques in chemistry can be traced back to the early work of Hammett, who laid the groundwork for a data-driven approach, using linear free energy relationships (LFERs) to develop quantitative relationships between structure and activity.[83] While the Hammett equation uses a single parameter for a given substituent, Taft also treated steric, inductive and resonance effects.[84,85] Influenced by Taft's latter work, Charton employed multiple regression to investigate more complex relationships between chemical data.[86]

These early models saw a renaissance in the past decades, with the development of quantitative structure-activity/property relationship (QSAR/QSPR) methods,[87] employed by biologists and medicinal chemists to predict biological activities (or other properties) from structural and topological descriptors, such as molecular shape and size, number of heteroatoms and numbers of hydrogen-bond donors and acceptors (Figure 2). Multivariable linear regression (MLR) is commonly employed in QSAR analysis as it considers multiple independent descriptors to predict an activity or a property. Tropsha, Roitberg and many others[87–90] summarized the recent advances in QSAR highlighting the applicability of algorithms, modeling methods, and validation practices in synthesis planning nanotechnology, materials science, biomaterials and clinical informatics.

This Thesis focuses on the development of multivariable linear regression models for catalyst design. This approach has been popularized by Sigman and co-workers since 2008. Their initial work[91–95] explored the LFER between steric parameters and enantiomeric ratio.[92,96] Over time, they extended the use of MLR for predicting reaction selectivity.[97,98] We refer the reader to the relevant reviews in MLR for further details.[99–103]

Currently, many different ML architectures, beyond MLR, are being used for molecular property prediction, reaction outcome prediction, reaction conditions prediction, reaction optimization, molecular design, and retrosynthesis. In the following paragraphs, we highlight key achievements within each domain. Since not all these domains are directly pertinent to the focus of this Thesis, the discussion will be succinct and primarily oriented toward those most relevant to our study.

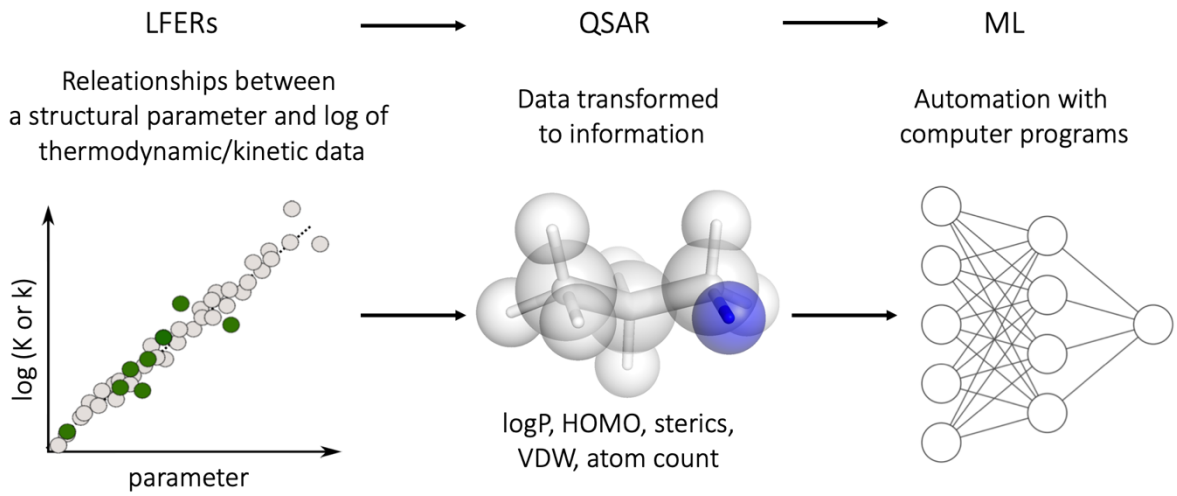


Figure 2: The evolution of data science in chemistry. From linear free energy relationships to machine learning.

Molecular property prediction

Molecular properties such as solubility, viscosity, bioactivity, toxicity, and melting points can be predicted either through equations based on empirical data (e.g., properties are functions of temperature or pressure) or by modeling. More recently, predictions are accomplished by readily available data-driven ML models. Examples include the OPEn structure-activity/property relationship app (OPERA), which has been trained on thousands of compounds with known properties,[104] and Chemprop, which utilizes message passing NNs.[105,106] Both of these tools can predict a variety of physicochemical properties, including octanol-water partition coefficient (logP), water solubility, boiling point, bioactivity, toxicity, and others. In the context of this Thesis, we are interested in predicting solubility and viscosity, consequently, these properties will be discussed in greater detail in the subsequent sections.

Reaction outcome prediction

The synthesis of new molecules is often a costly and time-consuming effort, often done through trial-and error. To tackle this challenge, in 2016, Alán Aspuru-Guzik and co-workers employed NNs for predicting the main reaction product using 1D representations of the reactants as inputs.[107] Since then, convolutional neural networks (CNN) and recurrent neural networks (RNN) have also been trained with databases of thousands of organic reactions to predict main products and byproducts.[108–112]

Reaction conditions prediction and reaction optimization

Reaction conditions, including solvent, catalyst, reagent, temperature, concentrations, reaction time, purification method, have also been explored using ML models. For example, Jensen *et al.*[113] introduced an ML-based program to predict reaction conditions for organic reactions, based on over 11 million data. One limitation is that it does not predict concentrations and reaction times. Another approach by Zare and co-workers[114] uses a RNN that iteratively searches for the best reaction conditions, this was tested for condensation, addition, oxidation, and dehydrogenation reactions. In 2021, Doyle *et al.* developed an open-source software tool to optimize chemical synthesis for Mitsunobu and deoxyfluorination reactions, using Bayesian optimisation.[115] Finally, Zimmerman *et al.*[116] proposed the use of transfer and active learning, with a combination of prior data and new experiments, to accelerate the development of new reactions; they illustrated this in Pd-catalyzed cross-coupling reactions.

Molecular design

ML has emerged as a valuable complementary tool for designing new molecules, as it can generate huge numbers of molecules in a short time. Several ML algorithms, such as variational autoencoders, adversarial autoencoders, RNN and graph convolutional networks (GNC) have been used for molecular design.[117,118] These algorithms generate molecular structures either as SMILES (Simplified Molecular Input Line Entry System) strings or directly as graphs and they are trained on millions of molecules. To support the training and evaluation of ML models for molecular design, several databases have been built. The ZINC[119,120] database with commercially available compounds, the QM9[121,122] dataset which is derived from quantum chemical calculations and provides quantum properties and descriptors for small organic molecules, and the ChEMBL[123–125] database, that focuses on bioactive molecules and their associated biological activities, are only a few such databases that contain millions of molecules and are freely available.

1.3. Machine learning for catalyst design

In the field of catalyst design, a range of ML approaches including MLR, RF, support vector machine (SVM) and NNs have been employed to guide the development of new catalysts accelerating the catalyst discovery. Some efforts focus on building more accurate ML models, while others focus on interpretable ML models. Both avenues are important and offer valuable insights into catalyst design and optimization.[1,97,126–131]

As mentioned before, Sigman and co-workers have pioneered the use of MLR for predicting reaction selectivity for a wide range of reactions. For example, their study on the BINOL-based phosphoric acid catalyzed nucleophilic additions to imines,[97] considered 313 steric and electronic parameters to describe substrates and catalysts for 367 reactions ($R^2 = 0.88$). Further analysis of the transition state geometries for *E/Z* imines, enabled the authors to recover the importance of sterics, with large catalyst and imine substituents leading to higher levels of enantioselectivity for the *E*-over the *Z*-imine model, which was favored with smaller substituents. They also guided the design of Minisci reactions of Diazines by constructing a MLR model for 55 reactions using the same physical-chemical descriptors as before and they obtained an $R^2 = 0.88$ (Figure 3a).[98]

Doyle and co-workers have employed RF to predict reaction yields for 740 alcohol deoxyfluorination reactions (R^2 = 0.93)[132] and 4,608 Buchwald-Hartwig amination reactions (R^2 = 0.92)[133] using physical chemical descriptors (Figure 3b). However, subsequent investigations revealed that this method lacked true feature learning and instead relied on capturing patterns present in the data.[134] In 2021, Luo *et al.*[135] used a deep convolution NN for the same type of reactions, but this time they trained their model on 3,690 reactions and 120 physical chemical descriptors, achieving yield predictions with even higher correlation and low error (R^2 = 0.96 and RMSE = 4.95%). By using PCA they identified the 64 most important descriptors, with the charge of the catalyst having the greatest influence.

In 2018, Sunoj and co-workers, predicted the regiochemical outcomes of 66 regioselective difluorination reactions of alkenes catalyzed by hypervalent iodine. They employed 63 features, such as charges, nuclear magnetic resonance shifts, electrophilic and nucleophilic Fukui indices, steric parameters achieving an average accuracy of 90%. However, the authors noted the limited interpretability of the model, which led them to generate a DT model to identify the most relevant descriptors. From this model, it was found that 1,2-difluorination was favored with electron-deficient benzylic carbons, whereas electron-rich terminal carbons exclusively lead to 1,1-difluorinated products.[136] They later used NNs with 153 physical chemical descriptors for predicting the enantioselectivity (in enantiomeric excess, *ee*) of 240 β-C(sp^3)–H functionalization reactions, obtaining an RMSE of 7.8 % *ee* for the enantioselective arylation of cyclobutyl carboxylic amide, an RMSE of 5.0% *ee* for the alkenylation of isobutyric acid, and an RMSE of 7.1% *ee* for the C(sp^3)–H arylation of free cyclopropylmethylamine (Figure 3c).[137] Finally, they developed a transfer learning protocol, trained on 1 million data, to predict yield and *ee* for Pd-catalysed Buchwald-Hartwing reactions, with low errors (RMSE = 4.9% for yield and RMSE = 8.6% for *ee*).[138]

Denmark *et al.* have utilized SVMs to predict *ee* for the nucleophilic addition of thiols to *N*-acyl imines. The model was built using data from 1,075 reactions, introducing a new shape descriptor named *average steric occupancy*, alongside electronic parameters; this resulted in 16,384 features (3D & 4D descriptors) which were then reduced with the help of PCA. A high correlation (R^2 = 0.91) was achieved and the most selective catalyst (96.5% *ee* within 3% *ee* of the experimental value) was predicted, illustrating the applicability of the model in finding more selective catalysts beyond the training set (Figure 3d).[139]

Hong and co-workers trained a RF model to predict regioselectivity in 8,580 radical C−H functionalization reactions. In total, 50 steric and electronic of descriptors were used, including

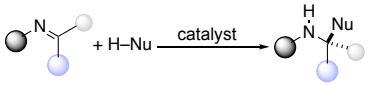
the smooth overlap of atomic positions (SOAP), buried volume, molecular fingerprints, frontier molecular orbital energies, and atomic charge, among others. After analysis, the authors ultimately implemented a predictive model using 32 physical organic descriptors to achieve an accuracy of 90%, advising against the use of SOAP (15,876 descriptors) and molecular fingerprints (1,358 descriptors) which would require a significantly larger training set to match the feature space. The use of a smaller set of physical organic descriptors also facilitated interpretation of the model, which revealed the key role of heteroarenes in reactivity and regiocontrol (Figure 3e).[140]

Corminboeuf and co-workers developed a gaussian kernel ridge regression, to predict the density functional theory (DFT)-computed *ee* of Lewis base-catalyzed propargylation reactions. As features two- and three-body potentials, which derive from the atomic coordinates calculated by quantum mechanics (QM), and molecular fingerprints were used. The authors achieve an almost perfect correlation ($R^2 = 0.97$) between DFT calculations and ML predictions (Figure 3f).[141]

Jensen and co-workers have published over the years several predictive models for physical properties,[142] condition reactions,[113] chemical reactivity[108] and synthesis.[59] One of the first predictive models they reported was a NN model to predict the major product of 15,000 reactions (chlorination, amide synthesis, isoxazole synthesis, sulfamide synthesis, etherification, Suzuki coupling, azidation, and alkylation, among others). The model was constructed with 1,055 features, including Morgan fingerprints, number of hydrogen atoms and atomic number. An accuracy of 71.8% was achieved in predicting the major product (Figure 3g).[109]

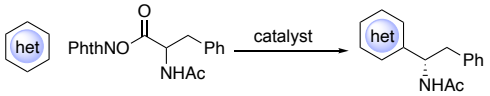
a) Sigman - MLR

Nucleophilic additions to imines - 2019



- 3D descriptors
- 367 reactions
- R^2≃0.88

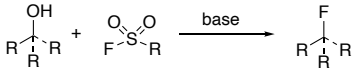
Minisci reactions of Diazines - 2019



- 3D descriptors
- 55 reactions
- R^2≃0.88

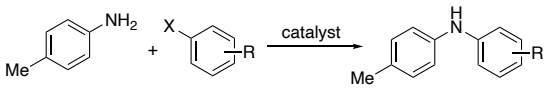
b) Doyle - RF

Alcohol deoxyfluorination - 2018



- 3D descriptors
- 740 reactions
- R^2≃0.93

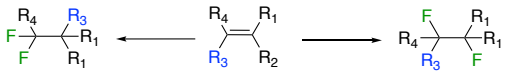
Buchwald-Hartwig amination - 2019



- 3D descriptors
- 4608 reactions
- R^2≃0.92

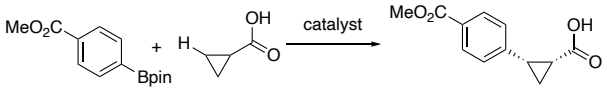
c) Sunoj - NN

Difluorination of alkenes - 2018



- 3D descriptors
- 66 reactions
- R^2≃0.90

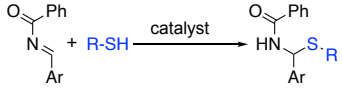
β-C(sp^3)–H functionalization - 2022



- 3D descriptors
- 240 reactions
- RMSE≃0.07 %*ee*

d) Denmark - SVM

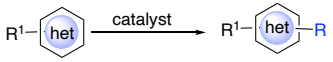
Nucleophilic addition of thiols to *N*-acyl imines - 2019



- 3D & 4D descriptors
- 1075 reactions
- R^2≃0.91

e) Hong - RF

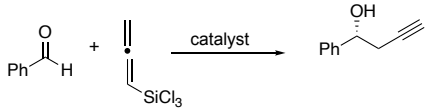
Radical C−H functionalization - 2020



- 3D descriptors
- 8580 reactions
- R^2≃0.90

f) Corminboeuf - RF

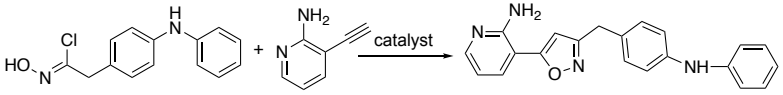
Lewis base-catalysed propargylation - 2021



- 2D & 3D descriptors
- 754 reactions
- R^2≃0.97

g) Jensen - NN

Isoxazole synthesis - 2017



- 1D & 2D descriptors
- 15000 reactions
- R^2≃0.72

Figure 3: Machine learning in organocatalysis.

Overall, the works discussed above demonstrate the power of ML techniques to assist in the prediction of reaction outcomes and guide the design of optimal reaction conditions and catalysts. However, several challenges remain, including:

Datasets. There is a lack of comprehensive, high-quality datasets available for model training. This problem is often exacerbated by the bias towards positive results. This also means that the most robust models are typically produced by larger groups who generate their own data.

Interpretability. This is particularly evident when using NNs, often referred to as 'black box' models, as they make it difficult to rationalize the origin of a specific prediction. This is particularly problematic in cases where researchers unfamiliar with ML may choose inappropriate techniques or misinterpret results.

Generalization. Being able to make predictions beyond the specific datasets on which the models were trained is challenging. Often, test sets that are remarkably similar to the training data are used, this can limit the model's ability to make accurate predictions in novel scenarios. Data leakage, where information from the test set inadvertently influences the training process, is another problem that can undermine the integrity of model validation and lead to overly optimistic performance estimates. Additionally, a common practice is to perform a single random split of data for model validation and cease further evaluation. While convenient, this method can sometimes result in a 'lucky split' that may not accurately represent the diversity and complexity of the data. This risk poses a significant issue as it can lead to the creation of models that appear robust within the narrow confines of their validation set, but perform poorly when faced with new, real-world data.

In response to these challenges, we introduce our workflows for selectivity prediction in organocatalysis in Chapters 3 and 4, aiming to demonstrate how these issues can be effectively addressed and mitigated.

1.4. Ionic liquids

Ionic liquids (ILs) are ionic compounds whose melting point is below 100 °C.[143] They are often formed by an organic cation and an organic or inorganic anion whose bulkiness and asymmetry lead to low lattice energies and consequently low melting points. Moreover, their ionic composition renders them highly polar but still soluble due to the presence of alkyl chains on

the cations. They also have high thermal stability and electrical conductivity, and negligible vapor pressure.

Even though the first IL, ethylammonium nitrate, was reported more than 100 years ago by Paul Walden,[144] the popularity and use of ILs has only become evident in the past two decades.[145] Commonly used cations include alkyl-substituted imidazolium, ammonium, phosphonium, pyrrolinidium, piperidinium, and pyridinium; popular anions include bis(trifluoromethane)sulfonimide, phosphate, and borate. These systems are used in a wide variety of chemical processes including organic synthesis, catalysis, electrochemistry, separation of metals, gas separation, biomass processing, pharmaceuticals, and energy storage devices, such as batteries, supercapacitors, and fuel cells.[146–148] Functioning as catalysts,[149] reagents,[150] and solvents,[151] ILs showcase versatility across chemical processes. Further emphasizing their utility, ILs are often considered as "green solvents" owing to their immiscibility with many other solvents. This characteristic enables easier solvent extraction, thereby enhancing process efficiency and environmental sustainability.

The unique physicochemical properties of ILs also facilitate significant tunability, which emerges from the combination of different cation-anion pairs and variation in the cation core structure, hydrocarbon chain lengths, and functional groups. This makes them powerful alternatives for challenging applications where molecular liquids cannot be used, such as in electrochemical systems, gas absorption processes, and selective separation techniques.[146–148]

Despite their many positive aspects some challenges exist. For instance, while ILs often replace toxic solvents they can themselves be toxic towards aquatic organisms, therefore toxicity testing is essential.[152] ILs can be more expensive than conventional solvents, however the initial increase in capital cost can be offset by improvements in solvent recyclability, catalyst recovery, reaction rates, selectivity, and product separation.[148] The main disadvantage of ILs is their high viscosities compared to water and other widely utilized organic solvents, which can lead to slower diffusion rates or challenges to pump and handle. This undesirable property poses one of the major obstacles to successful applications of these novel solvents (Figure 4a).[153]

Finally, the limited understanding of the underlying mechanisms that govern their behavior means that tuning their properties through rational design remains challenging. Efforts have focused on understanding why ILs deviate from classical liquids and what allows them to be good cocatalysts.[154] To understand ILs scientists have applied a wide range of spectroscopic

techniques, such as Raman, infrared spectroscopy and scanning electron microscopy, which they have combined with electrochemical characterization methods, computational modeling, including molecular dynamics simulations[155–160] and DFT.[161–167] More recently, ML techniques have been employed to predict key properties such as density, toxicity, viscosity, and solubility. In the following paragraphs we delve into these predictive models, and in Chapter 5 we present our methodology for the prediction of viscosity and solubility.

1.4.1. Ionic liquids for carbon capture

Emissions of carbon dioxide (CO_2) into the atmosphere impact the environment as their release causes the greenhouse effect, leading to global warming. Developing technologies that can capture CO_2 are now more pressing than ever. One such technology is carbon capture utilization and storage, where aqueous amines, such as ethanolamine (MEA) and dimethylethanolamine (MDEA) are utilized (Figure 4b).[168] However, these solvents have several disadvantages such as high volatility, high cost, high energy consumption, corrosiveness, and easy degradation.[169]

In recent years ILs have been utilized for carbon capturing as an alternative to amine solvents. The pioneering work of Blanchard *et al.*[170] showed for the first time that IL 1-butyl-3-methylimidazolium hexafluorophosphate $[BmIm]^+[PF_6]^-$ can be employed in CO_2 capture. Since then, a plethora of research in conventional and functionalized ILs has been reported,[171,172] including imidazolium based cations, with PF_6^-,[173–175] tetrafluoroborate (BF_4^-),[173,176,177] and bis(trifluoromethylsulfonyl)imide (TF_2N^-)[177,178] (Figure 4b).

Experimental and modeling studies have demonstrated that in conventional ILs, the anion plays a crucial role in the dissolution of CO_2, while the cation has a secondary role, due to its relatively weaker interactions with CO_2.[179,177–179] Different IL systems show that different aspects contribute to CO_2 uptake. For example, in TFA^- based ILs, this is determined by the acid-base interaction between acetate and CO_2, in 1-hexyl-3methylimidazolium tris (pentafluoroethyl) trifluorophosphate ($[HmIm]^+[FEP]^-$), the amount of free volume in the IL system is key, while in $[HmIm]^+[PF_6]^-$ and $[HmIm]^+[FEP]^-$ CO_2 uptake seems to be determined by hydrogen and halogen bonding interactions (Figure 4b).[172]

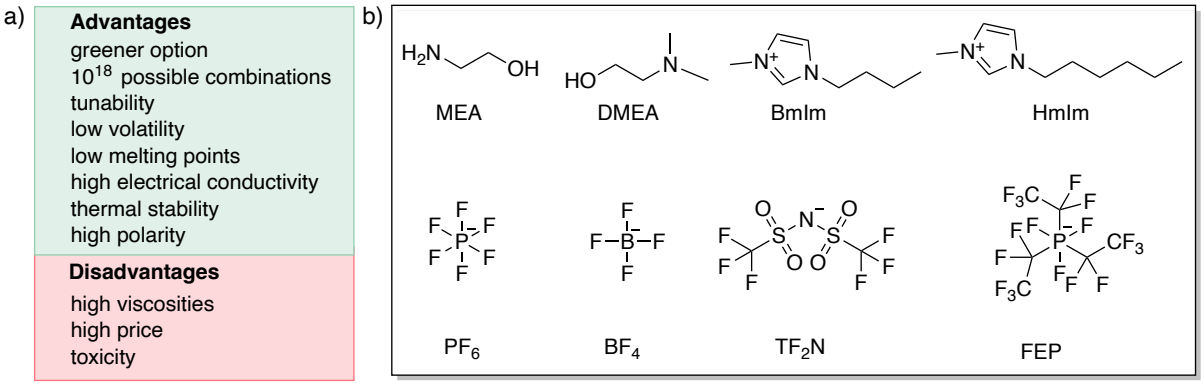
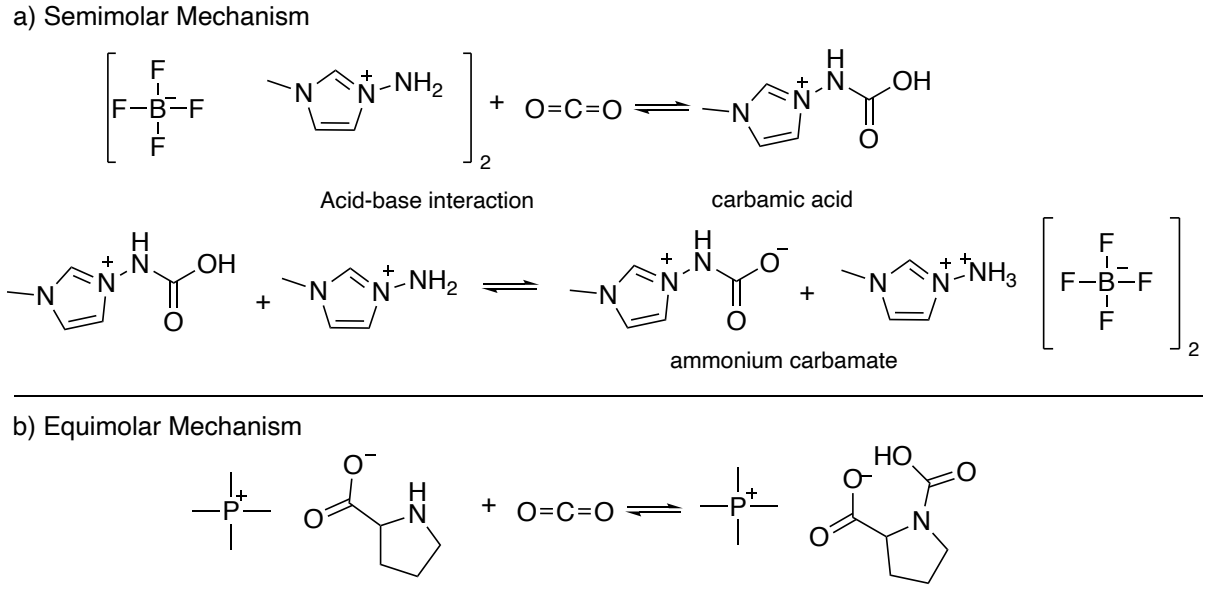


Figure 4: a) Advantages and disadvantages of ILs, b) Common molecules used for carbon capturing. MEA and DMEA are amine solvents; BmIM, HmIm commonly used cations; PF_6, BF_6, TF_2N, FEP commonly used anions.

The capture of CO_2 by amine compounds can follow either a *semimolar* mechanism, where the cation is tethered with amine moiety, or an *equimolar* mechanism, where the amine is part of the anion (Scheme 1).[180–182] In the former, the CO_2 first reacts with the amine moiety to form carbamic acid, which further reacts with another molecule of amine to yield ammonium carbamate as the final product. Thus, for one mole of CO_2, two moles of amine are required. In the *equimolar* mechanism, one mole of amine captures one mole of CO_2 and stabilizes the product as carbamic acid.[183] Various studies have claimed *multimolar* absorption in ILs because of the presence of a stabilizing group in the vicinity of NH_2-moiety of the amine.[184]



Scheme 1: Carbon capture mechanisms. a) Semimolar mechanism adapted from Davis *et al*.,[180] and b) Equimolar mechanism adapted from Brennecke *et al*.[182]

1.4.2. Machine learning models for ionic liquids property predictions

ILs have been extensively studied using ML techniques, fostering the development of robust predictive models for their physicochemical properties. In 2006, Zhang *et al*. produced a

database of physical properties for 588 ILs obtained from literature. It included information of melting, glass transition, decomposition, freezing, and clearing points, as well as density, viscosity, surface tension, conductivity, polarity, and electrochemical windows. The authors noticed the lack of physical chemical property data and significant deviations for identical entries from different sources.[185]

Zhang *et al.* generated a QSPR model employing electrostatic, QM, and topological descriptors to predict melting points for 19 $[Im]^+[BF_4]^-$ and 29 $[Im]^+[PF_6]^-$ based ILs. The test sets were produced randomly, leaving three compounds as the test set for the $[Im]^+[BF_4]^-$ and four compounds as test set for $[Im]^+[PF_6]^-$, resulting in high correlations ($R^2 > 0.9$).[186] In 2007, Soloven and co-workers explored a series of algorithms and descriptors to predict the melting points in 717 bromides of nitrogen-containing organic cations; unfortunately their model was poor ($R^2 < 0.7$).[187] Later Lazzus employed a group contribution (GC) method on an experimental data set of 400 ILs to predict melting temperatures, obtaining high correlation ($R^2 \approx 0.9$).[188] Toreccilla and co-workers[196] employed NNs with 3D descriptors, to predict melting points for 97 ILs (test set was randomly chosen constituting 15% of the total sample), achieving high correlation ($R^2 > 0.9$). Finally in the same year Bini *et al.* [197] predicted melting points for 126 ILs (26 of which were used as a test set) employing NNs and graph theory and achieved good correlations ($R^2 \sim 0.8$). It is important to note that the datasets employed in these studies were small and extrapolation on these models should be done cautiously.

The density of ILs is another property that has been explored over the past years. Wang *et al.* used a GC method to predict densities of imidazolium-based ILs over a range of temperatures and pressures achieving good accuracies (AARD[1] < 0.6%).[189] In 2012, Paduszynski and Domanska, generated a GC method to predict densities in a variety of temperatures and pressures, with a dataset of 18,500 data points (1,028 unique ILs), resulting to an AARD = 0.45%, which is the one of the lowest values compared with similar correlations reported in literature. Moreover, they were the first ones to make entire dataset openly available.[190] In 2019, Paduszynski published a new GC scheme,[191] this time including 41,250 data points and achieving small errors AARE[2] = 0.9%. Shirazian and co-authors combined a support vector

[1] AARD is the average absolute relative deviation and is defined as: $AARD = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\widehat{y_i}-y_i}{y_i}\right| \times 100\%$. In general, a lower AARD value indicates a more accurate model, while a higher AARD value indicates a less accurate model. It shows the average deviation of the predictions from the actual values.

[2] AARE is the average absolute relative error and is defined as: $AARE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\widehat{y_i}}{y_i} - 1\right| \times 100\%$. It is expressed as a unitless metric, quantifies average error as a proportion of actual values, thereby providing a means to evaluate prediction errors relative to the magnitude of the values being predicted.

regressor and GC to predict density of 918 ILs in different pressures and temperatures, also achieving perfect correlations ($R^2 = 0.99$).[192]

The accurate evaluation of the toxicity of ILs is crucial to accurately determine the environmental impacts of these compounds.[193–195] In recent years, several ML models have been reported for the prediction of toxicity. For example, Mujtaba *et al.* reported a MLR model employing 14 QM-based descriptors, to predict toxicities for 17 ILs (9 were used as a test set) achieving a perfect correlation.[196] With such a small dataset it would be inappropriate to assume that their model can extrapolate beyond the IL families included in the study. Later Zhao and co-workers generated MLR, SVM and NN models employing 119 ILs (with a 80%-20% data split to train and test sets) and QM based descriptors to predict toxicity, achieving high correlations ($R^2 \approx 0.90$).[197] In 2022, Chong *et al.* presented a probabilistic model built from deep kernel learning to predict toxicity. The model was built using 24 structural descriptors obtained from RDKit,[198] for 155 ILs (140 for training and 15 for testing, split randomly), resulting in low errors and high correlation (RMSE = 0.23 log EC_{50} and $R^2 = 0.94$). Moreover, the authors made the model freely available through a web-based tool.[199] Finally, Yan and co-workers developed *ILTox*,[200] an online curated dataset containing toxicities of 1,183 ILs (1,199 today) in different conditions (leading to 6,700 data points; 6,726 today), which was used to build a QSPR model, employing MACCS (Molecular ACCess System) fingerprints and PCA. Once the predicting capabilities of the model were established, they screened 8 million ILs. Tools like OPERA,[104] ToxiM[201] and TEST[202] can also be used to predict toxicity; however, to the best of our knowledge, they have not been used for ILs.

Viscosity predictions

Viscosity, in particular the dynamic viscosity (η), describes a fluid's internal flow resistance. Simple liquids obey the Arrhenius law, according to which plotting viscosity against the reciprocal of temperature (1/T) yields a straight line. However, this is not always true for ILs, for which viscosity is better described by the Vogel-Fulcher-Tammann (VFT) equation (Eq. 1.1):

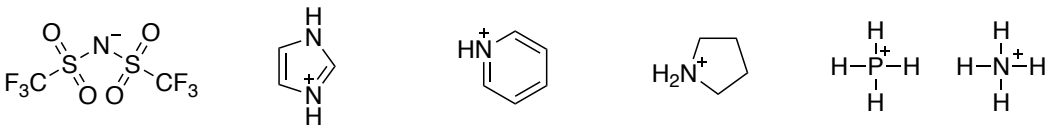
$$\eta = A \exp\left(\frac{B}{T - T_0}\right), \quad (1.1)$$

where A, B, and T_0 are specific adjustable parameters.[203,204]

Viscosity is a key property that needs to be considered when exploring potential ILs for CO_2 capture, as the use of ILs with high viscosity adversely affects mass transfer and power requirements,[205,206] as well as decreases the rate of the reaction.[203]

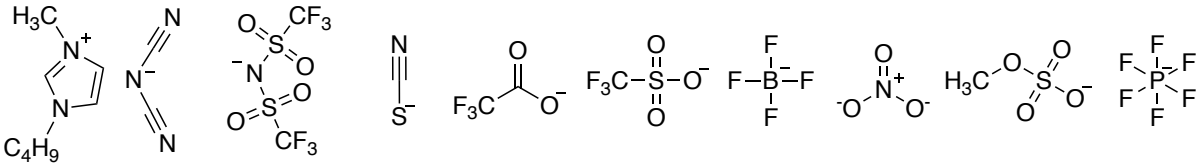
Jacquemin *et al.*[207] have studied the effect of different cations paired to the common $[TF_2N]^-$ anion, observing the following trend for viscosity: $[imidazolium]^+ < [pyridinium]^+ < [pyrrolidinium]^+ < [PH_4]^+ < [NH_4]^+$. They also report the effect of the anion using the $[BmIm]^+$ cation, with viscosity increasing as follows: $[DCA]^- < [TF_2N]^- < [SCN]^- < [TFA]^- < [TFO]^- < [BF_4]^- < [NO_3]^- < [MeSO_4]^- < [PF_6]^-$. The relatively high viscosity of $[BmIm]^+[PF_6]^-$ is interesting, since fluorinated ILs generally show lower viscosities (Scheme 2). Finally, they reported that viscosity increases almost linearly with alkyl-chain length for all ILs.

a) Effect of the cation for $[TF_2N]^-$ anion with viscosity showing the following order



$[TF_2N]^-$ $[imidazolium]^+ < [pyridinium]^+ < [pyrrolidinium]^+ < [PH_4]^+ < [NH_4]^+$

b) Effect of the anion for $[BmIm]^+$ cation with viscosity showing the following order



$[BmIm]^+$ $[DCA]^- < [TF_2N]^- < [SCN]^- < [TFA]^- < [TFO]^- < [BF_4]^- < [NO_3]^- < [MeSO_4]^- < [PF_6]^-$

Scheme 2: Viscosity of ILs depending on a) the cation and b) the anion.

Additionally, the viscosity of ILs exhibits a distinct pressure dependence, a complex relationship that has yet to be fully characterized by a comprehensive model. Despite numerous attempts, the intertwining factors of viscosity, temperature, and pressure in ILs remain a challenging puzzle. Various theoretical and empirical models have been proposed over the years, each striving to encapsulate this relationship with increasing precision. It is important to note that properties such as density and viscosity are also influenced by the amount of water and other impurities present in the IL.[208]

Initial models used theoretical approaches (hole theory),[209] molecular-based (volume approach),[210] thermodynamic based information or simple empirical correlations between viscosity and other thermophysical properties,[211] while more recent approaches have moved to generalized correlations,[212] QSPR models,[153,213–218] GC methods,[219–222] and combinations of

those.[223] In the ensuing discussion, we explore some of the most significant contributions in this field from recent years. A summary of these studies is also encapsulated in the accompanying table for convenient reference (Table 1).

Yu *et al.*[224] developed eight QSPR models, employing structural descriptors at different temperatures, to predict the viscosity of 344 $[TF_2N]^-$ based ILs, achieving high correlations ($R^2 > 0.82$). The study identified interionic electrostatic and hydrogen-bonding (HB) interactions as the major factors affecting IL viscosity. In molecular solvents viscosity is primarily influenced by intermolecular HBs interactions and steric attributes. It was also noted that the significance of these interionic interactions to viscosity fluctuates with temperature; for example, Van der Waals (vdW) interactions become more prominent at lower temperatures, whereas sterics or geometric factors become more substantial at higher temperatures.

Chen and colleagues[223] established a QSPR model employing GC for the viscosity prediction of 26 imidazolium-based ILs across varying temperatures, resulting to 304 data points. Given the absence of a designated split between a training set and a test set, it can be inferred that all the data points were utilized for model training, hence the reported correlation ($R^2 = 0.99$). However, without independent validation, the predictive power of this model cannot be accurately assessed, limiting its informative value.

Mirkhani and Gharagheizi[225] established a simple QSPR model which included 435 data points of 293 ILs in different temperatures. Although the specific quantity and nature of descriptors used remain unclear, the model's validation process appears to be robust, instilling confidence in the reported results (RMSE = 0.23 cP).

Zhao *et al.*[226] constructed two QSPR models (MLR and SVM) employing the Sσ-profile descriptor. These models were developed using a dataset of 1,502 experimental viscosity data points, and in contrast to the models mentioned above, which did not include information about pressure, covered a broad range of both temperatures and pressures for 89 different ILs. For the test set 297 data points were chosen randomly. The SVM model yielded high correlation ($R^2 = 0.94$, AARD = 6.58%), while the MLR model produced slightly lower correlations ($R^2 = 0.80$, AARD = 10.68%).

Gharagheizi and colleagues[227] utilized a GC along with MLR to predict the viscosity of 443 ILs in a wide temperature range at atmospheric pressure. The model was trained and evaluated on a dataset comprising 1,672 data points, 336 of which were reserved for testing. The results revealed a robust correlation ($R^2 = 0.87$, AARD = 6.32%). Lazzus and Pulgar-Villaroel[228]

established a MLR using a GC, which included 1,445 data points for 326 ILs in a wide range of temperatures and achieved high correlation (R^2 = 0.94) for the test set (335 datapoints). In 2018, Yan *et al*.[229] employed 64 descriptors arising from molecular graph theory and MLR to predict viscosity for 3,228 data points, in a wide range of temperatures and pressures for 349 ILs. The data were randomly divided into training and test sets, with 2,591 data points chosen as training set and the remaining 637 data points being chosen as the test set, achieving high correlation (R^2 = 0.96 and AARD = 4.6%).

In 2014, Paduszynski *et al*.[153] compiled and revised a comprehensive set of literature data on viscosity as a function of temperature and pressure for 1,484 ILs leading to 13,470 data points. By 2019 this dataset was further expanded to encompass 1,974 ILs, bringing the total data points to 15,372, however this time all data points were taken at atmospheric pressure.[230] In this work the GC scheme developed for their previously referenced work on density predictions[191] was used, coupled with a two-stage modeling protocol where viscosity was calculated using a reference term derived from a NN, and a temperature correction computed with a SVM model. Both internal and external validation techniques were implemented to ensure the robustness of the scheme. Despite achieving satisfactory accuracies, the author acknowledged the challenges associated with modeling viscosity in this manner, that is why the AARD for each IL family was reported separately, with some results being deemed more reliable than others. Lastly, a classification model was developed, demonstrating an overall accuracy of 87% for the proposed methodology.

Table 1: Summary of the relevant literature for viscosity predictions. It includes the publication, the number of datapoints tested, the method used, and the reported metrics. The Reliability column is based upon efficient validation as reported by the authors. "Not tested" is for models that no validation was performed, "Limitations" is for models that only a random split was performed, and "Yes" is for models that were validated sufficiently.

Authors	*Data*	*Method*	*Metrics*	*Reliability*
Yu *et al.*[224]	344	QSPR	$R^2 > 0.82$	Not tested
Chen *et al*.[223]	304	QSPR + GC	$R^2 = 0.99$	Not tested
Mirkhani & Gharagheizi[225]	435	QSPR	RMSE = 0.23 (cP)	Yes
Zhao *et al*.[226]	1502	QSPR+ MLR	$R^2 = 0.94$, AARD = 6.6	Limitations
		QSPR +SVM	$R^2 = 0.80$, AARD = 10.7	Limitations
Gharagheizi *et al*.[227]	1672	GC +MLR	$R^2 = 0.87$, AARD = 6.3	Yes
Lazzus & Pulgar[228]	1445	GC +MLR	$R^2 = 0.94$	Yes
Yan *et al*.[229]	3228	Graphs + MLR	$R^2 = 0.96$, AARD = 4.6	Limitations
Paduszynski[230]	15372	GC + NN+SVM	Accuracy 87%	Yes

While useful, each of the models discussed above presents different limitations. For example, models employing physical chemical descriptors, while providing some interpretability, are costly as they require QM calculations to compute these descriptors. Furthermore, many of

these physical chemical models do not predict the temperature-dependence of the viscosity, or they are valid only for specific families of ILs. On the other hand, QSPR involving only GC descriptors are easy to compute because all the occurrences of groups can be easily identified from the chemical structures of the cation and anion. However, the lack of clarity and consistency in reported models (e.g., using awkward definition of groups) makes it difficult for users to employ them for their own predictions successfully. To our knowledge there has not been an end-to-end model to predict viscosity, with regression algorithms without having to model separate reference and correction terms, or without having to define several GC before getting the best results. In Chapter 5, we present our methodology which was tested on the 2019 dataset of Paduszynski.

Solubility predictions

The solubility of CO_2 in ILs is a key characteristic determining their efficacy in carbon capture and sequestration applications. Solubility is influenced by various factors, including the nature of the IL, the partial pressure of CO_2, and the temperature.

A fundamental law describing the solubility of gases in liquids is Henry's Law. It states that at a constant temperature, the amount of gas that dissolves in a liquid is directly proportional to the partial pressure of the gas above the liquid. The proportionality constant is known as Henry's Law constant, and it depends on the nature of the gas, the liquid, and the temperature.[231] The solubility of CO_2 in ILs can be expressed as mole fraction (dimensionless), molality (mol $CO_2 \cdot kg^{-1}$), and volume concentration (mol $CO_2 \cdot L^{-1}$). The mole fraction unit has been adopted by most of the references relevant to gas solubility in ILs because it can reflect the molecular interaction between gas and IL.[232]

In terms of the IL composition, ILs containing fluorinated groups (either in the anion or the cation) have been shown to exhibit higher CO_2 solubilities. This can be attributed to the high electronegativity of fluorine, which strengthens the ion-dipole interactions between the fluorinated IL and the CO_2 molecules.[204,233] It has also been observed that vdW forces between the ions dominate the behavior of CO_2 dissolution in ILs, with electrostatic interactions and HB having secondary importance.[234] The fluorination of the cation has also been shown to improve solubility, although to a lesser degree than anion-fluorination. Finally, the CO_2 solubility increases slightly by increasing the alkyl chain length on the imidazolium cation. This increase can be attributed to enhanced dispersion forces between the CO_2 molecules and the IL, which subsequently promote greater CO_2 uptake.[235]

Various models based on thermodynamic and molecular parameters have been developed to predict the solubility of CO_2 in ILs. A commonly used method for predicting activity coefficients in liquid mixtures is UNIFAC (UNIQUAC Functional Activity Coefficients) which is based on GCs. However, it can be inaccurate as it does not account for long-range interactions between different groups in the mixture. Another approach based on QM calculations is COSMO-RS (COnductor-like Screening MOdel for Realistic Solvation). While it can be highly accurate, it requires a significant number of computational resources, making it time-consuming and computationally expensive for larger systems. Additionally, COSMO-RS does not account for the effects of intermolecular interactions on molecular structure and properties, which can limit its applicability in certain scenarios.[236]

Several ML models to predict solubility in ILs have been reported to date. Sedghamiz *et al.*[237] developed a NN to predict CO_2 solubilities for 2,930 data (39 ILs), and H_2S solubilities for 664 data (14 ILs) in different temperatures and pressures. Eslamimanesh and co-workers[238] developed a NN to predict CO_2 solubilities for 128 data points of 24 ILs at different temperatures and pressures. Tatar and co-authors[239] generated four ML models (three different NNs and a SVM) to predict CO_2 solubility in 14 ILs (728 data in different temperatures and pressures). Jia and co-workers,[240] used deep learning to predict 218 CO_2 solubilities for 13 different ILs. In all cases the same input parameters were used, including pressure, temperature, IL critical temperature and pressure, and the acentric factor ω[3], and molecular weight, achieving high accuracy; however, these models were only tested using one random train-test dataset split; raising the question about their true predictive power and ability to extrapolate to unseen data.

In 2020, Zhou *et al.*[241] compiled a comprehensive database of 10,116 CO_2 solubility data measured in various types of ILs, temperature and pressure ranges. This is the largest dataset available including all publications until 2020. They generated a NN-GC and an SVM-GC model, both of which achieved low error (R^2 = 0.98 and MAE = 0.02). The authors admit to the difficulty of building a GC model as the molecules must be decomposed into building groups in advance, manually. Moreover, the authors made the MATLAB code freely available; however, a closer look to the dataset they reported indicates that > 600 data points are duplicated, many of which coexist between train and test set.

[3] acentric factor is a conceptual number introduced by Kenneth Pitzer in 1955, proven to be useful in the description of fluids.[509]

A notable recent contribution by Farimani and colleagues explores the use of graph neural networks (GNNs) to investigate CO_2 solubility in ILs.[242] Their work, which closely resembles the research presented in Chapter 5 of this Thesis, focuses on the development of fingerprint-based, GC-based, and GNN-based models for predicting CO_2 absorption in ILs. Their dataset comprises of the same 10,116 data points mentioned in the work of Zhou *et al.*[241] The authors achieved remarkable results, surpassing previous ML models (MAE = 0.01 and R^2 = 0.99). To construct the input graph for the GNN models, the authors create a unified undirected graph that combines the anion and cation components of the IL. They employ different GNN architectures such as graph convolutional networks (GCNs), graph attention networks (GATs), and graph isomorphism networks (GINs).

They also developed an IL explainer, which aims to identify the important subgraph and determine the contributions of different fragments within the IL molecule to the prediction. To do so, the authors adopted a graph classification perspective, where a separate model is used for explanation rather than prediction. This approach is chosen due to the limited availability of explainability techniques for GNNs in regression tasks.

At the time of publication of the aforementioned paper, our research objective had already been defined and our workflow established. Consequently, we decided to explore and evaluate the tool developed by Farimani *et al.* to comprehend its functionality and assess the effectiveness, reliability, and performance of their workflow. Throughout our investigation, we identified duplicate instances within their dataset. Although their results are reproducible, it is noteworthy that the training process is time-consuming when executed on a CPU (hours of training to reach convergence). Additionally, their approach to splitting the dataset into train and test sets appears to lack thorough attention, as they simply employ a random split. Therefore, their reported results may vary if a different test set is utilized.

Furthermore, when we attempted to apply their tool to our own data, we encountered difficulties. While the authors shared the format of their original data, which includes SMILES representations for the cation and anion, and temperature and pressure values, they subsequently converted the data into a NumPy file format containing only numerical values. We expected the converted format to retain the SMILES representations as strings. However, the process of accomplishing this conversion was not clearly elucidated, posing a challenge to the further utilization of their tool.

1.5. Thesis aims and outline

In this chapter, the journey from statistics to ML has been introduced. Through a comprehensive literature review, existing ML methodologies and their applications in organocatalysis and ILs, have been showcased. This literature review serves as a crucial foundation for the subsequent chapters, allowing us to build upon the existing knowledge and contribute to the field.

In Chapter 2, the theoretical background of the methods employed in the Thesis is presented. First the principles of DFT are introduced, followed by a discussion on ML algorithms. This chapter aims to provide a comprehensive understanding of the methods utilized throughout this Thesis, as well as suggestions for best practices.

Chapter 3 introduces *Pythia*, the ML toolkit, which builds upon the methods and concepts discussed in Chapter 2. The toolkit utilizes a range of input features (fingerprints, Mordred, and QM descriptors) for training and testing models. It employs a range of shallow learners and ensemble models to tackle regression and classification tasks. One of the key advantages of *Pythia* is its accessibility and ease of use. Developed in Jupyter Notebooks, it enables researchers with varying degrees of programming experience to interact with the code and customize it to suit their needs.

In Chapter 4, the general applicability of *Pythia* is illustrated in the context of selectivity prediction in organocatalysis. Specifically, we investigate three organocatalytic reactions, namely the enantioselective formation of β-fluoramines, the Strecker synthesis of α-amino acids, and the Pictet-Spengler cyclisation of hydroxylactams. We describe the process of feature engineering and selection and discuss the quality and interpretability power of the models.

Finally, Chapter 5 explores the application of a GNN for the prediction of viscosity and CO_2 solubility in ILs. We present the architecture of the developed GNN model. Then through a comprehensive evaluation and comparison with existing approaches, we shed light on the effectiveness of GNN models and their potential applications in this domain. Furthermore, we provide detailed insights and analysis of the GNN performance, enabling a deeper understanding of their predictive capabilities. Finally, we rigorously challenge our models by testing their performance on diverse datasets, pushing the boundaries of their predictive capabilities.

2. Methods and theory

2.1. Introduction to electronic structure methods

2.1.1. Ab initio methods

Computational chemistry is a well-established tool for analyzing the molecular properties of complex chemical systems. QM, or first principles or ab initio methods, are being used to calculate molecular and periodic systems containing up to thousands of electrons, using a variety of approximations depending on the system size.

The Schrödinger equation governs the behavior of nuclei and electrons of atoms and in its time-independent form is expressed by (Eq. 2.1):[243]

$$\widehat{H}\Psi(\boldsymbol{r}) = E\Psi(\boldsymbol{r}), \quad (2.1)$$

where $\widehat{H}$ is the system's Hamiltonian describing the kinetic and potential energies of all particles described by the wavefunction (Ψ) with energy (E). This equation can only be solved exactly for single-electron systems; for larger molecules, approximations must be introduced. Given that the nuclei are much heavier than the electrons, one can independently solve the equations describing the motion of electrons, whilst keeping the nuclei fixed, which is known as the Born-Oppenheimer approximation.[244] This decoupling results in the nuclear kinetic energy term of the Hamiltonian being neglected and the nuclear repulsion to become a constant, giving rise to the Born-Oppenheimer Hamiltonian (Eq. 2.2):

$$\widehat{H} = -\frac{1}{2}\sum_{i=1}^{N}\nabla_i{}^2 - \sum_{i=1}^{N}\sum_{A=1}^{M}\frac{Z_A}{\boldsymbol{r}_{iA}} + \sum_{i=1}^{N}\sum_{j>i}^{N}\frac{1}{\boldsymbol{r}_{ij}} + \sum_{A=1}^{M}\sum_{B>A}^{M}\frac{Z_A Z_B}{\boldsymbol{r}_{AB}}. \quad (2.2)$$

The first term corresponds to the electron kinetic energy, the second defines the electron-nuclear Coulomb attraction, the third the electron-electron Coulomb repulsion, and the fourth the internuclear Coulomb repulsion (*i, j* represent electron indices, *A, B* are nuclear indices). We note that the third term implies correlation between the electrons, in that the interaction between electron *i* and electron *j* depends on their relative positions. The correlated nature of electrons makes such a many-body problem intractable; to overcome this the Hartree-Fock (HF) approximation can be invoked. This approximation treats each electron individually; each electron experiences an interaction with an averaged field representing the electrons. Utilizing

this method, each electron can be described by a one electron wavefunction otherwise known as an orbital. We can then construct a molecular wave function as a product of the one electron functions. This is known as the Hartree product (Eq. 2.3):[245]

$$\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3 \ldots \boldsymbol{r}_n) = \psi_1(\boldsymbol{r}_1)\psi_2(\boldsymbol{r}_2)\psi_3(\boldsymbol{r}_3)\ldots\psi_n(\boldsymbol{r}_n). \quad (2.3)$$

However, Eq. 2.3 has a major problem. One of the fundamental principles of QM, the Pauli exclusion principle, sets a requirement for all fermionic wave functions, of which electrons are one species of fermion, to be antisymmetric with respect to interchange. The Hartree product does not satisfy the antisymmetry requirement for the space and spin coordinates of the electrons. Electrons have a spin of ± 1/2 defined as α spin and β spin. If spin is included in the orbital definition, then the orbitals are defined as spin orbitals (χ_i). The molecular wave function must be anti-symmetric with respect to interchange of an electron's space and spin coordinates. Additionally, the Hartree product also requires that the electrons are distinguishable; this is not allowed in QM as electrons are by definition indistinguishable particles. These issues can be solved mathematically by representing the wave function in a Slater determinant (Eq. 2.4). The columns of a single Slater determinant represent the atomic orbitals (AO) and the rows represent the electron coordinates. As a result, each electron is at some point placed in each orbital, hence making them indistinguishable. The pre-factor is a normalisation.[245]

$$\psi_{HF}(\boldsymbol{r}) = \frac{1}{\sqrt{n!}}\begin{vmatrix} x_1(\boldsymbol{r}_1) & \cdots & x_3(\boldsymbol{r}_1) \\ \vdots & \ddots & \vdots \\ x_1(\boldsymbol{r}_n) & \cdots & x_3(\boldsymbol{r}_n) \end{vmatrix} \quad (2.4)$$

The HF approximation is a valuable one to make, as it makes it possible to "solve" the electronic Schrödinger equation. However, we do not obtain the exact solution, because the true wavefunction for a many-electron system is not the HF wavefunction. For an exact solution $\Psi(\mathbf{r})$ to the Schrödinger equation, the energy E obtained by solving Eq. 2.1 can be rewritten according to Eq. 2.5:

$$E = E \times \int \Psi^2(\boldsymbol{r})dr = \int \Psi(\boldsymbol{r}) E\, \Psi(\boldsymbol{r})dr = \int \Psi(\boldsymbol{r})\widehat{H}_{\text{electronic}}\Psi(\boldsymbol{r})dr. \quad (2.5)$$

In this expression the first equality is because the wavefunction is normalized (the sum of probabilities associated with all possible combinations of coordinates, or the integral over all possible values of *r* of the square of the wavefunction, is equal to 1). For the second equality, because *E* is a constant, it has simply been inserted into the integral, and the square has been written out explicitly. For the third equality the fact that Ψ is a solution of the Schrödinger equation, so that $E\Psi = \widehat{H}\Psi$, has been used. Given that the energy is not the true energy but

rather an approximation as explained above, one can define the "energy" as E_{approx} corresponding to a given approximate wavefunction Ψ_{approx} (Eq. 2.6):

$$E_{\text{approx}} = \int \Psi_{\text{approx}}(\boldsymbol{r})\widehat{H}_{\text{electronic}}\Psi_{\text{approx}}(\boldsymbol{r})dr. \qquad (2.6)$$

It can be shown that this approximate energy must be higher than the true ground state energy of the system. This is the variational principle, and it is central to the HF theory. The HF wavefunction of a given system is defined as the Slater determinant composed of the set of orthogonal molecular orbitals (MO) that return the lowest possible energy and it will be the energy closest to the exact one.

Inserting the Slater determinant expression of Eq. 2.4 for a system with n electrons into Eq. 2.6 leads after much algebra to the following expression for the energy:

$$E = \sum_{i=1}^{n} h_{ij} + \sum_{i=1}^{n}\sum_{j=i+1}^{n}\left(J_{ij} - K_{ij}\right). \qquad (2.7)$$

Here h_{ij}, J_{ij} and K_{ij} refer to various integrals carried out over the different MO. Each of the integrals h_{ij} only depends on the shape of a single MO χ_i or ψ_i and these integrals are called "one-electron" terms. They provide a measure of the energy due to Coulombic interaction between an electron occupying that orbital and the positively charged nuclei in the system, summed with the kinetic energy of the electron. The integrals J_{ij} require consideration of two MO each and provide a measure of the Coulombic repulsion energy between the electrons occupying these two orbitals:

$$J_{ij} = \iint \psi_i(\boldsymbol{r}_1)\psi_j(\boldsymbol{r}_2)\frac{1}{|\boldsymbol{r}_1 - \boldsymbol{r}_2|}\psi_i(\boldsymbol{r}_1)\psi_j(\boldsymbol{r}_2)dr_1 dr_2 = \left\langle \psi_i\psi_j \middle| \boldsymbol{r}_{12}^{-1} \middle| \psi_i\psi_j \right\rangle. \qquad (2.8)$$

Here $|\boldsymbol{r}_1 - \boldsymbol{r}_2|$ is the distance between points $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ also written $\boldsymbol{r}_{12}$.

Finally, the integrals K_{ij} also depend on two orbitals, and have an expression similar to that of J_{ij} but have a smaller magnitude, always > 0, so that the K term in the HF equation lowers the energy of the system. They provide a correction to the J_{ij} Coulombic repulsion between electrons that arise due to the antisymmetrization of the wavefunction.[246]

2.1.2. Density functional theory

DFT reformulates Eq. 2.1 into one that is easier to solve.[247,248] DFT is based upon two fundamental theorems, proposed by Hohenberg and Kohn: (i) the ground state energy from

Schrödinger equation is a unique functional of the electron density, and (ii) the electron density that minimizes the energy of the overall functional is the true electron density corresponding to the full solution of the Schrödinger equation.[249]

However, the true leap came when Kohn and Sham showed that the many-body electron problem, in the presence of the nuclei, can be solved self-consistently in terms of a set of non-interacting particles in an effective potential. This led to the Kohn-Sham one electron equation (Eq. 2.9):[250]

$$\left[\frac{-\hbar^2}{2m}\sum_{i=1}^{N}\nabla^2 + V(\boldsymbol{r}) + V_H(\boldsymbol{r}) + V_{XC}(\boldsymbol{r})\right]\varphi_i(\boldsymbol{r}) = \varepsilon_i\varphi_i(\boldsymbol{r}). \qquad (2.9)$$

Here, the terms within the bracket denote the kinetic energy of an electron, the interaction potential of an electron with surrounding nuclei, the Coulombic interaction of the electron with surrounding electrons, and the exchange-correlation potential. The last term V_{XC} compiles the missing interactions upon transforming a many-body electron problem to a non-interacting single electron problem. $\varphi_i(\boldsymbol{r})$ represents the *i*th orbital (wavefunction) for the non-interacting electrons and ε_i is the corresponding eigenvalue associated with the energy of the *i*th orbital.

Each term in Eq. 2.9 can be computed exactly, except for the exchange-correlation functional. While it has been proven that an exact functional exists to return the energy, this functional is unknown.[251] Over the past decades, there have been several attempts to approximate the exchange-correlation term, and research on this subject remains open.[252] This has resulted in the development of a large number of functionals, pejoratively referred to as the "functional zoo", with different functionals optimized for different tasks. As DFT is not variational, there is no certain way of knowing whether one functional returns a more accurate energy than another, leading to the necessity to benchmark DFT methods against experiment.[252,253]

Whilst there are no guarantees about the relative accuracies of the final energies, approximations to the exchange-correlation functional can be ranked in a hierarchy, commonly referred to as "Jacob's Ladder" (Figure 5).[254] Key approximations include (i) the *local-density approximation*, where exchange-correlation is described solely by the electron density at a given point in space (first rung), (ii) the generalized gradient approximation (GGA eg., PBE,[255] BLYP[256,257]), where exchange-correlation is dependent upon the gradient of the electron density, with respect to position (second rung), (iii) the meta-generalized gradient approximation (mGGA eg., TPSS[258]), where exchange-correlation is dependent on the second derivative of the density with respect to position (equivalent to the kinetic energy density –

third rung), (iv) the fourth rung is defined by inclusion of 'exact' Hartree–Fock exchange (hybrid, e.g., PBE0,[259] B3LYP,[260,261] M06-2X[262]), and (v) the fifth rung reached by dependence on virtual orbitals yielding a double hybrid functional, which usually comes with a significant computational cost.

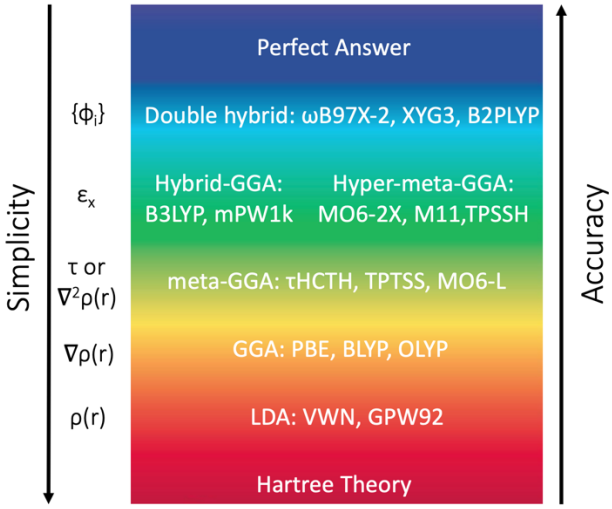


Figure 5: The hierarchy of exchange-correlation functionals represented by the rungs of Jacob's ladder. Adapted from Paton *et al.*[263]

Despite their continued evolution, recent developments have focused on achieving accurate energetics on benchmark sets, rather than reducing density errors.[252] This reduces the generalizability of these functionals and makes 'old' functionals e.g., PBE0 or B3LYP still a good choice without a benchmarking study.[264,265]

Another limitation of DFT is the description of dispersion, the inclusion of which is essential in weakly-bound systems as, for example, at PBE and B3LYP, the benzene dimer is purely repulsive.[266] Although in principle the exchange correlation functional should account for dispersion, it is a non-local property and thus expensive to include into the functional.[267] Instead, semi-empirical corrections (e.g., D3)[268] are generally used to account for the absent dispersion contribution. A damping function may be introduced, which has been found to increase accuracy on benchmark sets.[252] The latest generally implemented dispersion method is the damped D3 (D3BJ)[269,270] with the most recent D4 method not yet widely available.

With such a wide variety of functionals available a pragmatic approach is needed. Accepting an implicit error on reaction barriers and energies of ~ 5 kcal mol^{-1} and ~ 1 kcal mol^{-1} on double differences for similar systems, the transferability of the functional is more important than the absolute accuracy.[271,272] For that reason, the non-empirical PBE functional and its hybrid variant PBE0 in combination with D3BJ semi-empirical dispersion are used throughout

this Thesis. This method has been shown to be both transferable, accurate and is reasonably efficient.[252]

Despite the fact that the ever-increasing computer power has allowed for the development of more accurate and affordable electronic structure methods and software, DFT can still be time-consuming.[273–275]

2.1.3. Basis sets

DFT methods use a basis set to mathematically describe AO. Basis sets consist of basis functions located on each atom and they are typically comprised of Gaussian-type basis functions (GTO), with multiple GTOs required to approximate the solution to the Schrödinger equation for the hydrogen atom. Although Slater-type basis sets (STO) provide a more accurate description of the short- and long-range physical behavior of a hydrogenic orbital, their solution is more computationally demanding, requiring a solution of an infinite series to give the correct answer.[276] Hence, Gaussian-type basis sets are more commonly used to evaluate the electronic structure of organic molecules, with split-valence basis sets providing not only additional efficiency to the speed of calculations, but a better description of valence electron orbitals, which are key for determining the reactivity of a given system.

Basis set size has a large effect on the cost and accuracy of a computation. A small basis set, that has one basis function per AO (single-ζ basis set) does not provide the freedom required, to the electrons, to reach a realistic solution, which leads to loss of accuracy. Double-ζ and triple-ζ basis sets have double and triple the number of basis functions respectively and are both in common use. DFT has the advantage of fast basis set convergence,[277] with acceptable geometries using a double-ζ quality basis and energies at triple-ζ.[278] Once again, there is a balance between acceptable accuracy and cost in each calculation. Therefore, the split valence def2-SVP basis set will be used for geometry optimizations and the def2-TZVP for single point energy evaluations. These choices have been validated previously.[279]

2.1.4. Resolution of identity (RI) approximation

The basis set size required for calculations of synthetically interesting molecules (10s-100s atoms) makes the formal scaling of DFT too severe.[280] In electronic structure calculations, the computation of four-center two-electron repulsion integrals is the most computationally expensive step. These integrals describe the Coulombic interaction between one pair of

electrons, each electron being located in a different orbital centered on a different atom. The cost arises because there are many possible combinations of these two electrons, and each of these combinations must be evaluated separately. To reduce this computational cost, a resolution of the identity (RI) approximation is used. This expansion effectively approximates the four-center integrals in an auxiliary basis as two-center (Coulomb) and three-center (exchange) integrals, which can be evaluated much more efficiently. This reduces the overall computational cost of the electronic structure calculation while still producing accurate results.[281] Multiple algorithms for implementing RI are available, with the chain-of-spheres exchange (RI-JCOSX) for exact exchange terms being particularly efficient for large molecules.[282] All calculations in this Thesis were carried out with the ORCA electronic structure package (*v.4.1.1*),[283] and RI-JCOSX approximation was employed as standard, which uses separate auxiliary basis sets for the Coulomb, exchange and correlation integrals.[282]

2.1.5. Semi-empirical methods

In the past researchers have come up with creative schemes to accelerate ab initio modelling of both the thermodynamic and kinetic aspect of materials. Semi-empirical methods make explicit and systematic use of experimental (empirical) data in their elaboration, while maintaining the quantum mechanical framework. An example of semi-empirical methods based on DFT is Tight Binding (TB) DFT, which makes use of a Taylor expansion in the density and a minimal basis set to arrive at a general expression (Eq. 2.10):

$$E_{TB-DFT} = \sum_{a} f_a \sum_{i,j} c_i^a c_j^a H_{ij} + \sum_{A>B} \gamma_{AB}(R_{AB}) \Delta_{qA} \Delta_{qB} + \sum_{A>B} V_{rep}(R_{AB}), \quad (2.10)$$

where f_a is the occupation number of MO a, c_i is an AO coefficient, H is the parametrized Hamiltonian matrix, γ_{AB} represents parameters associated with the TB model, R_{AB} represents internuclear distances, Δq are partial charges and V_{rep} is a repulsive function depending on internuclear distances. This simple functional form for the energy is fast to evaluate but is highly parametrized, thus can be non-transferable.[284]

Recently, Grimme and co-workers have developed a variant of DFTB denoted GFN2-xTB,[285] parametrized on elements up to radon, which has been successful in a variety of contexts including molecular geometries and non-covalent interactions. The GFN2-xTB method uses a minimal basis with polarization without pair-specific parameters, making it more applicable than standard DFTB.[286]

Semi-empirical methods, sacrifice accuracy for speed and the ability to easily perform calculations with hundreds of atoms.[287] These attempts however, are not entirely satisfactory. For example, employing a minimal basis set generally leads to exaggerated anion instability and limited polarisation.[288] As such, quantitative predictions from derived energy differences are generally not possible.

2.1.6. Entropic contributions

Quantum mechanical calculations provide information on the electronic structure at a microscopic level, excluding temperature and pressure effects. However, this data alone is insufficient for studying real-world applications, such as catalysis and selectivity, which involve computing macroscopic properties such as Gibbs free energy, using concepts from thermodynamics and statistical mechanics. Bridging the gap between microscopic and macroscopic properties requires incorporating the effect of temperature and pressure into the results from DFT calculations.

The key quantity for studying macroscopy properties under constant temperature and pressure is the Gibbs free energy *G(T,p)*, comprising enthalpic (H) and entropic contributions(*TS*) (Eq. 2.11):

$$G(T,p) = H - TS. \qquad (2.11)$$

Here, $H = U + k_B T$; where U includes electronic (E_{el}), zero-point (E_{ZPE}), vibrational (E_{vib}), rotational (E_{rot}), and translational (E_{trans}) energies. *S* comprises contributions from translational (S_t), rotational (S_r), vibrational (S_v), and electronic (S_e) degrees of freedom.

Assuming the systems is in thermodynamic equilibrium, a bulk or homogeneous system (exhibiting ideal gas behaviour[289]) can be divided into subsystems, each treated separately within quantum mechanics. Entropy can then be calculated via the partition function *Q* of the system using statistical thermodynamics, so that:

$$S = R + R\ln(Q) + RT\left(\frac{\partial \ln Q}{\partial T}\right)_V$$

$$= R\left(\ln\left((q_t\, q_e q_r\, q_v)e\right) + T\left(\frac{\partial \ln Q}{\partial T}\right)_V\right). \quad (2.12)$$

Within the ideal gas approximation, the molecular partition function for a rigid molecule is approximated using analytic expressions derived from eigenvalues of a particle in a box, a rigid rotor, and a harmonic oscillator for translational, rotational, and vibrational contributions, respectively.[290] This approach yields the total entropy contribution from the translational partition function (Eq. 2.13), the total entropy contribution from the rotational partition function (Eq. 2.14), and the total entropy contribution from the vibrational partition function (Eq. 2.15).[290] Where R is the gas constant and T is the temperature in Kelvin.

$$S_t = R\,(\ln(q_t) + 1 + \frac{3}{2}), \quad (2.13)$$

$$S_r = R\,(\ln(q_r) + \frac{3}{2}), \quad (2.14)$$

$$S_v = R\,(\ln(q_v) + T(\frac{\partial \ln(q_v)}{\partial T})_V). \quad (2.15)$$

It's essential to acknowledge the approximations made in this process, firstly all the equations assume non-interacting particles and therefore apply only to an ideal gas. Secondly, for the electronic contributions, it is assumed that the first and higher excited states are entirely inaccessible. This approximation is generally not troublesome, but can introduce some error for systems with low lying electronic excited states.[289] Finally, low-frequency modes, which are prevalent in molecules with multiple degrees of freedom, are not well approximated by the harmonic oscillator model.[290] The partition function is crucial for computing entropy, but approximating it poses challenges, especially with increasing degrees of freedom.[290]

2.1.7. Implicit solvent models

Solvent effects play a crucial role in solution-phase reactions, they can be described either explicitly or implicitly. Explicit solvation places the solute into a pool of discrete solvent molecules, and therefore explicit solute-solvent interactions such as hydrogen bonds are present. Explicit solvation gives the closest match to the experimental system; however, it can become prohibitively expensive for large systems. Moreover, the configuration of solvent molecules must be well-sampled to include multiple configurations of similar energy that may be partially populated at a given temperature. There is the potential for an enormous number of solvent configurations due to the lack of strong ordering interactions, and errors may also

arise in the calculation of entropic contributions due to an abundance of low-energy vibrational modes.

Implicit modeling of solvent effects is an efficient alternative which constructs a charged, solvent-accessible cavity around the solute. It has become a more commonplace and less computationally demanding alternative for mimicking the effects of solvent on the reactive components. Although this technique cannot account for hydrogen-bonding and other explicit solvent-solute interactions, it provides several other advantages, including the ability for direct optimization of charge distribution and other electronic properties of the system for a given solvent environment. For this latter reason, the SMD implicit solvent model was applied throughout our own computational investigations to account for solvation of all relevant species.[276]

2.2. Machine learning algorithms

In general, when performing ML, we are interested in collecting data on observations and leverage this information to make predictions or estimate properties for future, unseen observations. This data, also known as predictors, features or descriptors, have attributes $\boldsymbol{X} = \begin{bmatrix} x_{1,1} & x_{1,2} \dots x_{1,p} \\ x_{2,1} & x_{2,2} \dots x_{n,p} \end{bmatrix}$ describing processes and their corresponding properties, $\boldsymbol{y} = [y_1, y_2, \dots, y_n]$. n and p are the total number of observations and the dimensionality of the input attributes, respectively. From this data a predictive model can be established, by mapping $\boldsymbol{X}$ and $\boldsymbol{y}$ without the need to understand the underlying relationship. To construct an ML model, one needs to: (i) acquire data that is accurate and curated, (ii) represent the data in a machine comprehensible manner, (iii) choose a learning algorithm, and (iv) validate and verify the developed model itself (Figure 6).

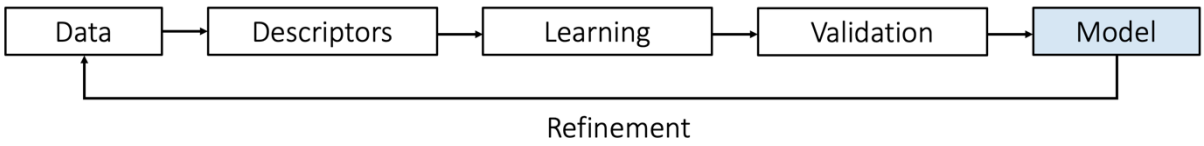


Figure 6: Key steps in constructing a ML model. After the data acquisition, descriptors need to be generated and the ML algorithm needs to be chosen. Then the model is validated. This process may be repeated if new data arise or if the descriptors or ML algorithm fail to generate a robust model.

2.2.1. Data set generation; chemical and structural descriptors

Data collection

The process of acquiring data and curating it for an ML model is a crucial step in developing accurate and effective models. The first step in acquiring data is to identify and collect relevant data sources that are representative of the problem being addressed. Data can be collected from online databases or through manual input. Once the data is collected, it needs to be cleaned and pre-processed to remove any irrelevant or noisy information. This can involve removing duplicates, filling in missing values, and correcting any errors. Before continuing with the ML and any preprocessing step, the dataset must be split into training and test sets. The training set is used to train the ML algorithm, and in some cases, it can be further divided into a validation set to fine-tune the model. Finally, the test set is reserved for evaluating the performance of the trained ML model (Figure 7).[2]

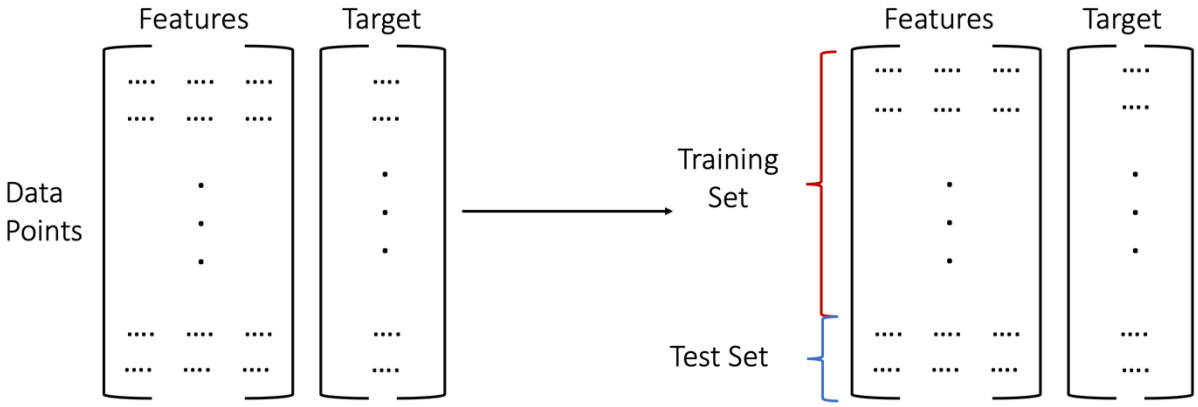


Figure 7: A dataset represented in a tabular format. The dataset must be split into training and test sets, where the training set is used to train the ML algorithm and the test set to evaluate it.

Feature generation

An important challenge during the construction of a model is the selection of features to describe the dataset under study. The features must be relevant to the nature of the data and the prediction task, otherwise, a relationship between the data and the output cannot be formed. Moreover, training ML algorithms on datasets with redundant features is computationally inefficient. The feature selection techniques can be categorized into three families: the filter-based, the wrapper-based, and the embedded methods.

Filter methods select features independently of the ML model being used. These methods evaluate the importance of each feature based on statistical measures such as correlation,

mutual information, and variance. Features are then ranked according to their importance, and a subset of the most relevant features is selected. Examples of filter methods include correlation-based feature selection, chi-squared feature selection, and variance thresholding.[291]

Wrapper methods select features by training an ML model and evaluating the performance of the model with different subsets of features. These methods can be computationally expensive since they involve training a model multiple times but can often lead to better feature subsets than filter methods. Examples of wrapper methods include recursive feature elimination and sequential feature selection. It is worth noting that here the feature subsets are biased towards the ML algorithm used.[292]

Finally, the embedded methods perform feature selection during the ML algorithm learning process, therefore the computational complexity is lower than in the wrapper-based methods. The embedded method is applicable in tree-based ML algorithms and ML algorithms with regularization terms, which compute how much each feature contributes to the training set performance.[291] In addition to these main categories, other dimensionality reduction techniques include PCA and *independent component analysis*, which involve transforming the data to a lower dimensional space and selecting the most relevant components.

In chemistry, the choice of the descriptors is driven by the underlying relation being explored. Nowadays, a broad range of chemical features are available, which can be categorized in 5 classes (Figure 8).[293,294]

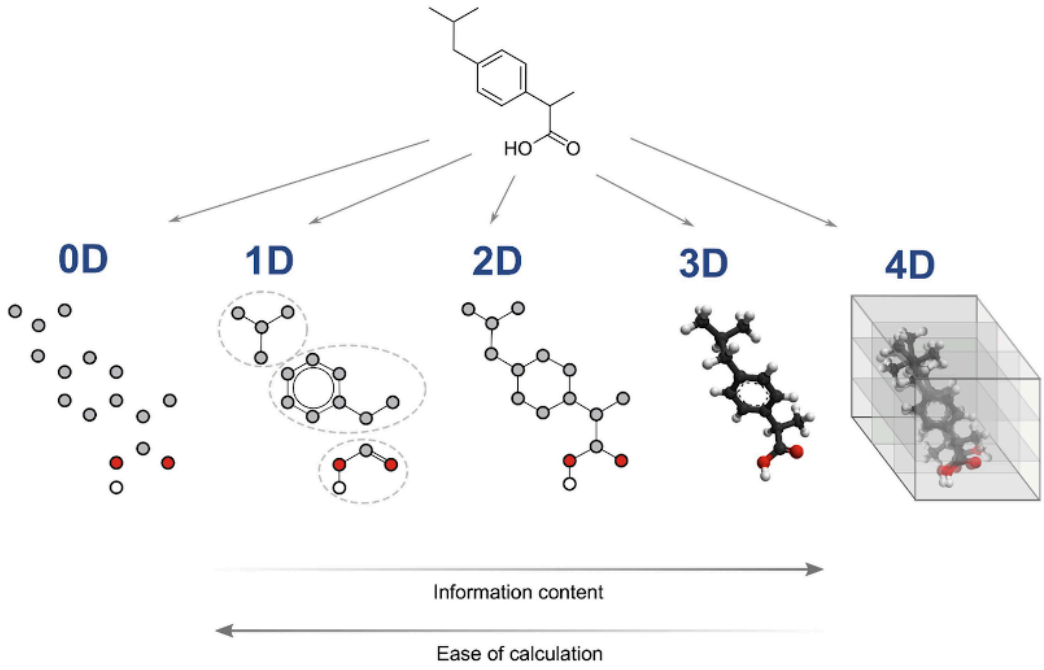


Figure 8: Representation of the five classes of theoretical descriptors and the relationship between their dimensionality, the information they provide and the ease of calculation. Figure from Consonni *et al.*[293]

0D descriptors are the simplest molecular representation which is the chemical formula, specifying the chemical elements and their occurrence in a molecule. They do not provide any information about the molecular structure or the connectivity of atoms. Some examples of 0D descriptors are atom counts, molecular weight, atomic vdW volumes.[291]

1D descriptors refer to molecular descriptors that can be calculated from a set of substructures such as functional groups, they are usually generated from InChI, SMILES or SELFIES. The most common 1D descriptors are fingerprints.[291]

2D descriptors refer to descriptors that provide information on molecular topology based on the graph representation of the molecules (nodes/vertexes are the atoms and edges are the bonds) and structural information that can be computed from the 2D structure (the number of benzene rings, the number of hydrogen bond donors). This category of descriptors is sensitive to structural features of the molecule (size, shape, and symmetry). 0D, 1D & 2D descriptors are easily obtained, however they show a low information content.[291]

3D descriptors include all geometrical descriptors that provide information about the spatial coordinates of atoms in a molecule. From 3D structures and electronic structure calculations (at various levels of theory), different physical chemical descriptors arise, such as partial atomic charges, electrostatic potentials, orbital energies, ionization energies, electron affinities, and bond orders. However, because of their complexity, the cost/benefit of using 3D descriptors is case-dependent and must be carefully evaluated.[295] Smooth Overlap of Atomic Positions (SOAP)[296] descriptors have emerged as an alternative. They are based on a representation of the local environment around each atom in a molecule. This is accomplished by dividing the space around each atom into a series of overlapping atomic regions, and then computing a set of basis functions that capture the distribution of nearby atoms within each region. Other well-known 3D descriptors are the 3D-MoRSE (Molecular Representation of Structures based on electronic diffraction) descriptors.[297]

The 4D descriptors are also called *grid-based descriptors*. These descriptors, in addition to the molecular geometry, introduce a fourth dimension. This new dimension usually characterizes the interactions between the molecule(s) and the active site(s) of a receptor or the multiple conformational states of the molecule(s), such an example is the CoMFA (Comparative Molecular Field Analysis).[298] An advantage of the 4D descriptors is that they provide more information than the other descriptors, however, they are not easy to obtain because of their higher complexity.

The descriptor categories listed here are not necessarily distinct, many hybrid descriptors have been generated that combine one or more categories. For example, Mold2[299] calculates 779 1D and 2D descriptors, the Mordred calculator uses over 1800 1D, 2D and 3D descriptors,[300] and the PaDEL-Descriptor generates 797 descriptors (663 1D and 2D descriptors, 134 3D descriptors) and 10 types of fingerprints.[301] Another examples are the charge indices, described by Galvez *et al.*[302] which combine partial charge information and topological connectivity. Similarly the *charged partial surface area* descriptors combine surface area and partial charge information.[303]

In this Thesis we focus mostly on fingerprint descriptors, Mordred descriptors and physical chemical descriptors. In each chapter we describe in detail the features used and the process followed to extract them.

Molecular fingerprints

Molecular fingerprints are high-dimensional vectors that encode the structure of a molecule into binary digit (bits) strings.[108] There are different types of molecular fingerprints, with the most widely used being circular fingerprints, also known as Morgan fingerprints.[304–306] Within this approach, the algorithm visits every atom of the molecule, obtains all possible paths through this atom with a specific radius, and hashes them into a bitmap (map of bits). The larger the radius, the bigger the encoded fragments. Moreover, the larger the bit number, the more discriminative the fingerprint can be (Figure 9).

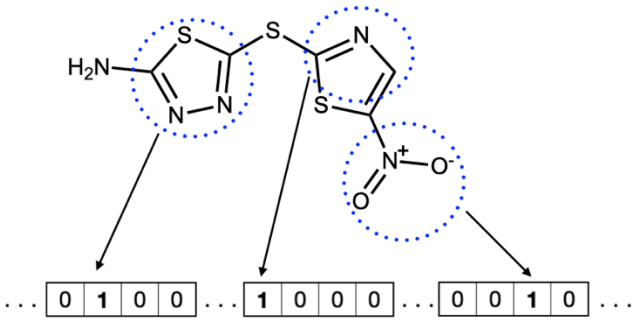


Figure 9: Representation of the calculation of Morgan fingerprints. Figure adapted from Xu *et al.*[307]

MACCS keys[308] are based on a predefined set of binary substructure keys that describe specific chemical features, such as functional groups and ring systems. Each substructure key is assigned a binary value of 1 or 0, depending on whether it is present or absent in the molecule. MACCS keys are generally less sensitive to small structural changes than circular fingerprints

but are more focused on capturing specific chemical functionalities. The MACCS keys are composed of 166 binary bits.

Extended-connectivity fingerprints (ECFP)[309] are another type of fingerprint that aim to capture molecular structure through the connectivity of atoms in the molecule. They are generated by recursively traversing the molecular graph and encoding the local environment of each atom in a fingerprint. ECFP fingerprints are size-invariant and can be generated with different radii to represent different levels of detail. They are particularly useful for capturing local structural motifs, such as hydrogen bonding patterns and ring systems.

Atom-pair fingerprints[310] are a type of fingerprint that encode the pairwise distances between pairs of atoms in the molecule. They are generated by defining a set of atom pairs and calculating their Euclidean distance in three-dimensional space. The resulting distances are then encoded as a binary vector. Atom-pair fingerprints are highly sensitive to the precise spatial arrangement of atoms in the molecule and are useful for capturing global structural features.

Topological fingerprints are based on the concept of encoding the molecular topology, or the way in which atoms are connected in the molecule. They are generated by defining a set of substructures, such as rings and chains, and encoding their presence or absence in the molecule as a binary vector. Topological fingerprints are generally less sensitive to small structural changes than other types of fingerprints but are useful for capturing global structural features.[311]

To compare the similarity of two molecules, we can use the bitmaps generated by the fingerprints and the *Tanimoto* metric (Eq. 2.16).[312] The resulting values range from 1 (identical molecules) to 0 (two molecules have nothing in common).

$$Tanimoto = \mathrm{BC} \,/\, (\mathrm{B1} + \mathrm{B2} - \mathrm{BC}). \qquad (2.16)$$

Here:

- B1: bits of fingerprint 1 (*F1*), the number of 1s in *F1*
- B2: bits of fingerprint 2 (*F2*), the number of 1s in *F2*
- BC: bits (*F1* and *F2*), the number of 1s in common between *F1* and *F2*. It is computed as the number of bits in common divided by the total number of bits.

Mordred descriptors calculator

In 2018, Moriwaki *et al*.[300] implemented the Mordred descriptors, a freely available software via GitHub (a list of the available descriptors can be found at reference[313]). Mordred have shown to be widely applicable as so far they have been used (on their own or in combination with other descriptors) to predict the properties of small molecules[314] or lead compounds and their binding affinities.[315,316] They have also been used in environmental studies for the prediction of fish bioconcentration factors,[317] and capture capacity predictions.[318]

From these descriptors, one should only be interested in those that have a notable correlation with the target value being investigated. A way to identify these descriptors is to set a Spearman or a Pearson correlation cutoff and further analyze these features for significance using a two-tailed p-test[300] over a random sample of permutations using the Spearman/Pearson correlation coefficient as the test statistic. With this approach, only features which have a significant p-value at 95% are considered. Following the feature generation, one-hot encoding for categorical features (features with specific increments such as counts) and scaling for continuous features should be applied.

Population analysis

Charges. Atomic charge is another property often used to rationalize structural and reactivity differences. Despite not being directly observable in experiments they are often invoked in discussions of bonding, as they are linked to electrostatic properties (e.g., attraction or repulsion).[319] Approximate charges can be computed by different methods. These methods can broadly be divided into two categories:[320]

1. Separation of the one particle density matrix in the Hilbert space (e.g., Mulliken, Löwdin, natural bond orbital analysis)
2. Separation of the electron density in real space. (e.g., Hirshfeld)

Mulliken population analysis distributes the electrons into atomic contributions. The contributions from all AO located on a given atom A are summed up to give the number of electrons associated with atom A. This sum is then subtracted from the nuclear charge of atom, Z_A (Eq. 2.17):

$$q_A = Z_A - \sum_{\mu \in A} (PS)_{\mu\mu}, \qquad (2.17)$$

where *P* and *S* are density and overlap matrices respectively (in the AO basis) and the sum is over orbitals centered on atom *A*.

A few common problems when population analysis is based on partitioning the wave function in terms of basis functions are the following:

1. The diagonal elements may be larger than two, implying that more than two electrons exist in an orbital, violating the Pauli principle.
2. The off-diagonal elements may become negative, implying negative number of electrons between two basis functions, which is physically impossible.
3. Dividing the off-diagonal contributions equally between the two orbitals is not a well-grounded technique as one might argue that the most electronegative orbital should receive most of the shared electrons.
4. A basis function centered on atom *A* may have a small exponent, describing the wave function far from atom *A*.
5. The dipole moments are not conserved.

Based on the above it is preferred to base the population analysis on properties of the wave function or electron density itself (and not on the basis set chosen). Hirshfeld charges are based on using atomic densities for partitioning the molecular electron density (Eq. 2.18):

$$Q_A = Z_A - \int \frac{\rho_A^{\text{atomic density}}(\boldsymbol{r})}{\sum_A^{M\text{atoms}} \rho_{\text{A}}^{\text{atomic density}}(\boldsymbol{r})} dr. \quad (2.18)$$

An ambiguity in the Hirshfeld method is the source of the atomic densities. Usually, spherically-averaged ground state densities are used for neutral atoms, but in some cases other valence configurations may be considered.[245,321] Based on the work of Saha *et al.*[322] the magnitude of the Hirshfeld charges is in general smaller than for Mulliken, and the Hirshfeld charges are also less basis set-dependent, as expected for a density-based population scheme. In this Thesis the Hirshfeld charges are used as they are implemented by ORCA (*v.4.1.1*).

Bond orders. The bond order (BO) between two atoms is calculated by taking the difference between the number of electrons in the bonding and anti-bonding orbitals of the atoms. The BO is then calculated as half of the difference. This method assumes that electrons are equally shared between the atoms in a bond (Eq. 2.19):

$$BO = \frac{N_{\text{bonding}} - N_{\text{antibonding}}}{2}. \quad (2.19)$$

In this Thesis BOs (between atoms of interest - explained in each chapter) are used as they are defined by ORCA (*v.4.1.1*) and its natural bond orbital analysis method.

Fukui - nucleophilicity descriptor

Nucleophilicity (electrophilicity) is the tendency to donate (or accept) electrons. Nucleophilicity/electrophilicity of an atom can be described by the Fukui functions which describe the density changes upon variation in the number of electrons. Fukui functions are conveniently computed using a finite difference approximation,[323,324] leading to Eq. 2.20 for nucleophilicity and Eq. 2.21 for electrophilicity:

$$f_r^{+}: q_i(n+1)(\boldsymbol{r}) - q_i(n)(\boldsymbol{r}), \quad (2.20)$$

$$f_r^{-}: q_i(n)(\boldsymbol{r}) - q_i(n-1)(\boldsymbol{r}), \quad (2.21)$$

where *n* is the actual target system, *n+1* is the anion (one more electron), and *n-1* is the cation (one less electron). Different studies have shown that nucleophilicity can be an important descriptor when it comes to constructing an ML model,[325] therefore we investigated if it could be a suitable descriptor for our systems. The analysis is presented in Appendix B1, and we concluded that the nucleophilicity/electrophilicity descriptor should not be considered for our ML models, as it showed poor correlation with our target values (*ee*) and it required time consuming calculations.

Frontier molecular orbital

Examining the MOs involved in a reaction can provide further insight. MOs can be obtained from a WFT or DFT calculation (delocalized MOs), with the highest occupied MO (HOMO) of one molecule acting as an electron donor and the lowest occupied MO (LUMO) of another molecule acting as an electron acceptor. The overlap between the HOMO and LUMO determines the strength of the interaction and therefore the reactivity of the molecules. Generally, molecules with a high-energy HOMO and a low-energy LUMO are more reactive than those with a low-energy HOMO and a high-energy LUMO. The HOMO-LUMO gap can be used to rationalize the activation enegry.[326,327] Natural Bond Orbital (NBO) theory can transform DFT-derived MOs into localized representations of lone pairs and bonds, similar to a Lewis representation of chemical structure.[328] However, the physical significance of MOs is debated[329,330] since there is no unique definition and the connection to experiment is complex,[322] at the same time, while canonical MOs can be understood quite clearly in small

molecules, they become much harder to interpret for larger molecules, due to delocalization and lack of high symmetry, so caution is necessary to ensure any chemical interpretations are not heavily dependent on the chosen method. The HOMO and LUMO are also included as descriptors in the following chapters.

Steric descriptors

Steric effects in chemical and biological systems have been a topic of debate within the scientific community for several years. Different parameter sets have been developed both experimentally and computationally, each with varying degrees of success. Among these are: (i) the A-values, which were derived from the study of mono-substituted cyclohexane rings by Winstein and Holness,[331] (ii) the interference values, which are experimentally determined steric parameters based on the heat-induced half-life of racemization in 2,2′-substituted biphenyl systems, [332,333] (iii) the molar refractivity, which is a steric parameter defined by the Lorentz-Lorenz equation and has been used in many early QSAR studies, but only describes the total steric volume and ignores molecular shape,[92] (iv) the Tolman cone angle, which may be limited to phosphine ligands,[334,335] and (v) the Taft parameter,[336,337] which has been subject to various redefinitions and manipulations. Notably, Charton found a correlation between Taft's experimentally measured rates and the calculated minimum vdW radii of each symmetrical substituent in esters.[86,338,339] Charton corrected the experimental values of non-symmetrical substituents to agree with the calculated vdW radii, creating a set of computational but experimentally rationalized parameters. Hansch validated Charton's parameters by extrapolating Charton's correlation to previously unmeasured substituents, resulting in agreement between predicted and measured values.[340]

In the 1970s Verloop and colleagues criticized the parameters used in the past to quantify steric effects for their inability to provide meaningful steric-based LFERs, possibly due to the complex nature of steric effects. In response, they developed the Sterimol program,[341,342] which calculates various dimensional properties for a single substituent based on Corey-Pauling-Koltun atomic models CPK.[343] Instead of grouping all spatial information into a single cumulative value, Sterimol created subparameters, each of which describes a different dimensional property of interest. The three Verloop parameters are composed of two width parameters (B1 and B5) and a length parameter (L) (Figure 10). The width subparameters are determined based on the substituent's profile when viewed down the axis of the primary bond. B1 represents the minimum profile width of the substituent from the primary bond axis, and

B5 represents the maximum width from the same axis. B1 is influenced by branching at the first carbon center and increases with increasing substitution. The length parameter is the total length of the substituent along the primary bond axis. In 2012, Sigman and co-workers introduced Sterimol parameters to construct a QSAR between sterics and enantioselectivity.[92] Since then, Sterimol parameters have also been applied in medicinal chemistry and asymmetric catalysis.[344,345]

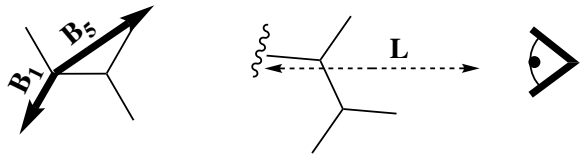


Figure 10: Illustration of the Sterimol parameters. B1 and B5 are the minimum and maximum widths of the group when viewed in profile looking down the primary axis and L is the total length along the same axis illustrated. Figure adapted from Sigman *et al*.[92]

In 2019 Paton and co-workers[346] developed the wSterimol, an automated software to extract sterimol descriptors. Discussion on how we have used wSterimol is provided in the Appendix B2.

NMR shifts

In NMR spectroscopy, a magnetic field is applied to align the nuclear spins, then excitation with radio waves takes place and the frequency of the emitted radio waves is measured providing information about the chemical environment of the atoms in the molecule. The chemical shift is a measure of this absorption or emission relative to a reference compound. In computational chemistry NMR shifts are typically calculated using WFT or DFT methods. The NMR shieldings are calculated using the Gauge-Including AO (GIAOs method, sometimes also referred to as London orbitals)[347–349] which involves adding a gauge term to the Hamiltonian of the molecule, accounting for the effect of the magnetic field.[350] The GIAO method is advantageous because it is relatively efficient and accurate for calculating NMR chemical shifts. However, it is important to note that the results obtained from the GIAO method are dependent on the choice of basis set and functional used in the calculation. Careful selection of these parameters is necessary to obtain reliable results.[351,352] The ORCA software uses the absolute shielding (isotropic chemical shieldings) where the value for a given nucleus is defined as the difference between the magnetic field experienced by that nucleus and the external magnetic field, expressed in parts per million (ppm). The calculated absolute shielding

values are then converted to NMR chemical shifts using a reference compound or experimental data.[353]

Normalization of data set

After discussing the available chemical descriptors suitable for ML, it is important to consider data preprocessing techniques to ensure optimal model performance. Feature engineering techniques such as scaling and normalizing may be required to transform the data into a more appropriate format and bring the descriptors' values within a consistent range, preventing any particular descriptor from dominating the model due to its larger scale. Similarly, normalizing the target values can help ensure equal importance is given to different ranges of the target variable, leading to more robust and reliable predictions.[354]

Scaling is a process of transforming the data so that it fits within a specific range. Normalizing is a process of transforming the data so that it has a mean of zero and a standard deviation of one. There are different forms of scaling and normalizing data, such as:

Min-max scaling, which scales the data to a specific range, usually between 0 and 1 (Eq. 2.22):

$$X_{\text{scaled}} = (X - X_{\text{min}}) / (X_{\text{max}} - X_{\text{min}}), \quad (2.22)$$

where X_{scaled} is the scaled value, X is the original value, X_{min} is the minimum value in the data, and X_{max} is the maximum value in the data.

Z-score normalization, which scales the data so that it has a mean of zero and a standard deviation of one (Eq. 2.23):

$$X_{\text{norm}} = (X - X_{\text{mean}}) / X_{\text{std}}, \quad (2.23)$$

where X_{norm} is the normalized value, X is the original value, X_{mean} is the mean value of the data, and X_{std} is the standard deviation of the data.

Log transformation, which transforms the data using a logarithmic function (Eq. 2.24):

$$X_{\text{log}} = \log(X), \quad (2.24)$$

where X_{log} is the transformed value and X is the original value.

Feature curation is an ongoing process that involves monitoring and updating the model as new information becomes available. It is important to ensure that the features remain relevant, and representative of the problem being addressed. This can involve regular re-training of the ML

model to ensure that it remains up-to-date and effective. At this point we would like to summarize some of the most important points associated with the choice of descriptors:

- Descriptors should correlate with the targets.
- Descriptors should generate dissimilar values for structurally different molecules.
- Not all descriptors are suitable for all sizes of molecules. Some descriptors are only useful when applied to small molecules, whereas other descriptors are defined specifically for large molecules such as polymers and proteins.[355]
- The amount of data required in a ML project is highly related to the complexity of the problem. For a small dataset, high-dimensional descriptors are not recommended. In fact, they increase the dimensionality of the problem and thus make the data sparser. The training becomes more complex and therefore more data are needed to obtain a model with a satisfying predictive performance. In general, the amount of data points must be four times larger than the number of parameters[356] otherwise, the model will be too flexible for the amount of training data.

2.2.2. Machine learning algorithms for supervised learning

The learning algorithm is another important ingredient in developing accurate ML models. Depending upon the scientific question that one intends to solve, the learning algorithms broadly fall under two classes: (i) supervised learning[3,357,358] (linear and non-linear regression, classification), and (ii) unsupervised learning[3] (cluster analysis, PCA, feature selection). The former is used when the desired output is known, and the latter is used for data with no historical labels, and where the goal is to explore the data and find patterns within (Figure 11). In this Thesis we focus on supervised learning, which can be further subdivided into regression and classification tasks.

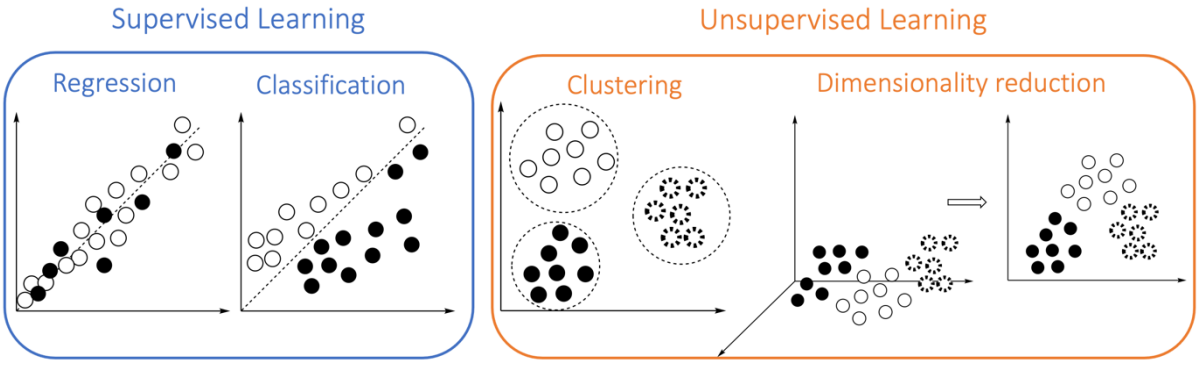


Figure 11: On the left hand-side in blue the two major supervised categories: regression and classification, and on the right hand-side in orange the two major unsupervised categories: clustering and dimensionality reduction.

Before proceeding further, it is important explain the concepts of over-fitting and underfitting in the context of ML. Over-fitting occurs when a model becomes overly complex and fits the training data extremely well, to the point where it fails to generalize effectively to new, unseen data. This can lead to poor performance and inaccurate predictions in real-world scenarios. On the other hand, underfitting refers to a situation where the model is too simplistic and fails to capture the underlying patterns in the data, resulting in suboptimal predictive performance.[359,360] Regularization methods, can introduce constraints or penalty terms to the learning process, discouraging excessive complexity and promoting a more generalized model.[6] Over the next paragraphs we will be referring to these terms.

Regression

Regression algorithms are applied to data with continuous outputs. In regression, the goal is to find the relationship between the input variables (also known as independent variables or predictors) and the output variable (also known as dependent variable or response).

The most basic and widely used regression algorithm is linear regression. It assumes that the relationship between the independent and dependent variables is linear, i.e., a straight line can be used to model the data. It tries to find the best-fit line that minimizes the sum of the squared errors between the predicted and actual values. This relationship can be represented by the equation (Eq. 2.25):

$$y = \beta x + \alpha, \quad (2.25)$$

where y is the dependent variable and x is the independent variable, β is the slope and α is y-the intercept.

In higher dimensions, where we have more than one independent variables x, the line is called a plane or a hyper-plane. This relationship can be represented by the equation (Eq. 2.26):

$$y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \alpha. \quad (2.26)$$

The goal is to estimate the values of β and α that minimize the sum of the squared errors between the predicted and actual values of y, also known as the residual sum of squares (RSS) (Eq. 2.27):

$$RSS = \sum_{i=1}^{n} (y_i - \alpha - \beta_i x_i)^2. \quad (2.27)$$

By taking the partial derivatives of the sum of squared differences with respect to β and α, set them equal to zero and solving for β and α we get:

$$\hat{\beta} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) / \sum_{i=1}^{n}(x_i - \bar{x})^2, \quad (2.28)$$

$$\hat{a} = \bar{y} - \hat{\beta}\bar{x}, \quad (2.29)$$

where $\bar{x}$ and $\bar{y}$ are the mean values of x and y, respectively. If $\hat{\beta} > 0$, then x and y have a positive relationship. If $\hat{\beta} < 0$, then they have a negative relationship. We note here that, the difference between β and $\hat{\beta}$ is that β represents the true population regression coefficients, while $\hat{\beta}$ is the estimated regression coefficients based on the sample data. In practice, we use $\hat{\beta}$ as an estimate of β, because we do not have access to the true population regression coefficients.

LASSO regression. To enhance the prediction accuracy of least squares and introduce constraints, regularization techniques can be introduced. The least absolute shrinkage and selection operator (LASSO)[6] penalizes large feature coefficients, which mathematically means it minimizes the least squares-penalty plus the regularization term (Eq. 2.30). LASSO estimates sparse coefficients and reduces the number of variables the model depends on, by shrinking the coefficients of some parameters toward zero. Variables with a coefficient equal to zero are excluded from the model. On the other hand, variables with non-zero coefficients are used in the model.

$$\hat{\beta}^{\text{lasso}} = \text{argmin}_{\beta} \sum_{i=1}^{n}\left(y_i - \sum_{j} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p}\left|\beta_j\right|. \quad (2.30)$$

Here, λ denotes the amount of shrinkage. If $\lambda = 0$ it is implied that all features are considered, and it is equivalent to the linear regression (only the residual sum of squares is considered). If $\lambda = \infty$ it is implied that no feature is considered. Therefore, as λ closes to infinity it eliminates more and more features. The bias increases with increase in λ, and variance increases with decrease in λ.

Ridge regression. An alternative to LASSO is Ridge regression.[6] The difference between the two is that Ridge regression adds a penalty equivalent to the square of the magnitude of coefficients (Eq. 2.31) This practically means that the shrinkage penalty will shrink all of the coefficients towards zero, but it will not set any of them exactly to zero (unless $\lambda = \infty$).

$$\hat{\beta}^{\text{ridge}} = \text{argmin}_{\beta} \sum_{i=1}^{n} \left(y_i - \sum_{j} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \ . \quad (2.31)$$

Elastic net regression. Elastic net regression[6] was created as a combination of both LASSO and Ridge regressions. In Eq. 2.32 there are now two λ terms. λ_1 is the value of penalty for the LASSO part of the regression and λ_2 is the value of penalty for the Ridge regression. The final penalty is a ratio of λ_1:λ_2. When setting the ratio = 0 it acts as a Ridge regression, and when the ratio = 1 it acts as a LASSO regression. Any value between 0 and 1 is a combination of Ridge and LASSO regression (Figure 12).

$$\hat{\beta}^{\text{elnet}} = \text{argmin}_{\beta} \sum_{i=1}^{n} \left(y_i - \sum_{j} x_{ij}\beta_j \right)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \ . \quad (2.32)$$

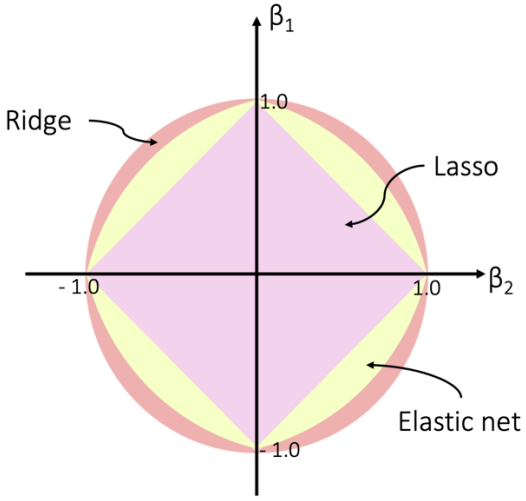


Figure 12: Graphical representation of geometrical analysis of LASSO (salmon), Ridge regression (pink) and elastic net (yellow). Figure adapted from Dehmer *et al.*[361]

In the cases where the data do not follow a linear relationship, polynomial regression should be considered. It allows for non-linear relationships between the independent and dependent variables, by using polynomial functions of degree greater than one to model the data.

Classification

Classification algorithms are used when the data outputs are discrete, i.e., the data are separated into classes, and they automatically learn to map the data to specific classes. They can be categorized into linear models, DT, neighbor-based, generative models, NN, and others.

Linear models generate a formula that optimally separates the classes. They create a decision boundary based on the linear combination of data features (Eq.2.33):

$$y = f\left(\sum_j w_j\ x_j\right), \quad (2.33)$$

where x_j is the value of feature j, w_j is the weight of feature j, and $f(\cdot)$ is a threshold function that assigns the dot product $\underset{w}{\rightarrow} \cdot \underset{x}{\rightarrow}$ in a specific class, for example a sign function. The weight vector $\underset{w}{\rightarrow}$ is optimized based on the training set. Optionally, another parameter b, called bias, is also summed but it is omitted herein for simplicity. For binary classification, the problem is a weight optimization task (Eq. 2.34):[362]

$$\min f(\boldsymbol{w}) \equiv\ r(\boldsymbol{w}) +\ C\ \sum_i L(\boldsymbol{w}^T \boldsymbol{x}_i, y_i), \quad (2.34)$$

where $\boldsymbol{x}_i$ is the feature vector of sample i, y_i the class of sample i, $\boldsymbol{w}$ the weight vector, $L(\cdot)$ is a loss function that measures the deviation/error between the predicted output from the true label, $r(\cdot)$ is a regularization term, and C is a constant that balances the regularization term and the sum of the losses. Depending on the loss function and the optimization algorithm that finds the best weight values, different classifiers occur. If the loss function is the logistic function (Eq. 2.35):

$$L(\boldsymbol{w}^T \boldsymbol{x}, y) =\ \log\left(1\ +\ e^{-y\boldsymbol{w}^T \boldsymbol{x}}\right), \quad (2.35)$$

then the logistic regression (LR) algorithm results,[363] and if the loss function is the hinge loss (Eq. 2.36):

$$L(\boldsymbol{w}^T \boldsymbol{x}, y) =\ \max(0,1\ -\ y\boldsymbol{w}^T \boldsymbol{x}), \quad (2.36)$$

then the SVM algorithm results.[364] For the LR and SVMs classifiers (Figure 13) various optimization algorithms can be applied,[362,365,366] including the stochastic gradient descent (SGD). SGD enables online learning, allowing the model to be trained with individual samples or mini-batches of samples at each step. This feature enables continuous learning when new samples become available. Several variations of the abovementioned classifiers have emerged, for example, non-linear SVMs that apply non-linear kernel functions, and LR and SVMs with SGD.[367–369]

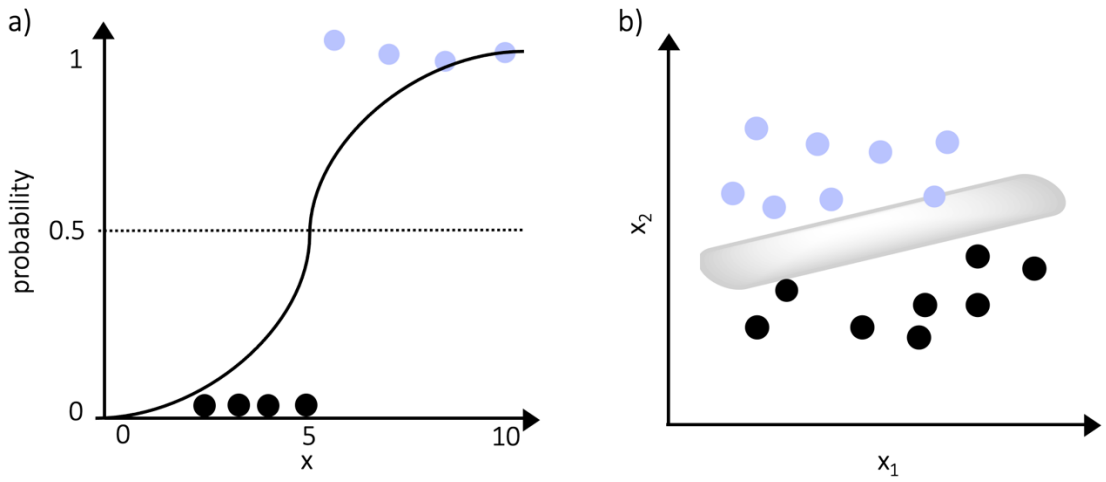


Figure 13: Graphical representation of a) logistic regression where the decision boundary is 0.5 and the data fall into class 0 or class 1 and b) support vector machines, where the two classes are separated by a hyperplane shown here in a grey.

ML classifiers can also be generative, meaning that they try to learn the probability distribution of the input data. From a statistical point of view, instead of calculating the joint distribution $p(y, \boldsymbol{x})$ from Eq. 2.37:

$$p(y, \boldsymbol{x}) = P(y|\boldsymbol{x})p(\boldsymbol{x}), \quad (2.37)$$

in generative learning the joint distribution $p(y, \boldsymbol{x})$ is calculated from Eq. 2.38:

$$p(y, \boldsymbol{x}) = p(\boldsymbol{x}|y)P(y), \quad (2.38)$$

where $\boldsymbol{x}$ is the feature vector of a sample, y its class, $P(y)$ is the class probability and $p(\boldsymbol{x}|y)$ is the conditional distribution of the input given the class label. In Bayesian classification, the Bayes' theorem is applied (Eq. 2.39):

$$P(A, B) = P(A|B)P(B) = P(B|A)P(A), \quad (2.39)$$

where $P(A|B)$ is the probability of event A occurring given B, $P(B|A)$ is the opposite, and $P(A)$ and $P(B)$ are the probabilities of A and B, respectively. Resulting in the *a-posteriori* probability (Eq. 2.40):

$$P(y|\boldsymbol{x}) = \frac{p(y, \boldsymbol{x})}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|y)\, P(y)}{p(\boldsymbol{x})}. \quad (2.40)$$

Then, the error is minimized by partitioning the feature space so that an unknown pattern, represented by the feature vector $\boldsymbol{x}$, is assigned to class *i.* If (Eq. 2.41),

$$P(y = i|\boldsymbol{x}) > P(y = j|\boldsymbol{x}) \; \forall \; i \neq j, \quad (2.41)$$

the LDA classifier (Linear Discriminant Analysis)[370] assumes that the conditional density $p(\boldsymbol{x}|y)$ follows the multivariate Gaussian distribution and share the same covariance matrix $\boldsymbol{\Sigma}$ (Eq. 2.42):

$$P(\boldsymbol{x}|y) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_k})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_k})\right), \quad (2.42)$$

where d is the number of features and $\boldsymbol{\mu_k}$ is the mean vector of class k. The generalization of LDA, where the covariance matrices differ, is called quadratic discriminant analysis.[371]

The conditional density $p(\boldsymbol{x}|y)$ cannot be solved easily and the previous ML algorithms suffer from the curse of dimensionality.[5] This issue is addressed in the naive Bayes algorithm which assumes that all features *x* are mutually independent given the class label, resulting in Eq. 2.43:

$$p(\boldsymbol{x}|y) = \prod_j p(x_j|y), \quad (2.43)$$

and in Eq. 2.44:

$$P(y|\boldsymbol{x}) = \frac{\prod_j p(x_j|y)P(y)}{p(\boldsymbol{x})}. \quad (2.44)$$

The $p(\boldsymbol{x})$ does not depend on y and can be omitted and the resulting formula can be rewritten and solved using the maximum *a*-post*eriori* rule (Eq. 2.45):

$$\hat{y} = \text{argmax}_k\, P(y = k|\boldsymbol{x}) = \text{argmax}_k \prod_j p(x_j|y)P(y) \quad (2.45)$$

The decision tree (DT) classifier is a set of if/else decision rules that continuously splits the training set according to a criterion that maximizes the separation of the data.[372] The result is a tree-like structure where branches represent the data split and leaves represent the outcomes. A DT is built by recursively partitioning the data into subsets based on the values of one or more input features. The algorithm splits the data into smaller subsets based on the feature that provides the most information gain. Information gain is the difference between the impurity of the parent node and the sum of the child node impurities. The lower the impurity of the child nodes, the larger the information gain. This process is repeated until the subsets are pure, or a stopping criterion is met (Figure 14a). A DT with a larger maximum depth or a smaller minimum number of samples in a leaf node will have more leaf nodes, which may lead to over-fitting of the training data. On the other hand, a DT with a smaller maximum depth or a larger minimum number of samples in a leaf node may underfit the training data.

Finally, neighbor-based algorithms like *k*-nearest neighbors (kNN) classify a sample based on the majority vote of its *k* nearest neighbors (Figure 14b).[373] A variation of kNN is the radius nearest neighbors algorithm, which classifies a sample based on the majority vote of the neighbors within a fixed radius. A different approach is followed by the nearest centroid algorithm, where each class is represented by the centroid of its samples. Then, the new sample is classified based on the class of the nearest centroid.

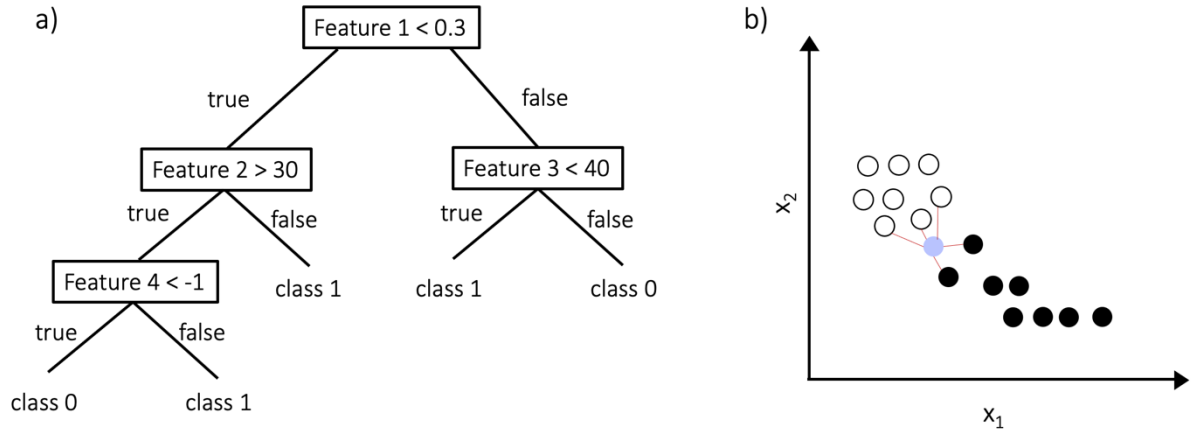


Figure 14: a) Example of decision tree with four features in a binary classification task, b) graphical representation of kNN algorithm the new data point (in purple) is classified according to its neighbors. In red the distances between the new point and its k=5 nearest neighbors.

Here, it is worth mentioning certain algorithms that were originally developed for classification but have found application in regression as well. One such algorithm is the support vector regressor (SVR) algorithm, which utilizes the SVM to map the input data into a higher dimensional space. The basic idea behind SVR is to find the best fit line, which is the hyperplane that has the maximum number of points. Unlike other regression models that try to minimize the error between the real and predicted value, the SVR tries to fit the best line within a threshold value. The threshold value is the distance between the hyperplane and boundary line.

Bayesian regression is another regression technique that leverages the Bayesian inference for estimating regression coefficients and make predictions. This method involves specifying *a-priori* distribution for the regression coefficients, and then updating this distribution based on the observed data to obtain a posterior distribution. DTs have also been adapted for regression. DT regression involves splitting the feature space based on decision rules to create a tree-like structure. The predicted value in regression is obtained by traversing the DT based on the features of the new data point. Finally, K-nearest neighbor regression makes predictions based on the k nearest neighbors to the new data point. The predicted value is the average of the dependent variable values for the k nearest neighbors.

Limitations of shallow learners

The aforementioned regressors and classifiers (shallow learners) are powerful tools for predictive modeling, nonetheless, they pose their own challenges. Linear and logistic regression, excel in cases where the data exhibits linearity or simplicity. However, they may encounter difficulties when faced with complex and high-dimensional datasets, or non-linear relationships.[6,363]

A drawback of SVMs is their computational complexity, especially when dealing with large datasets. As the number of samples increases, the training time and memory requirements of the algorithm can become significant. Additionally, SVMs may struggle when faced with datasets that have a large number of features or when the classes are overlapping or inseparable, making it challenging to find an optimal hyperplane to separate the data points accurately. SVMs also lack inherent interpretability, as the resulting models often provide limited insights into the underlying relationships within the data.[364]

DTs are susceptible to over-fitting, especially when the model becomes overly complex and closely fits the training data. DTs are also sensitive to small variations in the training data. A small change in the training dataset can potentially lead to a significantly different DT. This instability can make DT less robust and prone to high variance in the predictions.[372]

kNN struggles with datasets that have imbalanced class distributions, since it makes predictions based on the majority class among the *k* nearest neighbors, imbalanced data can lead to biased predictions favoring the majority class. Bayesian classification methods may also face challenges when dealing with imbalanced datasets or when prior assumptions are not well-informed.[373]

Class imbalance problem

When the size of one class outnumbers the size of the other, ML algorithms tend to under-predict the infrequent class. To tackle this issue, the classes must be transformed into balanced classes, either by over-sampling the minority class or by under-sampling the majority class. An approach to perform over-sampling is to generate synthetic samples based on the feature values of the minority class samples until both classes consist of an equal number of samples. The most common algorithm for generating synthetic samples is the Synthetic Minority Over-sampling Technique (SMOTE) technique, which synthesizes artificial new minority samples between existing minority samples (Figure 15).[374]

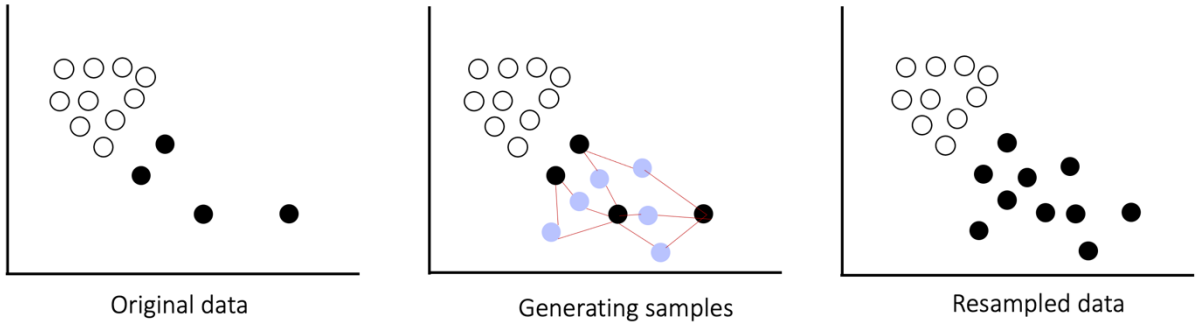


Figure 15: Illustration of the SMOTE algorithm. Starting from the original data, it generates synthetic samples (purple) for the minority class (black). This is done by generating a synthetic point between two original points. The points are "connected" with red.[374]

Another algorithm for over-sampling is the *adaptive synthetic sampling* algorithm, which is similar to SMOTE, but attempts to infer which points in the minority class would be the most difficult for a model to learn and attempts to place a higher ratio of synthetic data close to these points.[375]

To under-sample the majority class one can randomly delete samples until both classes have an equal number of samples. A more sophisticated selection technique is the *condensed nearest neighbor* method, which iteratively uses the k nearest neighbor rule to decide if a majority sample should be removed or not.[376] If the majority of the nearest neighbors are in the minority class, then the majority sample is kept, otherwise, the majority sample is discarded. Another selection technique is the *instance hardness threshold*, where a ML algorithm is trained on the training set and removes the samples with the lowest probabilities.[377]

An alternative approach is to use a combination of over- and under-sampling techniques. Because over-sampling using synthetic sample generation algorithms may lead to the generation of noisy samples, an under-sampling method may be used afterwards to clean the training set.

Finally, a different approach for dealing with the class imbalance problem in classification is to use penalization with class weights. In this approach, when a sample is misclassified, a weighted cost is imposed on the model, biasing the model to emphasize the minority class. In the scikit-learn Python package[378] the weights can be automatically assigned according to Eq. 2.46:

$$w_j = \frac{n\_samples}{n_{\_classes} * n\_samples_j}, \qquad (2.46)$$

where w is the weight of class j, n_samples is the total number of samples of the j training set, n_classes is the total number of classes, and $n\_samples_j$ is the total number of samples in class j in the training set.

Graph neural networks

A special case of ML is deep learning or artificial neural networks, or simply referred to as NNs, which are inspired by the brain. In the neuron, information comes from the dendrites, this information is summed in the cell body, and the response is output to the axon and to other neurons, via an electrical signal (Figure 16a). The simplest NN is called a perceptron and follows the same principles as the biological neuron (Figure 16b). The relationship between the input vector $\boldsymbol{i}$ and output o of a single perceptron is given by Eq. 2.47. The inputs are weighted (weight vector: $\boldsymbol{w}$), summed, and then an activation function is applied leading to the output. An activation function is a mathematical equation, for example the sigmoid function, that determines whether a node should be activated or not. If a node is activated, it will pass data to the nodes of the next layer. The activation function can be calculated by multiplying input and weight and adding a bias.

$$o = f\left(\sum_j w_j \cdot i_j + b\right) = f(\boldsymbol{wi} + b). \quad (2.47)$$

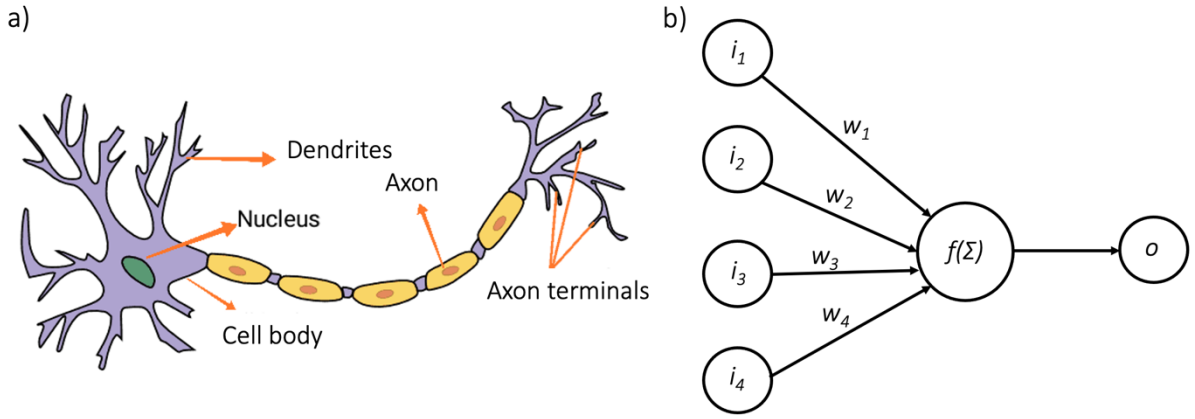


Figure 16: Analogy between (a) a biological neuron and (b) a perceptron i: inputs; w: weights; o: output; f(Σ): activation function. Figure adapted from Gemm *et al.* [379]

The combination of multiple perceptrons provides a multi-layer perceptron (MLP) (Figure 17a and Eq. 2.48).

$$\boldsymbol{y} = \boldsymbol{f}_2[\boldsymbol{W}_2 * \boldsymbol{f}_1(\boldsymbol{W}_1 * \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2]. \quad (2.48)$$

Where $\boldsymbol{y}$ is the model output vector; $\boldsymbol{x}$ is the model input vector; $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are the activation functions, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are the bias vectors, for layer 1 (hidden layer) and layer 2 (output layer), respectively. If the MLP consists of more than one hidden layers a deep learning MLP is produced (Figure 17b).

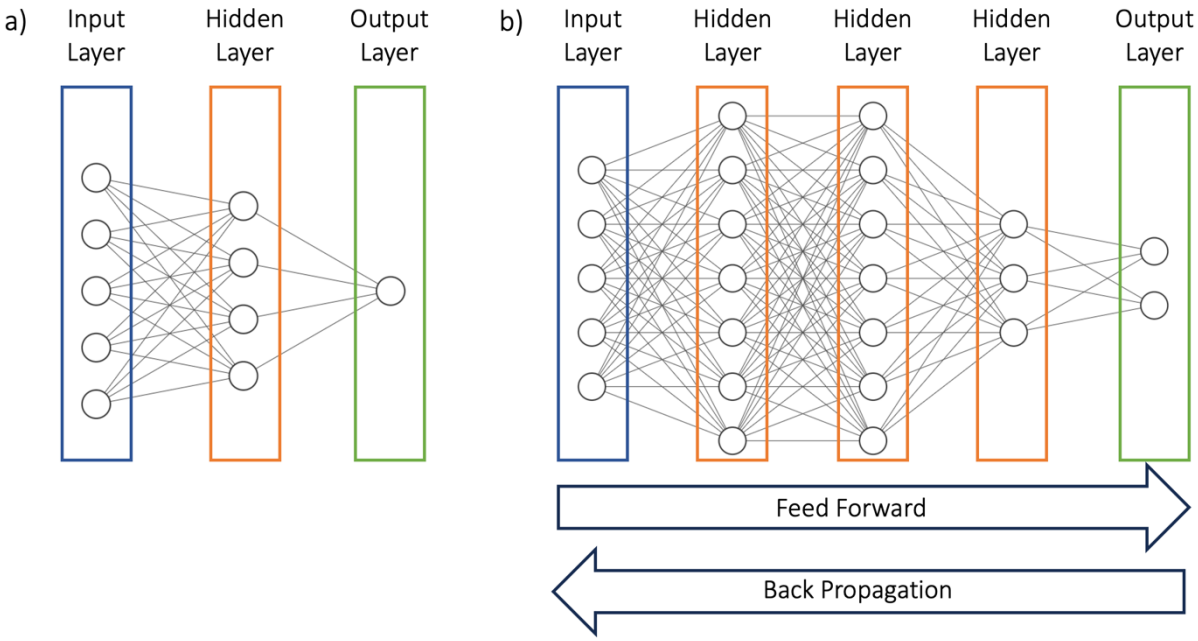


Figure 17: Comparison of (a) a shallow MLP with (b) a deep learning MLP showing the feed forward and back propagation techniques. Figure adapted from Gemm et al.[379]

MLPs belong to the category of feedforward algorithms, wherein inputs are combined with initial weights in a weighted sum, and subject to an activation function. This process is propagated through successive layers, to the output layer, with each layer feeding the next with the result of its computation, forming an internal representation of the data.

However, simply propagating the results to the output layer is not sufficient for effective weight adjustment to minimize the loss function. To train MLPs, the optimal weights must be found, typically through an iterative process called backpropagation. During backpropagation, the gradient of a loss function is computed, and in each iteration, the weights are adjusted to minimize this gradient. The loss function needs to be a differentiable convex function. The derivate of a function at *x* is the slope of the graph of the function at point *(x, f(x))*. The slope of a linear function is the rate at which it rises or falls, describing the direction of that function.

As an example, a loss function can be the mean squared error (MSE) (Eq. 2.49) and the gradient method to minimize the loss can be the gradient descent algorithm (Eq. 2.50).

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2, \quad (2.49)$$

where n is the number of predictions.

$$w = w - a\frac{dJ}{dw}, \quad (2.50)$$

where w are the weights, J is the loss function, in this case the MSE, and α is an adjustable parameter called learning rate, determining the step size at each iteration. When the learning rate is too big, the algorithm is taking big steps, risking stepping over the minima, similarly when the learning rate is too small, the algorithm is taking tiny steps, and therefore taking longer to reach the minima.

Here it is worth mentioning that the advantage of gradient decent lies in its simplicity, as it only requires the derivative of the loss function. However, it can be inefficient, especially when dealing with very small gradients. More advanced methods, such as Newtonian methods like BFGS (Broyden–Fletcher–Goldfarb–Shanno), offer improvements over gradient descent by taking adaptive steps based on the Hessian matrix, which contains second-order derivative information. Nevertheless, these methods are computationally expensive.[380] Adam (short for Adaptive Moment Estimation) is another optimization algorithm that adapts the learning rate for each parameter, allowing for different learning rates for each parameter rather than using a global learning rate like traditional gradient descent. Additionally, Adam incorporates momentum, which helps accelerate the optimization process by accumulating a decaying moving average of past gradients.[381]

In each iteration, after the weighted sums are forwarded through all layers, the gradient of the MSE is computed across all input and output pairs. Then, to propagate it back, the weights of the first hidden layer are updated with the value of the gradient. This process continues until the gradient for each input-output pair has converged.

In classification, the output layer consists of neurons equal to the number of classes in the dataset. Each neuron in the output layer represents the probability of belonging to a particular class. The output is often passed through a softmax activation function to obtain probabilities that sum up to one. In regression, the MLP is configured to produce a single output value,

representing the continuous prediction. A common choice for the activation function in this scenario is either the linear activation function or having no activation function at all.

Another class of NNs are the convolutional neural networks (CNNs).[382] In CNNs, usually an image is represented as a matrix containing the image pixel values. Each pixel value corresponds to the intensity or color of a specific location in the image. These pixel values form the input data for the CNN. Then a filter, or kernel, is applied to capture the spatial features from the image to produce feature maps. These feature maps are then reduced in size with a process called pooling, which separates the feature maps into subregions and merges the features of each subregion. Finally, the pooling layers are flattened and an MLP is applied (Figure 18).

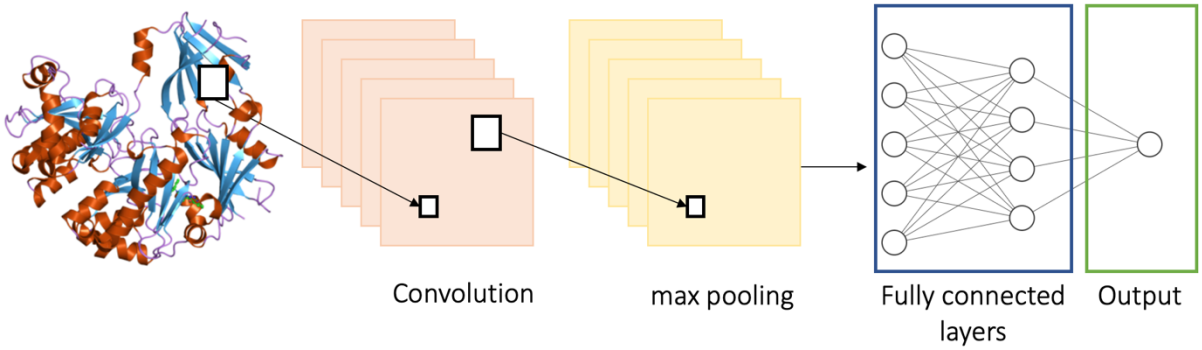


Figure 18: Example of a CNN for image classification. The pixels from the image are passed through a kernel at the convolution stage and then through a max pooling they are reduced in size. Finally, an MLP is used. Figure adapted from Santos *et al.*[383]

Fifteen years ago, a new concept for applying NNs in graph data structures emerged, called graph neural network (GNN).[384,385] GNNs are highly influenced by Networks CNNs and graph embedding. In many scientific fields, including chemistry and biology, data can be naturally represented by graph structures. In software engineering for example, a linked structure of websites can be viewed as a graph, Amazon used GNNs to detect fraud and LinkedIn uses GNNs to make social recommendations and understand the relationships between people's skills and their job titles. At the same time biopharma company GSK maintains a knowledge graph with nearly 500 billion nodes that is used in many of its ML models. In Chemistry, Duvenaud *et al.*[386] proposed neural graph fingerprints (Neural FPs), which calculate substructure feature vectors to get overall representations, while Kearnes *et al.*[387] modeled atom-atom pairs, independently, to emphasize atom interactions. Do *et al.*,[388] used graph transformation policy network that encodes the input molecules and generates an intermediate graph with a node pair prediction network and a policy network, to predict reaction products, and Jaakkola *et al.*,[389] evaluated various graph-to-graph translation models for molecular

optimization. Fout and co-workers[390] proposed a GNN method to learn ligand and receptor protein residue representation, and others have used GNNs for molecular property predictions.[142,391]

In all cases a graph can be thought of as a data structure that is used to describe relationships between entities. Molecular graphs represent the atoms of the molecules with nodes and their bonds with edges (a set of nodes V and a set of edges E). We denote nodes (vertices) with $v \in V$ and edges connecting two nodes v, w with $e_{vw} \in E$ (Figure 19). In addition, each node and edge are assigned a feature vector that stores specific atom and bond information, respectively. The node feature vector is denoted by $f^V(v)$ and contains, for example, information about the atom type or the formal charge of the atom. Note that hydrogen atoms are not represented as nodes but are treated implicitly as atom features by means of the count of hydrogen atoms bonded to a heavy atom. Analogously, the edge feature vector is denoted by $f^E(e_{vw})$ and typically includes information about the bond type. The set of nodes and edges with the corresponding feature vectors describes the attributed molecular graph $G(m) = \{V,E,f^V,f^E\}$ for a molecule m.[392,393]

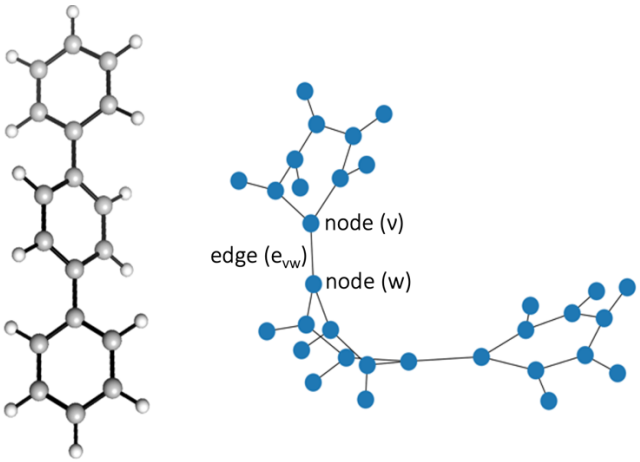


Figure 19: Illustration of a molecule turned into a graph, with the atoms being represented as nodes (v) and the bonds as edges (e_{vw}).

Message Passing Neural Networks

Many different GNN variants have been proposed and finally were unified under the same framework called message passing neural network (MPNN).[392] The MPNN framework consists of a message passing phase and a readout phase. In the message passing phase, structural information within the molecular graph is encoded by means of graph convolutions. The $f^V(v)$ and $f^E(e_{vw})$ of each node and edge aggregate messages from their neighbors iteratively and update the neighborhood. The updated feature vector of a node, after passing a graph

convolutional layer *l,* is typically referred to as the hidden state, h_v^l. The update process of a hidden node state within a graph convolution layer *l* can be depicted as message information from the neighborhood passed to a node and is denoted by (Eq. 2.51), where a message function M_l produces a message from the hidden state of the neighbor node *w* of the previous graph convolution layer h_w^{l-1} , and from the corresponding $f^E(e_{vw})$; then the sum of all messages is passed to the node and is combined with the hidden state vector of the node from previous graph convolution layer h_w^{l-1}. Then an update function U_l is applied, resulting in the updated hidden state of the node h_v^l.

$$h_v^l = U_l(h_v^{l-1}, \sum_{w \in N(v)} M_l\,(\,h_w^{l-1}, f^E(e_{vw}))\,). \quad (2.51)$$

In recent years, GRU-like update functions (Gated Recurrent Unit)[394] are used to incorporate information from the other nodes and from the previous layers to update each node's hidden state alleviating the exploding problem.

The most commonly used GNN architecture is the graph convolutional network (GCN).[395] By stacking multiple graph convolutional layers, the GCN allows each node to receive information from its neighbors. Each additional layer increases the local neighborhood information passed to a node by one additional hop (from node to node) along an edge. However, more sophisticated MPNN architectures have been introduced such as the Graph Attention Networks[396] and the GraphSAGE,[397] but these are beyond the scope of this Thesis.

In the readout phase, the local structure information of the individual nodes is aggregated into a continuous vector representation of the graph, the molecular fingerprint. This aggregation of the single node hidden states is conducted by means of a pooling function (node-wise, max/mean/sum/attention operations) that is applied on node features to get a global graph representation. There are other types of pooling that go beyond the scope of this Thesis.[394] Finally, this molecular fingerprint serves as an input to a feedforward NN, typically a multilayer perceptron (MLP). In Chapter 5, we demonstrate in detail how we built our GNN for the prediction of viscosity and CO_2 solubility in ILs.

Open problems in GNNs

Although GNNs have achieved great success in different fields in this section, we list some open problems that researchers need to be cautious with.[398]

Robustness. As a family of models based on neural networks, GNNs are also vulnerable to adversarial attacks. Compared to adversarial attacks on images or text which only focuses on features, attacks on graphs further consider the structural information. However, in the last few years researchers have made efforts to alleviate such issues.[399]

Interpretability. By their nature NNs are hard to interpret and researchers have only recently stopped treating them as black boxes. Using a GNN inserts an additional layer of difficulty when it comes to generating interpretable models, as graphs can be particularly complex. Only a few methods have been proposed to generate example-level explanations for GNN models.[400] The available techniques answer questions like which input edges or nodes are important, which edges or nodes features are important, and what graph patterns will maximize the prediction of a certain class. Unfortunately, all these questions cannot yet be answered at once.

Graph Pretraining. NN-based models require abundant labeled data, which is costly to obtain. Self-supervised methods have been proposed to guide models to learn from unlabeled data which is easier to obtain from websites or knowledge bases.[395,401]

Complex Graph Structures. Graph structures are flexible and complex in real life applications. Dynamic graphs or heterogeneous graphs have been proposed to deal with complex graph structures.

Ensemble machine learning

To improve the predictive performance, ensemble ML methods have been developed that combine multiple ML models. This approach is applicable to both shallow learners and deep NNs or GNNs. There are three main strategies for performing ensemble ML: bootstrap aggregation (bagging), boosting, and stacking.

In bootstrap aggregation, multiple models are trained, each one with a subset of samples and features, selected through sampling with replacement.[402] Then, the majority vote of the models gives the final prediction. An example of an ensemble model using DTs and bootstrap aggregation is the RF (Figure 20a).[35] RF splits the nodes based on the best split among a random subset of features at each node, adding an element of diversity to the ensemble. An alternative to RF is the Extremely Randomized Trees[403] or Extra Trees (ET). Unlike RF, ET does not bootstrap observations. Furthermore, nodes in ET are split based on random splits rather than the best splits. The randomness in ET does not arise from bootstrapping the data, but rather from the random splits of all observations. This makes ET robust in the presence of noisy

features, which is why it finds utility in this Thesis for maintaining higher performance under such conditions.

The second strategy is called boosting, where multiple models are trained sequentially, with each new model being influenced by the performance of the previous models.[404] Again, the majority vote of the models draws the final predictions, but the votes are weighted based on the models' performances. An example is AdaBoost,[405] where the output of "weak learners" (usually DT) is combined into a weighted sum that represents the final output of the boosted model. A modern version of AdaBoost is extreme gradient boosting (XGBoost)[406] algorithm, which has increased speed and performance, as it introduces regularization parameters to reduce over-fitting (Figure 20b).

In stacking, a meta-estimator is trained on the predictions of previous ML models (RF, a kNN, and a SVMs). For example, the LR model can be trained on the predictions of a DT, a kNN, and a SVM (Figure 20c).[407]

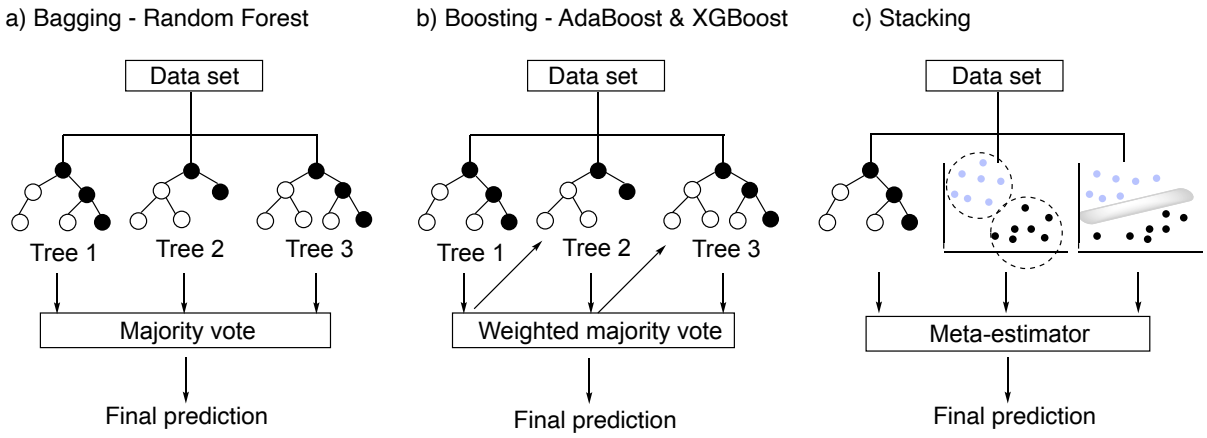


Figure 20: Representation of Ensemble learning algorithms. a) The bagging technique represented by RF. b) The boosting technique represented by AdaBoost and XGBoost. c) The stacking technique using DT, and kNN and SVM.

Hyper-parameter optimization

Each ML algorithm has hyper-parameters, which are parameters that control the behavior of the training algorithm and affect the model's capability to identify patterns and correlations. Examples of hyper-parameters include the number of the neighbors (k) in the kNN classifier, the kernel type in SVMs, the maximum tree depth in the DT classifier, and the number of DT in RF. Depending on the hyper-parameters, the model may capture the trends in the data, or it may suffer from under-fitting or over-fitting. Thus, the hyper-parameters must be tuned.

The main strategies for hyper-parameters tuning are the grid search and randomized search. In grid search, fixed parameter combinations are selected *a-priori*, while in randomized search a range of values for each parameter is specified and a finite number of parameter combinations are randomly selected from these ranges. Each parameter combination results in a different model. Then, each model is trained and validated and the model with the best performance is selected. Depending on the dataset, not all hyper-parameters are significant.[408] Randomized search is preferred over the grid search when the number of parameter combinations that can be evaluated is finite because the sampling of the important hyper-parameters is more effective.

It should be mentioned that more sophisticated techniques for hyper-parameter tuning exist[409] such as the Bayesian hyper-parameter optimization,[410] but these are beyond the scope of this Thesis.

2.2.3. Performance metrics for evaluating machine learning models

As shown in Figure 6, after selecting the appropriate descriptors and choosing a learning algorithm one needs to evaluate and validate that the correct choice of descriptors and algorithm has been made. Performance metrics serve as quantitative measures that capture various aspects of model performance. In both regression and classification tasks, different metrics are employed due to the distinct nature of the problems. Regardless of the task at hand, it is crucial to ensure the reliability and generalizability of the model's performance assessment.

This is achieved through the process of validation, which provides insight into how the model is likely to perform on unseen data. Validation can be categorized into internal and external validation. Internal validation is considered necessary but not sufficient.[356] Hence, external validation must be carried out prior to application of the model. During internal validation, also called cross-validation, the dataset is randomly split into k subsets (k-fold). In every iteration, a subset is the test set and the remaining of the subsets compose the training set. The error of the predictions is recorded in every iteration. The average of the k recorded errors is called the cross-validation error and will serve as the performance metric for the model (Figure 21).

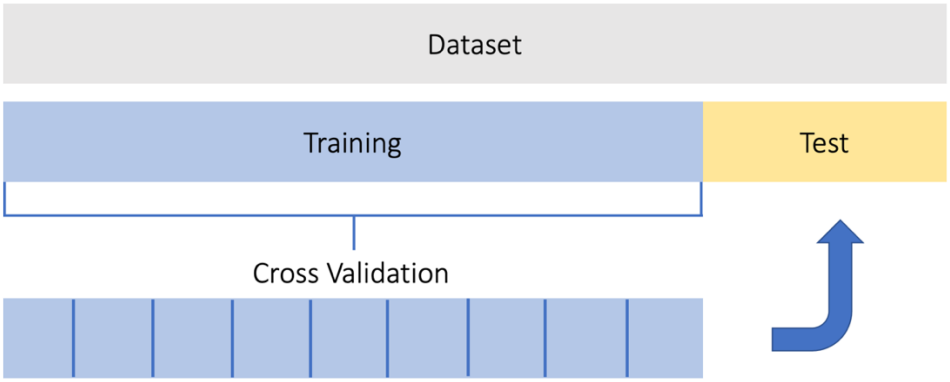


Figure 21: Graphical representation of the dataset split into Training and Test sets. The training set is split further for k-fold cross-validation, and the results are used for the external validation (test set).

To perform external validation, a set of data that has not been used to train the model is used (called test set). This set is held out of the original data set and is often chosen randomly. It is important to note that when the training and test sets are chosen randomly the error metrics may vary due to the inherent variability in the data. To establish a reliable conclusion about the model's performance, it is recommended to perform multiple splits of the data into training and test sets. For instance, in some of the projects described here, we adopt the practice of conducting 100 such splits. On the other hand, if the test set is chosen explicitly for a particular reason, such as representing a specific subset of data or mimicking a real-world scenario, repeating the runs may not be necessary. In such cases, a single evaluation with the chosen test set can suffice.

Regression

In most real-world problems, the error cannot be precisely calculated, and it must be estimated. It is therefore essential to choose an appropriate estimator of the error. Least-squares is a common method for model construction (Eq. 2.52). The goal of this method is to find the set of parameters that minimizes the sum of the squared residuals between the observed data and the predictions of the model. In other words, it seeks to find the line (or plane or hyperplane) that best fits the data (Figure 22).[6]

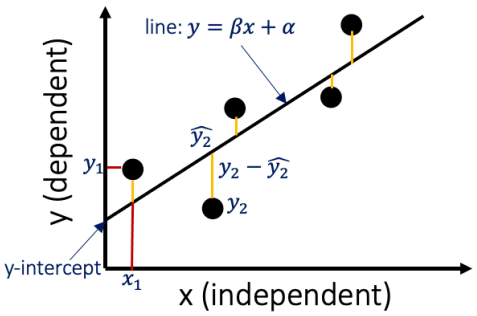


Figure 22: Illustration of the least squares method. The data points are away from the best fit line, highlighting the vertical distances (yellow) between each observed point and the corresponding point on the best-fit line, symbolizing the minimization objective defined by Equation 2.51.

$$\text{Minimize: } \sum_{i=1}^{n} (Y_i - \widehat{Y}_i)^2. \quad (2.52)$$

The most common statistical measure of good fitness is the correlation coefficient, R^2 (Eq. 2.53). If $R^2 > 0.5$, this means that the explained variance is greater than the unexplained variance. A good linear correlation (R^2 close to 1.0) between the predicted values and the measured values indicates that the obtained model adequately approximates the system under study. The acceptable value is up to user's judgement (recommended value by Tropsha is $R^2 > 0.6$).[411]

$$R^2 = 1.0 - \frac{\sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n} (Y_i - \bar{Y})^2}, \quad (2.53)$$

where Y_i, $\hat{Y}_i$, $\bar{Y}$ are the measured, predicted and averaged dependent variable, respectively and n equals the number of data points.

The performance of a model can also be measured in terms of prediction error. The MAE (mean absolute error - Eq. 2.54) represents the average of the absolute difference between the actual and predicted values in the dataset, and it measures the average of the residuals in the dataset.[6]

$$MAE = \frac{\sum_{i=1}^{N} \left|Y_i - \hat{Y}_i\right|}{N}. \quad (2.54)$$

MSE (mean squared error - Eq. 2.55) represents the average of the squared difference between the original and predicted values in the data set and it measures the variance of the residuals.[6]

$$MSE = \frac{\sum_{i=1}^{N} (Y_i - \hat{Y})^2}{N}. \quad (2.55)$$

The RMSE (root mean square error - Eq. 2.56) is the square root of the variance of the residuals, and it can be interpreted as the standard deviation of the unexplained variance. It has the useful property of being in the same units as the response variable. Lower values of RMSE indicate better fit. RMSE is a good measure of how accurately the model predicts the response, and it is the most important criterion for fit if the main purpose of the model is prediction.[6]

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Y_{i,} - \hat{Y})^2}{N}}. \quad (2.56)$$

Classification

In binary classification, the predicted probabilities or scores from a model can be converted into class predictions by applying a threshold (0-1, usually 0.5). Instances with predicted probabilities above the threshold are classified as positive, while those below the threshold are classified as negative. The predictions can be summarized into a confusion matrix, where TP stands for true positives, FP for false positives, FN for false negatives, and TN for true negatives (Figure 23a).[412]

The performance metrics for a ML binary classifier are computed from the confusion matrix. The most common performance metric is accuracy (Eq. 2.57), which is the fraction of correct predictions over the total predictions:

$$\text{Accuracy } = \frac{TP + TN}{\text{Total predictions}}. \quad (2.57)$$

The accuracy score metric may be misleading in imbalanced datasets. For example, consider that the number of samples of the minority class (e.g., positive class) is 100 and the number of samples of the majority class (negative class) is 9,900. If the algorithm classifies all samples to be in the negative class, the corresponding confusion matrix is shown in Figure 23b.

a)

TN Predicted: negative Actual: negative	FP Predicted: positive Actual: negative
FN Predicted: negative Actual: positive	TP Predicted: positive Actual: positive

b)

TN = 9,900	FP = 0
FN = 100	TP = 0

Figure 23: a) Confusion Matrix. b) An example where accuracy is a misleading metric as explained by Eq. 2.57.

and the accuracy score is derived as (Eq. 2.58):

$$\text{Accuracy } = \frac{0 + 9{,}900}{10{,}000} = 0.99, \quad (2.58)$$

which does not represent the actual performance of the model as the samples of the minority class were wrongly labeled in their entirety. To overcome this limitation of the accuracy score, more sophisticated metrics have been introduced.

Recall (or sensitivity) expresses the amount of correctly predicted TP (Eq. 2.59), while precision expresses the predicted true positives that are actually true (Eq. 2.60).

$$\text{recall or sensitivity} = \frac{TP}{TP + FN}, \quad (2.59)$$

$$\text{precision} = \frac{TP}{TP + FP}. \quad (2.60)$$

Increasing precision often comes at the expense of recall, as raising the threshold for classifying positive instances leads to fewer FP but potentially more FN. Similarly, emphasizing recall may result in higher FP but fewer FN.[412]

The F score is derived by taking the harmonic mean of the precision metric and the recall metric.[412] The general formula of the F score is derived based on a positive real variable β, where β determines the importance of recall over precision (Eq. 2.61). When $\beta = 1$ (F1 score), recall and precision are weighted equally (Eq. 2.62), when $\beta < 1$ more weight is given in precision, and when $\beta > 1$ recall is favored.

$$F_\beta = (1 + \beta^2)\frac{\text{precision} * \text{recall}}{(\beta^2 * \text{precision}) + \text{recall}}, \qquad (2.61)$$

$$F_1 = 2\frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \quad (2.62)$$

The F1 score measures the performance in each class separately, thus the macro F1 score (Eq. 2.63) which is the average performance of all classes is calculated with values ranging between 0 and 1:

$$\text{Macro } F_1 \text{ score} = < F_1 \text{score} >, \quad (2.63)$$

where angle brackets denote average over each class F1 score.

The geometric mean (g-mean) of sensitivity (Eq. 2.59) and specificity (Eq. 2.64) can also be measured and it is formulated as (Eq. 2.65):

$$\text{specificity} = \frac{TN}{TN + FP}, \quad (2.64)$$

$$\text{g} - \text{mean} = \sqrt{\text{sensitivity} * \text{specificity}}. \quad (2.65)$$

Another metric is the Matthews correlation coefficient (MCC), which receives values between -1 and 1 and is formulated as (Eq. 2.66):

$$MCC = \frac{(TP * TN - FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \quad (2.66)$$

Another way to evaluate the skill of a classifier is by plotting the ROC curves (or receiver operating characteristic curve) which summarize the performance of a binary classification model (Figure 24). The x-axis indicates the FP rate, and the y-axis indicates the TP rate (sensitivity or recall). Ideally, the TP rate will be 1 and the FP rate will be 0, achieving perfect skill. By evaluating the TP and FP for different threshold values, a curve can be constructed that stretches from the bottom left to top right and bows toward the top left. This curve is called the ROC curve. A classifier that has no discriminative power between positive and negative classes will form a diagonal line. Models represented by points below this line have worse than no skill. The ROC curve is a popular diagnostic tool for classifiers on balanced and imbalanced binary prediction problems alike because it is not biased to the majority or minority class. However, it can be challenging to compare two or more classifiers based on their curves. Instead, the area under the curve (AUC) can be calculated to give a single score for a classifier model across all threshold values. The score is a value between 0 and 1.

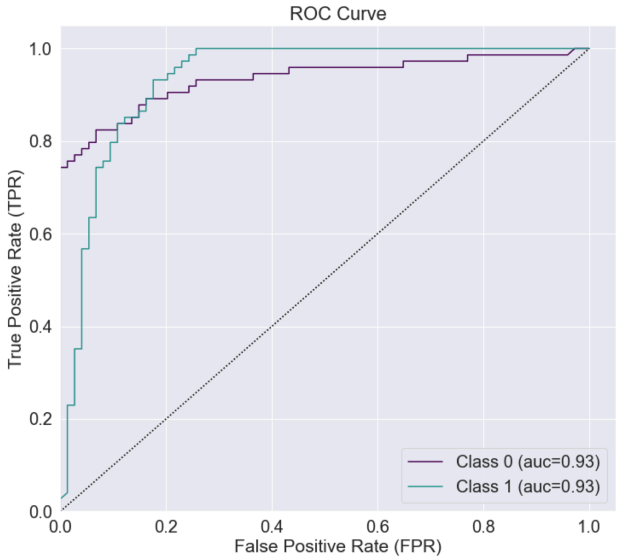


Figure 24: Example of a ROC curve. The classifier gains skill fast as it bows towards the top left. The AUC is also high (close from 1) for both classes.

A precision-recall curve (or PR Curve) is a plot of the precision (y-axis) and the recall (x-axis) for different probability thresholds. A skillful model is represented by a curve that bows towards a coordinate of (1,1). A no-skill classifier will be a horizontal line on the plot with a precision that is proportional to the number of positive examples in the dataset. For a balanced dataset this will be 0.5. The focus of the PR curve on the minority class makes it an effective diagnostic for imbalanced binary classification models. The precision-recall AUC summarizes the curve as a single score.

2.2.4. Interpretation of machine learning models

In the words of E. Wigner: *"It is nice to know that the computer understands the problem, but I would like to understand it too"*.[413]

Interpretability of ML models is an important area of research that has received significant attention in recent years. While the models can be highly accurate in predicting various chemical properties and reactions, it is often difficult to understand how they make their predictions. Lack of interpretability can limit the usefulness of these models, making it challenging to gain insights into the underlying chemical mechanisms, or to use the models for the design of new compounds or reactions. In organocatalysis there have been several attempts to offer interpretable models, among them Sigman[89,94,97] and Fey[414] have managed to capture and explain in detail the most important descriptors of their models, allowing for a holistic approach in predictive models.

One approach to address this challenge is to use visualization techniques to represent the model's predictions and to highlight the chemical features that are most relevant. Heat maps, PCA,[16] t-distributed stochastic neighbor embedding (t-SNE),[415] and tmap[416] are some of these techniques.

Heat maps are a type of graphical representation that use color to visualize the magnitude of values in a matrix. Heat maps are often used in combination with clustering algorithms to group similar samples or compounds together based on their feature profiles. PCA transforms high-dimensional data into a lower-dimensional space while preserving as much of the original variance as possible. This allows complex data sets to be visualized in two or three dimensions, making it easier to identify patterns or clusters in the data. In Chapter 3, we give an example of how heat maps and PCA are employed by *Pythia*.

t-SNE is a nonlinear dimensionality reduction technique that is commonly used for data visualization. Like PCA, t-SNE transforms high-dimensional data into a lower-dimensional space, but it does so in a way that preserves the relationships between individual data points. This allows t-SNE to produce high-quality visualizations that reveal subtle patterns and relationships in complex data sets (Figure 25). tmap is a package for creating thematic maps and geographic visualizations. In chemistry, tmap can be used to visualize the spatial distribution of chemical species in a sample or to plot the location of chemical or biological assays on a map (Figure 26).

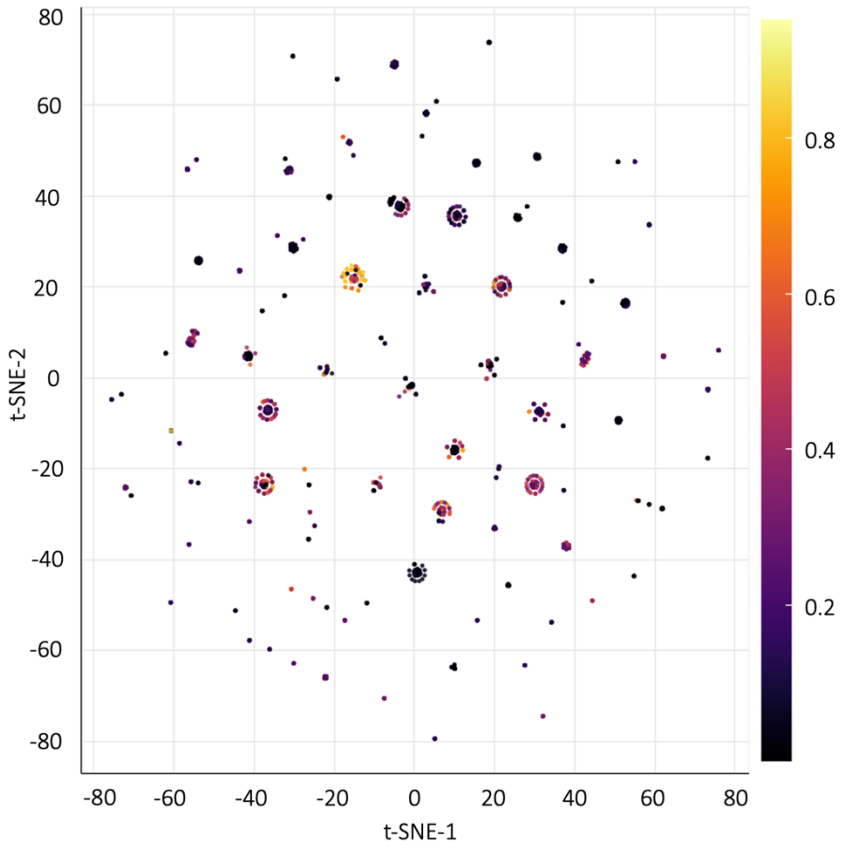


Figure 25: Illustration of a t-SNE map. The small clusters represent data that are structurally similar. The map is constructed from the solubility data in Chapter 5.

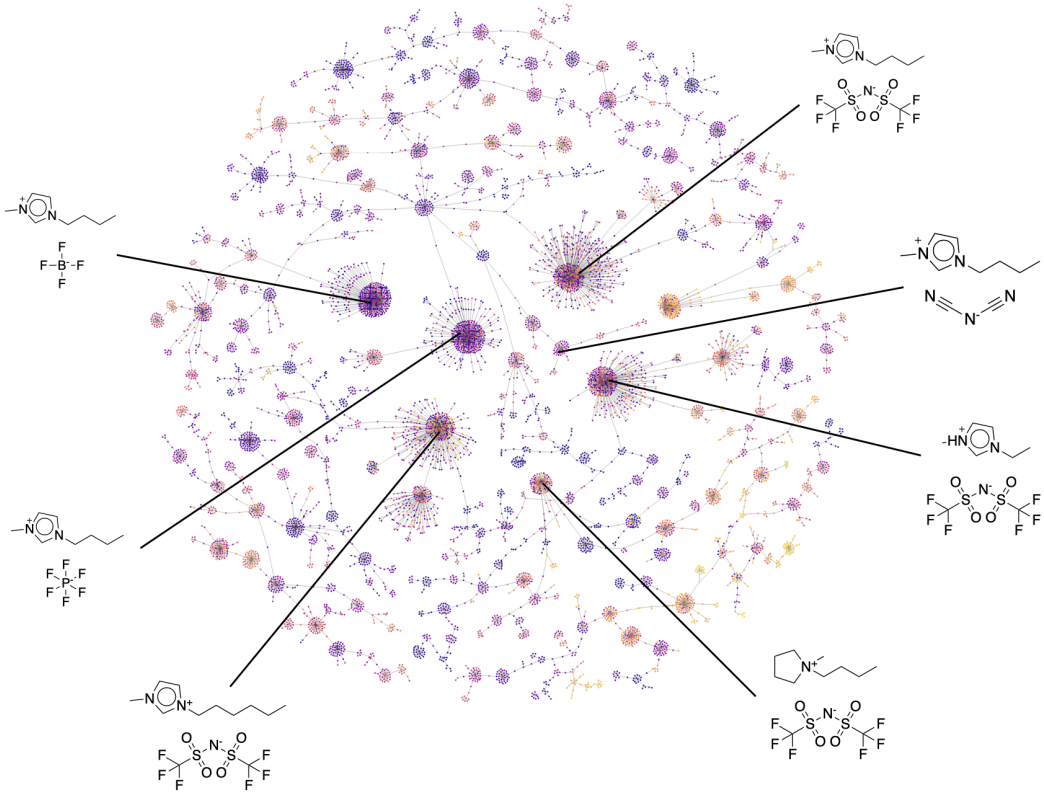


Figure 26: Illustration of a tmap. The clusters represent data that are structurally similar. Data that are not the same but have structural similarities are connected. The map is constructed from the solubility data in Chapter 5.

Another approach is to use feature importance analysis, which can be obtained from common ML packages like Scikit-learn. It involves identifying the chemical features that are most important for the model's predictions. Recursive feature elimination is another easy way to identify how the model changes with respect to feature changes. It works by recursively removing features and building a model on the remaining features until the desired number of features is reached and it can be used in any type of ML.

SHapley Additive exPlanations (SHAP) is a technique that provides a unified framework for explaining the output of ML models.[417] SHAP assigns each input feature an importance score that describes its contribution to the model's output, allowing users to gain insights into the underlying mechanisms of the model. In classification the Shapely values are calculated based on class 1, which usually represents good performance. Each training point is plotted for every feature. Points in red have high value of the feature and points in blue have low value of the feature. The Shapely values for each point are on the x axis. Positive Shapely values mean that this feature contributed positively to predicting this point is in class one, and the opposite for negative values. In Chapter 3, we give an example of how SHAP is employed by *Pythia*, and its results are further explained.

NNs are known for their black-box nature. However, interpretability techniques have been developed to help understand and explain their inner workings. One popular method is called *saliency mapping* or *gradient-based attribution*.[418] This method uses the gradient of the output with respect to the input to identify which features of the input were most important in making the prediction. This allows for the visualization of the parts of the input that contributed the most to the final output. Another technique is *layer-wise relevance propagation*,[419] which uses a backward propagation algorithm to attribute the output to the input, allowing us to understand how each layer of the NN contributes to the final prediction. These techniques go beyond the scope of this Thesis as we do not focus on NNs.

Finally, it is important to acknowledge that caution must be exercised when interpreting models that lack fundamental physical or chemical theory. Users should be wary of drawing definitive conclusions solely based on trends observed in non-physicochemical data.

3. Pythia, a machine learning toolkit

The code presented in this chapter has been written by myself. Zonghua Bo and Bernadette Lee have tested and used the code for their own research, their results are not presented in this Thesis. I am thankful to them for their constructive criticism and bug reporting.

This chapter introduces *Pythia*, a ML toolkit for chemistry, based on open-source software and the Jupyter environment.[420] It employs fingerprints, Mordred, and QM descriptors as input features and offers various options for shallow learners and ensemble models for regression and classification tasks. One of the key advantages of *Pythia* is its user-friendly format, made possible by its development in Jupyter Notebooks. This enables users to easily interact with the code and modify it to suit their needs. This user-friendly design makes the toolkit especially valuable for researchers who may not have extensive programming experience but are interested in applying ML techniques in their work. We would like to believe that just like *Pythia*, the legendary high priestess of the Greek god Apollo, who provided prophetic guidance, our *Pythia*, the ML toolkit, offers insights and predictions that can guide researchers in their quest for new discoveries in chemistry.

In the following sections, we will discuss other tools that offer ML predictions for chemistry, identify the existing gap, and highlight the need for a tool like *Pythia*. We will then illustrate *Pythia's* implementation and its use through tutorial-like examples. The chapter closes with general conclusions and a discussion for future implementations.

3.1. Literature review

In Chapter 1, we discussed several works on predicting chemical properties. Most of them use standard techniques, features, and ML algorithms, but do not make their codes publicly available. Others employ toolkits such as DeepChem,[75] ChemML,[421] ML4Chem,[422] ChemProp[106] and OpenChem.[423] These tools have played a critical role in providing users with helpful platforms for deep learning in drug discovery, quantum chemistry, material sciences and biology. While such tools have revolutionized the way chemistry is performed, they often lead to black box models which can be overwhelming for an inexperienced user. Below we briefly describe these tools, their range of applicability, advantages, and limitations.

DeepChem[75] is an open-source platform for drug discovery and materials science that uses deep learning and other ML methods to predict molecular properties, reactions, and structures.

The platform provides a range of tools and algorithms for cheminformatics and bioinformatics, including molecular featurization, graph convolutions, and generative models. One of the strengths of DeepChem is that it allows researchers to integrate new data and models into the platform. It also provides a user-friendly interface for data exploration and visualization, as well as a range of pre-trained models for common tasks, such as predicting toxicity and protein-ligand binding affinity. However, developing a new model with DeepChem requires writing a considerable amount of Tensorflow code. In addition, DeepChem does not allow modular design features, such as encapsulation and reusability of standard deep NN blocks (e.g., encoders, decoders, and embedding layers).[422] Finally, from our own experience, DeepChem cannot be used in combination with other non-integrated tools. For example, descriptors available in DeepChem are non-readable by other Python libraries.

Since 2014, Hachmann and co-workers have been developing a software ecosystem that combines computational modeling, virtual high-throughput screening and big data analytics which is composed from ChemLG, ChemHTPS, ChemBDDB, and ChemML. ChemML[421] is an advanced Python toolkit that uses deep learning (including active and transfer learning) to extract structure-property relationships.

ML4Chem[422] is another tool that implements deep learning algorithms for chemistry. Unlike DeepChem and ChemML, ML4Chem offers an atomistic module, where ML algorithms learn underlying relationships between molecules and properties, treating atoms as central objects, with the use of SOAP descriptors and autoencoders.

Jensen and co-workers have explored deep learning either in the form of NNs or GNNs, and over the years have built upon their codes resulting in ChemProp.[106] They have implemented MPNNs to predict the likelihood of a molecule to inhibit the growth of E. coli[424], molecular toxicity[105] and, very recently, the viscosity of binary liquids mixtures.[425]

OpenChem[423] is a PyTorch-based deep learning toolkit for computational chemistry and drug design available on GitHub.[426] It enables ML model building, compound generation, and property optimization in a single framework. The wide applicability of OpenChem has been illustrated for several tasks,[423] including: (i) logP predictions, for which they trained a GNN using 14,500 molecules obtained from the public version of the PHYSPROP database,[427] (ii) prediction of bioactivity for 12 receptors using data from the Tox21 challenge and an RNN, (iii) prediction of melting temperatures, using a MolecularRNN model pretrained on the curated ChEMBL24 data set of 1.5 million molecules.

While these highly advanced tools have revolutionized ML in chemistry and achieved state-of-the-art performance in numerous tasks, they are not a universal solution for all datasets. In many cases, simple techniques and shallow learners can provide comparable performance with much lower computational costs and greater interpretability. This is particularly true for datasets with relatively few features and a limited amount of training data, as deep learning models require significant computational resources and large amounts of labeled data to achieve optimal performance. Therefore, it is important for researchers to carefully evaluate the suitability of deep learning and other advanced techniques for a given dataset, considering factors such as the dataset size, feature complexity, and model interpretability. Ultimately, the choice of method should be based on a careful analysis of the dataset and the specific research question at hand.

In 2022, Asparu-Guzik, Sigman and co-workers released *kraken*, a discovery platform designed to explore monodentate organophosphorus (III) ligands.[428] Using semiempirical and DFT data to compute property descriptors for over 1,500 ligands they trained ML models to predict the properties of over 300,000 new ligands. The authors highlight how existing datasets can be leveraged to accelerate ligand selection during reaction optimization. They utilized several ML models such as regression models, RF, Gaussian Process (GP), graph CNN (using ChemProp), and finally ensemble models (stacking regressor), the later yielding the highest correlation and lowest errors. This example illustrates that deep learning is not always the ultimate solution, and there are instances where more traditional methods can yield significant power.

Pythia was developed, with the aim to bridge the gap between available ML frameworks and their use by the wider chemistry community. By leveraging simple ML techniques, *Pythia* aims to facilitate the development of data-driven prediction models for chemistry and enhance the efficiency and accuracy of such predictions while making them accessible to the broader chemical research community.

3.2. Description of Pythia

Pythia is a modular toolkit, implemented in Python *v.3.7* and organized in Jupyter Notebooks.[420] It takes advantage of available open-source libraries, such as RDKit[198] *v.2019.09.3* and scikit-learn[378] *v.1.0.2,* pandas[429] *v.1.2.3*, NumPy[430] *v.1.21.6*, Matplotlib[431] *v.3.0.2*, imbalanced-learn[374] *v.0.10.1*, statsmodels[432] *v.0.14.0*, SciPy[433] *v.1.7.3*. The ML

algorithms available in *Pythia* (through scikit-learn) are LASSO with cross-validation (LASSOCV), LR, SVM, kNN, GP, Bayesian Regression, DT, RF, ET, and AdaBoost. They have been selected for their ability to handle high-dimensional data, their effectiveness in handling noise and outliers and their low computational cost. While XGBoost was initially included, it was later removed from our default notebooks due to its higher computational cost. However, users can add additional ML algorithms based on their needs.

Pythia is organized in six modules, each consisting of relevant functions that can be called from the Jupyter Notebooks. These modules are documented and structured according to their intended purpose, ensuring ease of use for both experienced and novice users. This approach keeps the Jupyter Notebooks uncluttered, while providing users with the flexibility to add their own functions to the appropriate module. In Table 2 we provide the description for each module.

Table 2: Description of Pythia's modules.

Module	*Description*
classification metrics	calculation of confusion matrix, accuracy, g-mean, precision, recall, generalized f, MCC, AUC
fingerprints	generation of Morgan, rdkit, atom pair, torsion fingerprints and MACCS keys with rdkit
molecules and images	SMILES to molecules and images with rdkit
plot sklearn	plot of parity plots, ROC curves, confusion matrix with matplotlib
scaling	z, min-max, logarithmic scaling
workflow functions	correlation tests, training for regression and classification, ensemble learning with sklearn

Pythia's Jupyter Notebooks are also organized, and named after their intended purpose, offering standard workflows for regression and classification. Users have the flexibility to customize the workflow, by adding or removing steps, as needed for their specific research question. By structuring the Notebooks in this manner, *Pythia* provides users with a clear and flexible framework for implementing the various modules and functions included in the toolkit. In Table 3 we provide the description for each Notebook.

Table 3: Description of Pythia's Jupyter Notebooks.

Jupyter Notebook	*Description*
data analysis	data exploration, visualization, scaling
regression-fingerprints	regression with fingerprints, data set split, ensemble models
regression-Mordred	regression with Mordred, feature elimination techniques, data set split
regression-DFT	regression with DFT descriptors, PCA, data set split
classification-fingerprints	classification with fingerprints, feature exploration, synthetic data, data set split
classification- Mordred	classification with Mordred, feature elimination and exploration, synthetic data, data set split
classification-DFT	classification with DFT descriptors, synthetic data, data set split, interpretability

Irrespective of the chosen descriptors, model training across different notebooks follows a consistent methodology. Two primary approaches are offered for model training and evaluation: (i) employing cross-validation on the complete dataset through the utilization of the `kfold test regressor with optimization` and `kfold test classifiers with optimization functions` from the `workflow functions` module, and (ii) adopting a train-test split without cross-validation on the training set using the `split test regressors with optimization` and `split test classifiers with optimization` functions from the `workflow functions` module.

In technique (i), the data is partitioned into *k*-folds of equal size. The model is trained on *k-1* folds and assessed on the remaining fold. This process is repeated *k* times, with each fold serving as the validation set once. The final performance measure is obtained by averaging the performance scores derived from each fold. This approach presents the advantage of leveraging the entire dataset for both training and evaluation. Consequently, it provides a dependable estimation of the model's performance and facilitates an assessment of its generalization capabilities.

In technique (ii), the dataset is divided into two distinct subsets: a training set and a test set. The training set is utilized to train the model, while the test set remains separate and is employed to evaluate the model's performance. Unlike cross-validation, the training set is not further divided into multiple folds for validation. This approach offers simplicity and computational efficiency compared to cross-validation. However, it does possess certain limitations. The performance estimate acquired from a single train-test split may fluctuate based on the specific instances encompassed within the training and test sets. This inherent

randomness in the split can introduce a bias in the performance estimation, particularly when dealing with small or imbalanced datasets. Nevertheless, this technique proves valuable when a specific test set is predetermined. Alternatively, it is advisable to employ this technique multiple times and obtain the average metrics from all splits, thereby mitigating the impact of randomness and attaining more reliable performance estimates.

In both techniques, grid search optimization is executed to determine the optimal parameters for each algorithm, enhancing its performance. For each algorithm, the predictions either from the *k*-fold cross-validation and/or the single set of predictions, are stored in a dedicated folder. These predictions are then utilized to calculate metrics, which are then plotted using the `metrics for regression`, `get confusion matrix` and `calculate confusion-based metrics` functions from the `workflow functions` and `classification metrics` modules respectively. The generated folders aid in organizing the predictions, enabling easy retrieval and analysis of the metrics, which offer valuable insights into the performance of the models and facilitate informed decision-making. The availability of specialized functions for calculating metrics and generating visualizations streamlines the evaluation process, ensuring the comprehensive assessment of the model's performance.

For classification a bound needs to be determined by the user. Often, classes are imbalanced, so we utilize synthetic sampling methods to generate additional sampling points for the minority class or under-sample the majority class. Based on whether the descriptors are non-categorical, both categorical and non-categorical, and only categorical, the SMOTE, SMOTENC and SMOTEN algorithms are implemented respectively. For small datasets, we suggest oversampling the minority class instead of removing points from the majority class. A new data frame containing the synthetic points is created and used for the training and evaluation of the model in the manners described earlier. Finally, the function `which are misclassified` from the `workflow functions` module, can be used to identify data points that have been misclassified and interrogate the model further.

We also provide methods, such as PCA and SHAP, to gain a deeper understanding of the generated models, uncovering the underlying patterns and factors driving the predictions.

Pythia is available on GitHub (https://github.com/duartegroup/PythiaChem) streamlining the setup process and ensuring that dependencies are properly managed. Its modular structure allows for seamless integration with other Python toolkits, without disrupting the core functionality. Furthermore, detailed comments are provided throughout the code and Jupyter

Notebooks. Additionally, a description of *Pythia* is given in the README file, offering users a comprehensive guide to utilizing the toolkit. Please note that the GitHub page is subject to updates, and the version of *Pythia* described in this Chapter is the one provided in the supplementary material.

3.3. Discussion of the Jupyter Notebooks

In this section we discuss in detail the seven Jupyter Notebooks available in *Pythia*. We illustrate the capabilities of each Notebook, using as an example a dataset of enantioselective Strecker synthesis of α-amino acids,[434–438] containing 119 reactions, composed of 19 catalysts and 63 substrates in different temperatures. The target values for *ee* are reported in $\Delta\Delta G^{\ddagger}$ (kJ/mol). For classification, the bound is set to $\Delta\Delta G^{\ddagger} < 4$ kJ/mol (class 0) and $\Delta\Delta G^{\ddagger} >= 4$ kJ/mol (class 1). Here, this dataset is solely used to showcase the workflow and functionality of the code. A comprehensive discussion of the models generated, their accuracy, and key findings are presented in detail in Chapter 4 (§4.3).

Notebook 1: Data analysis

The purpose of this Notebook is to offer comprehensive data exploration, visualization, and statistical analysis, enabling users to uncover patterns, trends, and valuable insights from complex datasets.

First, the necessary modules (`scaling, fingerprints` and `molecules and images`) are imported. After this, the dataset is imported as a *.csv* file containing the SMILES entries for the molecules under study, along with any other information (e.g., in our case temperatures). *Pythia* can identify and remove duplicate entries and then a histogram displaying the frequency of the target values is generated, enabling users to understand the distribution of good or bad data points. For example, Figure 27, shows that there are about 23 data points with $\Delta\Delta G^{\ddagger}$ values between 0 and 1 and only two with a $\Delta\Delta G^{\ddagger}$ value of > 10 kJ/mol. This suggests that a model constructed from this dataset may struggle to accurately predict reactions with $\Delta\Delta G^{\ddagger}$ values higher than 8 kJ/mol as there are not enough examples in this region.

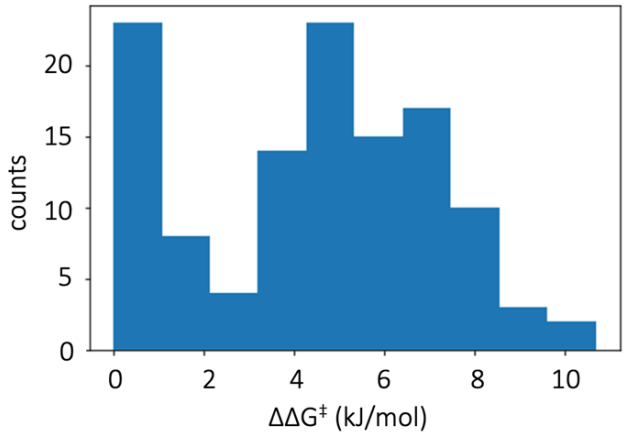


Figure 27: The $\Delta\Delta G^{\ddagger}$ values are represented with a histogram to identify the frequency of occurrences. High values (above 8 kJ/mol) are rare.

Similarly, the number of unique catalysts and substrates and their frequency of occurrence can be identified. The model might struggle to predict the outcomes of less frequently occurring molecules and it useful to identify these molecules early on.

Assessing the structural similarity of the molecules in the dataset is also important, as it may help estimate the general applicability of the model. For example, there might be cases where catalysts differ only by small substitutions, in which case the model might struggle to accurately predict structurally diverse catalysts. To determine the molecular similarity, we utilized Morgan fingerprints and *Tanimoto* similarity, which can be calculated by calling the `bulk_similarity` function from the `fingerprints` module. The code returns a list of tuples containing the index of the substrate or catalyst, and the average similarity metric for that molecule compared to the rest. Figure 28, shows the output of the corresponding cell, indicating that catalyst number **5** has the lowest *Tanimoto* similarity, while catalyst **17** has the highest. Therefore, we could expect catalyst **5** to be an outlier.

```
[(5, 0.25835061719204716),
 (6, 0.319314691424289),
 (13, 0.33495123828660134),
 (12, 0.33639116316944445),
 (16, 0.38428015261828974),
 (15, 0.417903502287532),
 (11, 0.4529434934790709),
 (8, 0.4666771926100395),
 (14, 0.472685747312499),
 (18, 0.49914050380270664),
 (9, 0.5399867438155438),
 (10, 0.5757240890297227),
 (2, 0.6038699401759114),
 (3, 0.6038699401759114),
 (1, 0.6076950151870116),
 (7, 0.6156704800119956),
 (4, 0.6181229051231236),
 (0, 0.6286069957532912),
 (17, 0.6286069957532912)]
```

Figure 28: Output of cell where the average *Tanimoto* similarity between the catalysts is calculated.

Applying PCA to fingerprint data can provide insights into the underlying structure and relationships between the molecules in the dataset. It can help visualize molecular similarities and identify potential clusters or trends. This is especially useful when exploring the chemical space or assessing the diversity of a compound library. We note here that in this case, the principal components (PC) serve as a means of capturing the variance in the dataset and identifying patterns, rather than providing direct insights into the molecular properties. Interpreting the meaning of the PC derived from fingerprints can be challenging, as they may not correspond to easily interpretable molecular features. *Pythia* provides a way to perform PCA and plot the chemical space based on fingerprints. Figure 29, illustrates the corresponding plot for the catalysts. The plot is interactive and by placing the mouse on the datapoint the user can see the number of the corresponding catalyst. The results from the average *Tanimoto* similarity (Figure 28) are not to be confused with the PCA plot (Figure 29) as the first shows the similarity between a molecule and all the other molecules, whereas the latter plots each point individually.

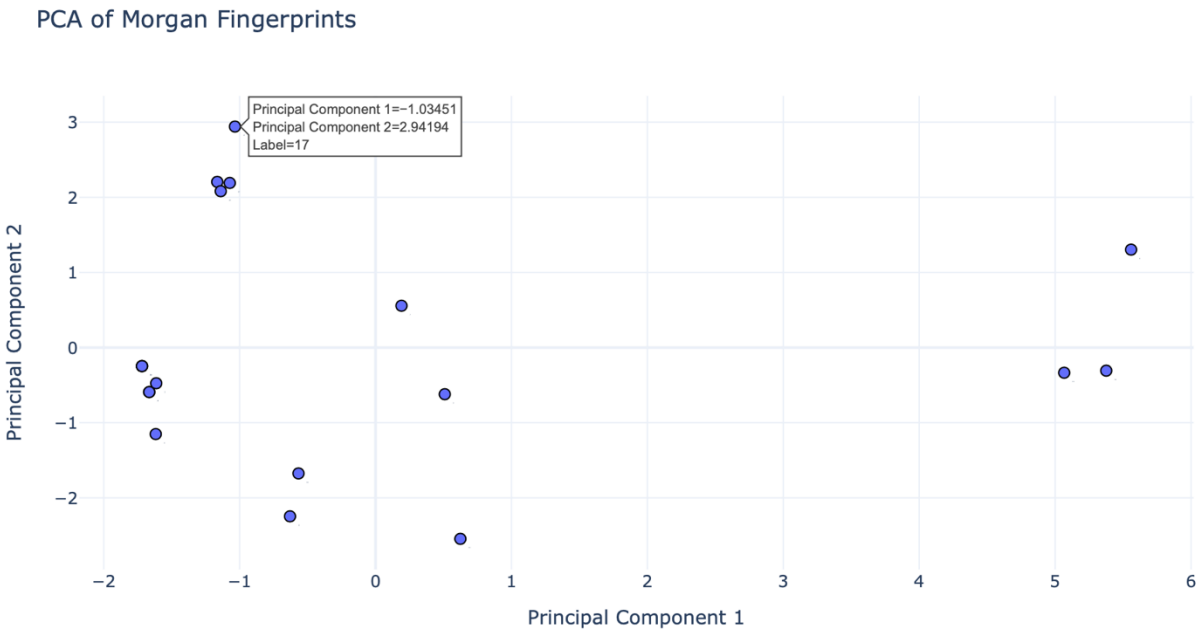


Figure 29: Visualization of the chemical space of the catalysts under study, using PCA and Morgan fingerprints. Data points are clustered together, indicating similarity among certain datapoints. However, non-clustered points also exist (e.g., left hand-side of the plot), suggesting diverse chemical characteristics within the dataset.

In certain cases, it may be necessary to apply scaling to datasets, either on the target value or the features, to bring the values within a consistent range (§2.2.1). To do so, we have incorporated a `scaling` module that encompasses various functions to perform different types of scaling, including z-scaling, min-max scaling, and logarithmic scaling. In this

demonstration, our focus is on utilizing the function specifically tailored for scaling the target value. In subsequent notebooks, we will also explore feature scaling.

We recommend starting with this Notebook, as the analysis it includes enables the user to identify potential biases or outliers in the data set, which is critical for developing robust and accurate ML models.

Notebook 2: Regression-Fingerprints

Regression Notebooks are useful when the target values are continuous. For this Notebook we leverage the versatility of fingerprints, which allow for the prediction of various properties, establishing a strong foundation for baseline models. Additionally, it incorporates ensemble learning to enhance predictive performance.

Once all necessary modules (`fingerprints, regression metrics, plotting sklearn, workflow functions, classification metrics, molecules and images`) have been imported the users can load their free of duplicates dataset (created by Notebook 1). The SMILES for the substrates and catalysts will be read and Morgan fingerprints will be generated, alternatively, the user can choose between RDKit fingerprints, MACCS keys, atom pair fingerprints, and torsion fingerprints, from the `fingerprints` module. Bits that correspond to 0 for all molecules will be automatically removed and a concatenated data frame containing fingerprints both for the substrate and the catalyst will be generated. Following, the regression algorithms are trained, and model evaluation is performed, as described in §3.2 (Figure 30).

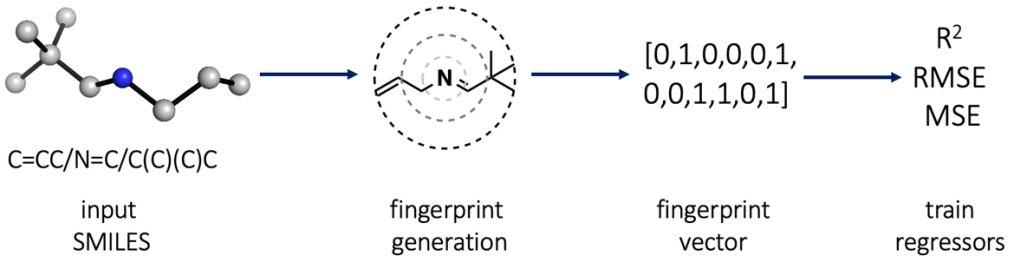


Figure 30: Workflow for generating fingerprints and performing model evaluation. Starting from the SMILES, fingerprints are generated with RDKit and the model is evaluated based on correlation and error.

In this Notebook, we introduce an approach to perform ensemble learning. While the code provided can be applied in various notebooks, we present it exclusively here to mitigate redundancy. Our methodology involves aggregating predictions from individual regressors on the test set, computing their mean values, and subsequently assessing performance metrics by contrasting these means with the corresponding actual values. We emphasize the importance

of discerning and excluding underperforming algorithms, as their predictions may not contribute meaningfully to the overall model.

Notebook 3: Regression-Mordred

In this Notebook, we harness the power of Mordred descriptors due to their ability to offer a diverse and comprehensive set of features, encompassing topological, geometric, and electronic properties. With the Mordred calculator generating over 1,800 descriptors, we seize the opportunity to introduce a feature elimination technique. This technique allows us to streamline and optimize our modeling process by selecting the most relevant descriptors, thereby enhancing the efficiency and interpretability of our predictive models.

To execute it, the user needs to import the Mordred library, along with the previously mentioned libraries and modules. Here, Mordred descriptors are generated for each of the SMILES corresponding to substrates and catalysts. Data cleaning is performed by: (i) dropping descriptors (columns) containing more than 90% of missing values (sometimes the Mordred calculator fails to generate descriptors for all molecules resulting to missing values), and (ii) columns with standard deviation less than 0.5 (as they may exhibit limited variability and thus contribute less valuable information to the analysis).

As not all descriptors will be relevant for the prediction of our target, an important step is to determine which descriptors correlate with the target value. This is done by calling the function `find correlating features` from the `workflow functions` module. This function takes as input the Mordred descriptors and returns a list of features that have a Pearson or Spearman correlation coefficient with the target values greater than or equal to a specified threshold (defined by the user). We suggest using Pearson correlation for continuous data and Spearman correlation for categorical data. The function can also plot the features with high correlation. Additionally, it can process numerical features that Mordred could not calculate for all molecules, by filling missing values with the mean value of the column. If the `significance` parameter is set to `True`, the function also performs a significance test to evaluate whether the correlations between the features and target values are statistically significant, given a significance level of 0.05 (p-value). This is a way to eliminate irrelevant features. A data frame containing only the significant features for the catalysts and substrates is then created. Following this, one-hot encoding for categorical features (features with specific increments such as counts) and min-max scaling for continuous features is applied. Following, the

regression algorithms are trained, and model evaluation is performed, as described in §3.2 (Figure 31).

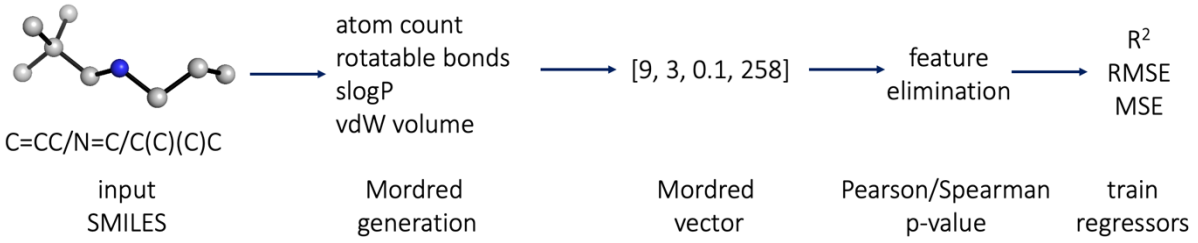


Figure 31: Workflow for generating Mordred descriptors and performing model evaluation. From the SMILES, Mordred are generated, and feature elimination is performed. The model is then evaluated.

Notebook 4: Regression-DFT

This Notebook focuses on training regression models with pre-calculated descriptors. These descriptors can include QM or MD calculated descriptors, or even descriptors calculated with a software not currently integrated in *Pythia*. This demonstrates *Pythia'*s flexibility to accommodate various types of data and highlights its capability to handle diverse descriptor sources. We also take the opportunity to introduce code for performing PCA. By incorporating this functionality, *Pythia* enables users to effectively analyze and interpret complex datasets.

In this study, we employ DFT descriptors which are imported in a *.csv* format, along with the corresponding target values. To reduce the dimensionality of the data and mitigate the effects of collinearity among descriptors, we employ PCA. Figure 32a corresponds to a representative output plot that shows how many components are needed to achieve 95% variance. A table (Figure 32b) is also produced indicating which features compose the PCs. These features are employed to train the models. Finally, the regression algorithms are trained, and model evaluation is performed, as described in §3.2.

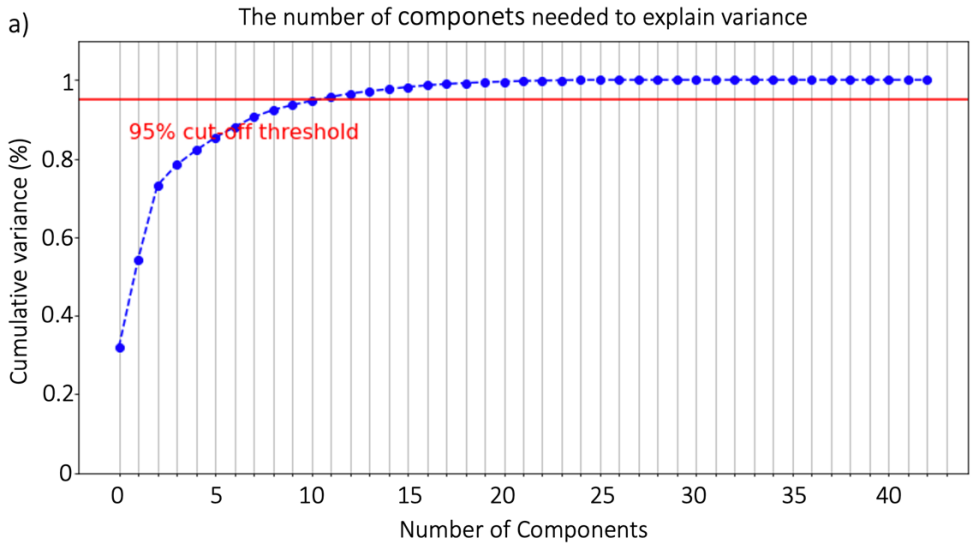


Figure 32: Example of a PCA output. a) On the x-axis the number of components, on the y-axis the variance achieved after each component is introduced. Once the threshold has been reached there is no need for extra components. b) The principal components that contribute to 95% variance sorted according to their contribution.

Notebook 5: Classification- Fingerprints

Classification Notebooks are valuable in scenarios where the target values are discrete or when the target values can be divided into two or three distinct categories. Here, we focus on binary classification. This Notebook demonstrates how feature exploration techniques, tailored for categorical data analysis, can be employed. To do so, we utilize MACCS keys that offer a reduced feature vector, enabling a streamlined approach to feature exploration. However, it is important to note that alternative fingerprinting techniques can also be utilized within the Notebook. These feature exploration techniques empower users to conduct in-depth analyses of patterns and relationships within the data, ultimately enhancing the accuracy and interpretability of the classification models.

Once the MACCS keys are generated, bits containing a value of 0 across all molecules are eliminated and the substrates and catalysts keys are merged into a unified data frame. We use the `describe` function from the pandas library to obtain descriptive statistics for each feature (bit) in the dataset, including count of non-null values, mean, standard deviation, minimum, maximum. These statistics provide insights into the range, spread, and skewness of the data in each column, and can help in identifying any potential outliers or anomalies.

As MACCS keys correspond to categorical data, we analyze the frequency distribution of each feature with respect to the binary class labels. In general, by examining the cross-tabulations, we can identify features that have a stronger association with one class than the other. Figure 33 illustrates this analysis for three features (22, 34, 119) in our dataset. When feature 22 is not

present (0-x axis), more than 40 molecules belong to class 0, while about 70 molecules fall into class 1. However, when the feature is present (1-x axis), only a few molecules fall into class 1, and none fall into class 0. Similarly, when feature 34 is not present (0-x axis), about 35 molecules fall in class 0, and roughly 55 belong to class 1. When the feature is present (1-x axis), 10 molecules belong to class 0, and approximately 19 molecules fall into class 1. Finally, for feature 119, which is always present (only 1-x axis), approximately 45 datapoints to fall in class 0 and about 70 datapoints fall into class 1. Hence, this data point suggests a stronger association with class 1.

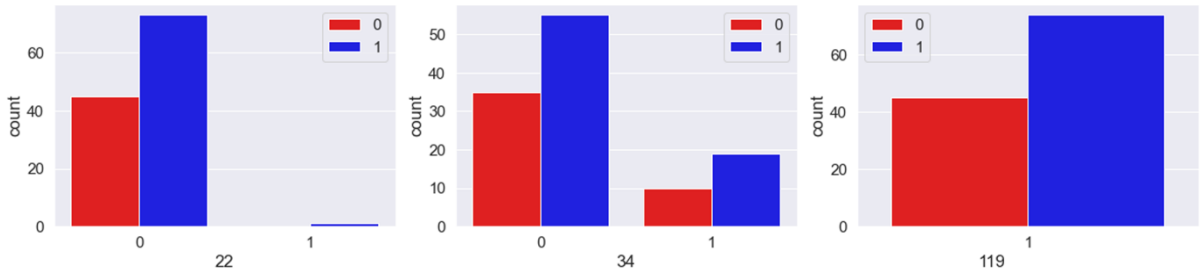


Figure 33: Representation of the frequency distribution of each feature (22, 34, 119) with respect to the binary class labels. For each plot, in the x-axis there are two options, 0 (feature not present) and 1 (feature present). Red represents the data points falling in class 0 and blue the ones in class 1. When feature 22 is not present more points fall in class 1. For feature 34 when the feature is not present more data fall in class 1 at the same time when it is present again more data fall into class 1, this feature needs further investigation. Feature 119 is always present and more data fall into class 1.

The chi-squared statistic is used to measure the degree of association between two categorical variables. The `chi2_contingency` function from the *`scipy.stats`* module is used to compute the chi-squared statistic and p-value. A higher value indicates a stronger association between the two variables, while a small p-value (< 0.05) suggests that the observed association is statistically significant. A new data frame that contains only the features that passed the chi-squared and p-value test is created. As always it is up to the user to decide if they want to perform this feature reduction or if they prefer to skip it and use all MACCS keys as they are generated.

As explained in §3.2 the classes in our dataset are imbalanced therefore, after constructing the feature vector, the SMOTEN algorithm is implemented for the generation of synthetic points. Once the synthetic points data frame is constructed, the classification algorithms are trained, and model evaluation is performed, as described in §3.2.

Notebook 6: Classification- Mordred

In contrast to Notebook 5, this Notebook offers feature exploration techniques tailored for continuous data analysis.

Here, Mordred descriptors are generated and filtered, for the substrates and the catalysts, as described in Notebook 3. It is important to note that Mordred descriptors contain both categorical and continuous features, however, to avoid redundancy, we focus only on the exploration techniques for continuous data. Users can refer to Notebook 5 and selectively incorporate other feature exploration techniques as needed.

The simplest technique for exploring continuous features is to calculate a correlation matrix for the features and create a heatmap to visualize it. Figure 34 shows the heatmap generated for our dataset when 21 Mordred features are considered for both the substrates and the catalysts. The colormap ranges from red (high positive correlation) to white (no correlation) to blue (high negative correlation) helping to detect multicollinearity issues between the features. In this case for example, many of the features are highly correlated (>0.9).

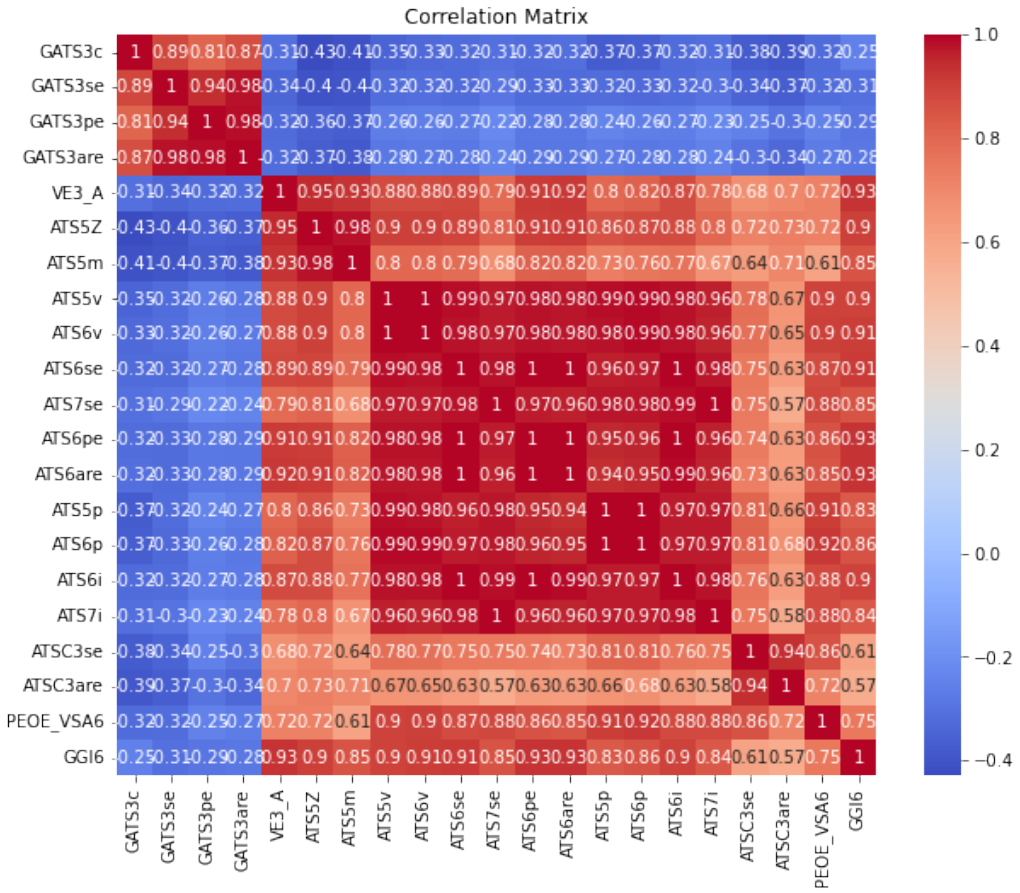


Figure 34: Representation of a correlation matrix as it is produced by the Notebook. Red implies perfect correlation between features, white implies no correlation and blue implies negative correlation.

It is worth noting that high negative correlation (blue) can lead to instability in the estimated coefficients and inflated standard errors, making it difficult to determine whether a predictor is statistically significant. Diagnostic tests such as variance inflation factor (VIF) can assess multicollinearity. VIF is calculated using the `variance_inflation_factor` function from the `statsmodels.stats.outliers_influence` module. The choice of VIF threshold depends on the specific context and goals of the analysis. As a rule of thumb, VIF thresholds of 5 or 10 are used to identify variables that may be contributing to multicollinearity. In practice, the user needs to try several different VIF thresholds (e.g., 2.5, 5, 10) and evaluate the impact of removing variables on model performance. If the goal is to identify a small set of highly predictive variables for a classification task, a more stringent VIF threshold may be appropriate. Alternatively, if the goal is to identify all variables that are significantly associated with the outcome, a less stringent threshold may be more appropriate to avoid potentially biased estimates of the coefficients.

To gain insights into feature distribution across different classes, we generate histograms for the features based on the class labels. First, histograms depicting the overall distribution of the features across all classes are generated (Figure 35a). Next, histograms illustrating the distribution of the features for samples belonging to class 1 (Figure 35b) and class 0 (Figure 35c) are generated. These histograms facilitate a comparison of the distribution of the features between the two classes, which can be useful in identifying features that are particularly important for distinguishing between the classes.

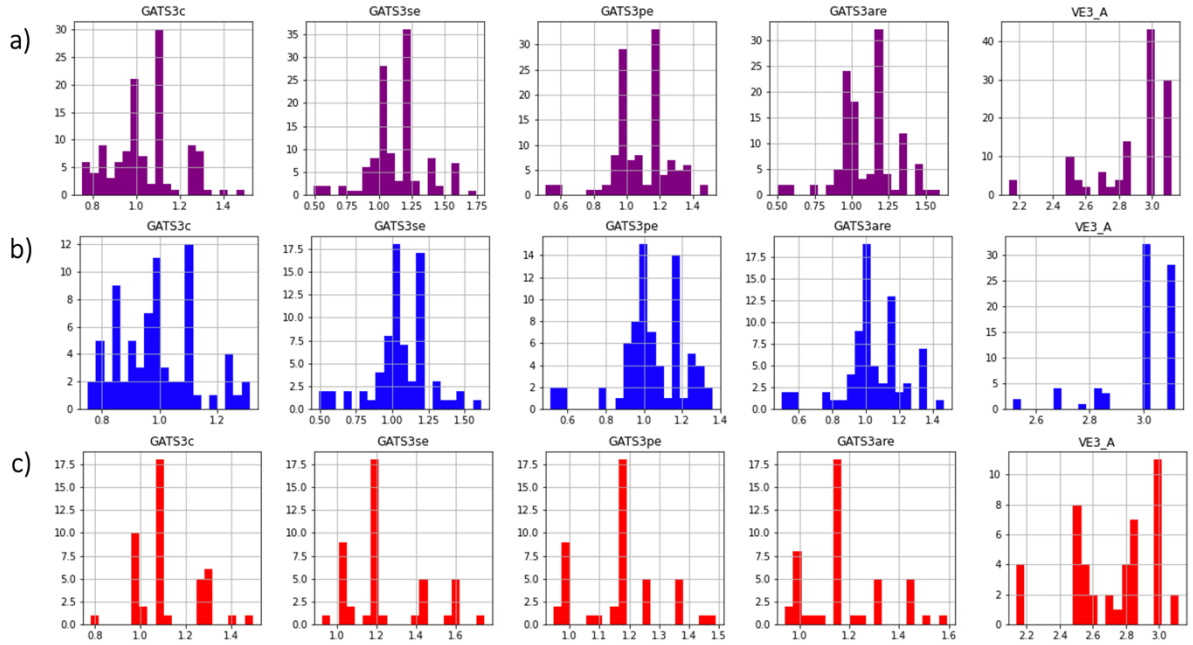


Figure 35: Histograms showings a comparison of the distribution of the features between the two classes for each feature. a) The overall distribution of the features across all classes. b) Histograms illustrating the distribution of the features for samples belonging to class 1. c) Histograms illustrating the distribution of the features for samples belonging to class 0.

As a last step we provide a way to a graphically represent the pairwise relationships between the features, with each feature plotted against every other feature (Figure 36). By incorporating class labels into the plot, it can also aid in identifying any features that are particularly important for distinguishing between the two classes. When scatter plots seem to be following a straight diagonal line, strong linear relationship between two variables is indicated. However, it is important to note that the presence of a linear relationship does not necessarily imply causation between the two variables. There may be other factors or variables that affect the relationship between the two variables, and it is important to consider these factors when interpreting the relationship.

For each feature along the diagonal a kernel density estimation (KDE) plot is also plotted. It is a non-parametric way to represent the distribution of that variable and estimates the probability density function using a Gaussian kernel (i.e., a bell-shaped curve). The height of the curve at a particular point represents the estimated probability density of the variable taking that value. The KDE plot can be useful for detecting any deviations from a normal distribution allowing to compare the distribution of each variable between the two classes (as indicated by the different colors).

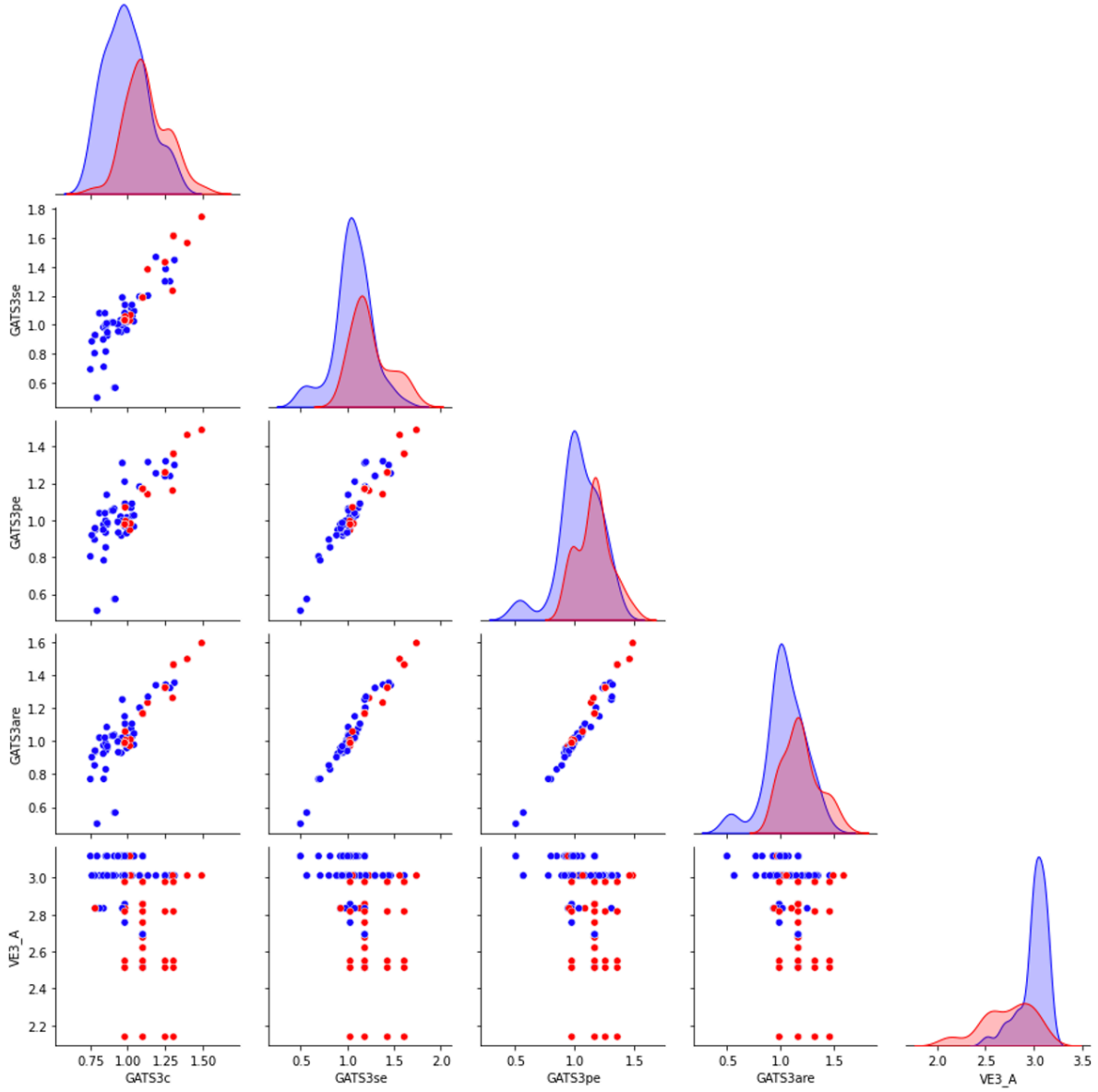


Figure 36: Pairwise relationships between the features, with each feature plotted against every other feature. Along the diagonal the KDE represents the distribution of that variable and estimates the probability density function using a Gaussian kernel. Here, only five features are shown for simplicity.

Based on the investigation described above, users can select which features they want to include in their model and proceed with generating synthetic data using SMOTENC, considering both categorical and non-categorical data.

Once the synthetic points data frame is constructed, the classification algorithms are trained, and model evaluation is performed, as described in §3.2. Here, it is important to scale the data. Scaling the data ensures that all features are on a similar scale, which helps prevent any particular feature from dominating the analysis or influencing the model disproportionately. This can be easily done by setting the scale flag to `True` when calling the `kfold test classifiers with optimization function`.

It should be noted that feature elimination techniques are more effective when used in combination with training, as there is no one-size-fits-all solution to selecting the best features for a given model. Therefore, it is recommended that users experiment with different combinations of feature selection and model training to identify the most effective approach for their specific application.

Notebook 7: Classification- DFT

Similarly to Notebook 4, this Notebook also utilizes pre-processed descriptors. However, it differentiates itself by focusing specifically on providing a comprehensive workflow for performing classification tasks. In addition to the classification workflow, this Notebook offers an approach to leverage SHAP for model interpretation. Interpreting the models is particularly important when working with DFT descriptors, as they often have intuitive physical meanings and can provide valuable insights. This interpretability feature adds a valuable layer of transparency and understanding to the classification models developed.

Once the DFT descriptors and the target values are imported, synthetic data are generated using SMOTE, as DFT descriptors are typically non-categorical. The classification algorithms are then trained, and model evaluation is performed, as described in §3.2.

The SHAP summary plot, provides insights into the feature importance scores. These scores are centered around 0, serving as a reference point for understanding the contribution of each feature. The plot visualizes the relationship between feature values and their impact on the prediction of class 1 (Figure 37). Positive SHAP values indicate a higher association with class 1, suggesting that these features contribute to higher predicted probabilities for this class. Conversely, negative SHAP values are indicative of a stronger association with class 0, as they contribute to lower predicted probabilities. The color gradient employed in the plot enhances the interpretability by encoding the feature values. Red hues represent higher feature values, while blue hues represent lower feature values. This color spectrum aids in identifying the

range and distribution of feature values within the dataset, enabling a comprehensive understanding of their influence on the prediction.

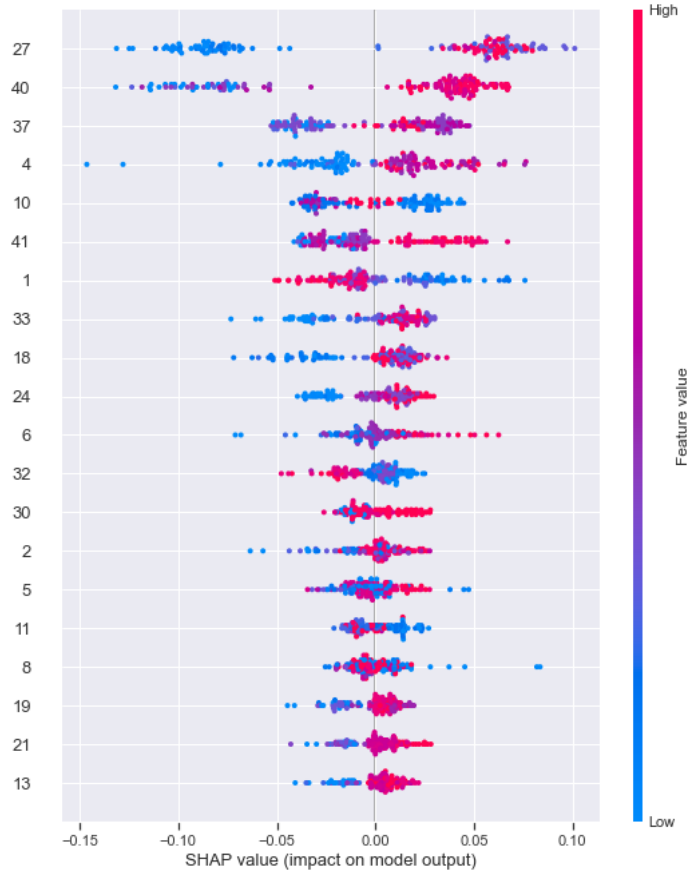


Figure 37: SHAP representation of feature interpretation. In blue features with low values, in red features with high values. On the x-axis the shaply values, centered around 0. On the y-axis the feature index.

Finally, a dependence plot is generated showing the effect a single feature has on the predictions made by the model. In Figure 38 each dot is a single datapoint prediction. The x-axis is the value of feature 13. The y-axis is the SHAP value for that feature. The color corresponds to a second feature (27) that may have an interaction effect with the feature we are plotting (by default this second feature is chosen automatically). Here it is suggested that the effect of feature 27 on feature 13 is not constant and varies nonlinearly. However, the points show distinct clusters that suggest the presence of interaction effects. In this case, the effect of the feature 13 on the prediction depends on the values or interactions of feature 27, leading to non-additive relationships.

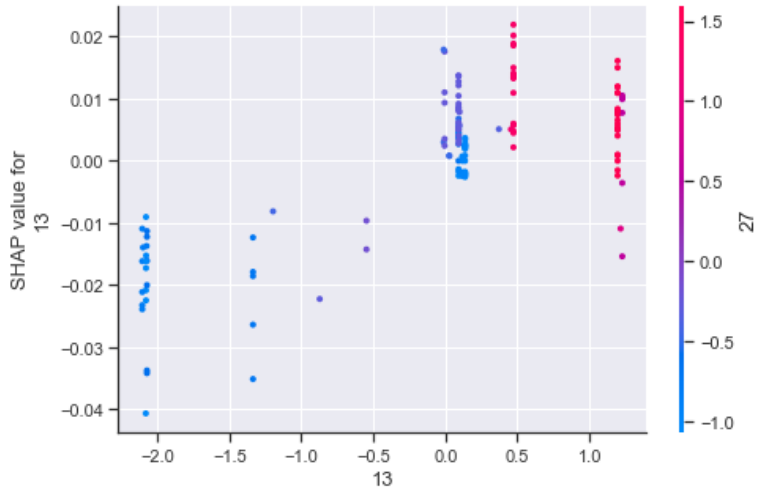


Figure 38: Dependence plot showing the effect a single feature has on the predictions made by the model. The x-axis represents the value of feature 13. The y-axis represents the SHAP value for that feature. The color corresponds to a second feature (27).

Generally, if the dependence plot does not show a clear, linear relationship and instead displays non-linear patterns, distinct clusters, irregular dispersion, or complex interactions, it suggests the absence of independence or additivity between the features, requiring further analysis to understand their combined effect on the prediction. It is essential to exercise caution when interpreting the results of the SHAP analysis. While SHAP offers valuable interpretability, it is worth noting that some classifiers may exhibit complex decision boundaries that are challenging to interpret. Additionally, certain classifiers may produce unreliable or uninterpretable feature importance scores. Therefore, it is crucial to interpret the SHAP analysis results carefully, irrespective of the classifier used.

3.4. Conclusions

In this chapter we introduced *Pythia*, an opensource platform for the development of data-driven predictive ML models, useful particularly when working with small datasets.

Pythia's distinctive features include its high level of automation, flexibility, and ease of use through Jupyter Notebooks, making it an accessible resource for beginners, while also offering experts the possibility to customize the toolkit to their specific requirements. Moreover, the utilization of data elimination techniques, shallow learning algorithms and ensemble modeling, enable users to analyze and interpret the models generated, which is crucial in chemistry applications, where understanding the models often takes a backseat.

While *Pythia* was primarily designed for applications in organocatalysis, its range of applicability has extended to other tasks, such as bioactivity and molecular property

predictions. Looking ahead, we envision the integration of *Pythia* with other open source tools (e.g., Morpheus),[439] and deep learning models. This expansion would unlock the potential for *Pythia* to tackle more complex and diverse datasets, providing improved accuracy and performance. Additionally, the incorporation of multi-objective prediction capabilities would enhance *Pythia's* versatility in scenarios where multiple properties or targets need to be predicted simultaneously. Furthermore, integrating *Pythia* with external chemical databases, would facilitate seamless data retrieval and integration into the modeling workflow. These advancements would solidify *Pythia's* position as a comprehensive and state-of-the-art platform for data-driven predictive ML models in various domains, offering both beginners and experts a powerful tool for their research and analysis needs.

Pythia represents a significant step forward in the application of ML techniques in the field of chemistry. It embodies our belief that a user-friendly, flexible, and interpretable toolkit can foster a broader adoption and deeper understanding of ML.

4. Predicting enantioselectivity with machine learning

The development and implementation of the workflows presented in this Chapter, including DFT calculations, and generation of ML models, as well as chemical interpretation of the results was carried out by myself. Another student in the group, Tom Watts, contributed with some DFT calculations and interpretation of the results in §4.2. A visiting student, Emanuele Casali, performed some of the DFT calculations presented in §4.3 and §4.4, as well as to their analysis.

4.1. Introduction to organocatalysis

Catalysis is the acceleration of a chemical reaction by a substance that is not consumed in the reaction and can therefore continue to act repeatedly. As a result, normally only sub-stoichiometric amounts of a catalyst are required to alter the reaction rate. The catalyst provides an alternative reaction pathway with a lower activation energy, than the non-catalyzed mechanism. Usually, the catalyst forms an intermediate, which then regenerates the original catalyst in a cyclic process.[440–444]

Catalysts can be divided into two categories based on whether the catalyst and reactants are in the same or different phases. Heterogeneous catalysts involve a solid catalyst and liquid or gaseous reactants, which facilitates the separation of the catalysts from the reaction mixture.[445] As a result, they have been widely adopted in the chemical industry[446–449] playing a key role in the synthesis of fertilizers,[450–452] clothing materials,[453–455] and fuels.[456] In the last few decades, this class of catalysts has also been applied to the pursuit of sustainable, eco-friendly energy solutions such as the remediation of atmospheric CO_2[66] and the renewable production of H_2.[459,460]

In homogeneous catalysis, the catalyst exists in the same phase with the reactants.[461] This allows for precise control over catalyst concentrations, and high activity and selectivity is achieved. Additionally, reactions can proceed under milder conditions facilitating easier monitoring through spectroscopy techniques. Some notable examples of homogeneous catalysis include, carbonylation, hydroformylation, liquid phase hydrocarbon oxidation, C-C coupling reactions (Heck and Suzuki), oligomerization, and metathesis and polymerization reactions.[462]

Both heterogeneous and homogeneous catalysts have their respective limitations in catalytic processes. Heterogeneous catalysts can face challenges related to mass transfer limitations, limiting the access of reactants to active sites, and resulting in lower reaction rates. Additionally, selectivity can be an issue due to the presence of multiple active sites, leading to unwanted side reactions and the formation of byproducts.[446–449] On the other hand, homogeneous catalysts suffer from difficulties in catalyst separation and recovery since they are in the same phase as the reactants and products. Catalyst stability can also be an issue due to reactions with other components in the reaction mixture. Furthermore, some homogeneous catalysts may rely on rare or toxic metals, raising environmental concerns.[463]

To overcome the limitations associated with heterogeneous and homogeneous catalysis, researchers have explored the potential of phase transfer catalysis (PTC) as a versatile approach to enable the reaction between molecules located in different phases. Since first reported more than 50 years ago, PTC has been extensively used in academia and industry, particularly in the area of asymmetric synthesis.[464]

Commonly employed PTC catalysts include quaternary ammonium salts or crown ethers. These species possess both hydrophilic and hydrophobic properties, enabling them to act as shuttles to transfer reactants of reactive ions into an organic phase where the reactants are in high concentration. The mechanism of PTC involves the formation of a tight ion pairing for interface crossing. PTC enables the utilization of reactants that are typically insoluble or poorly soluble in a specific solvent. By enhancing the contact between reactants, PTC can significantly improve reaction rates and yields, making it effective for reactions that would otherwise be slow or inefficient. Additionally, PTC can provide selectivity in reactions by controlling the transfer of specific reactants and hence safeguarding sensitive functional groups.[464–467]

Among others, the application of PTC in asymmetric reactions has enabled the synthesis of unnatural amino acids through alkylation of α-imino esters,[468] the asymmetric epoxidation of enones using a chiral ammonium salt and an aqueous solution of sodium hypochlorite as the oxidant,[469] the formation of highly enantioenriched fluorinated β-keto esters by utilizing chiral ammonium salts,[470] and the asymmetric fluorinations of insoluble salts by employing chiral phosphates.[471] Despite extensive investigations, achieving enantioselective C-F bond formation using solubilized fluoride salts with chiral phase-transfer catalysts has proven challenging due to the difficulties in controlling the reactivity of the resulting naked fluoride species.

HB catalysis is a major subdiscipline of organocatalysis. It focuses on the use of small molecule H-bond donors to accelerate the rate of a reaction. HB catalysis is closely related to Brønsted acid catalysis, with the distinction lying in whether the catalyst fully protonates the substrate in the mechanism – a distinction that is not always clear.[472] The H-bond donors can be alcohols, amines, (thio)ureas and phosphoric acids which are typically used to bind an electrophile, lowering its LUMO, and thus increasing its reactivity towards nucleophiles.[473,474] Use of a chiral H-bond donor can facilitate an asymmetric transformation by providing a chiral environment.

A number of approaches employing HB[474,475] and PTC[471,476] separately have been reported. Merging HB and PTC has led to the field of hydrogen bonding phase-transfer catalysis (HB-PTC). Here, HB interactions are used for tuning reactivity and enantioselectivity, while PTC enables the reaction to occur by bringing together otherwise immiscible solvents (Figure 39).[477]

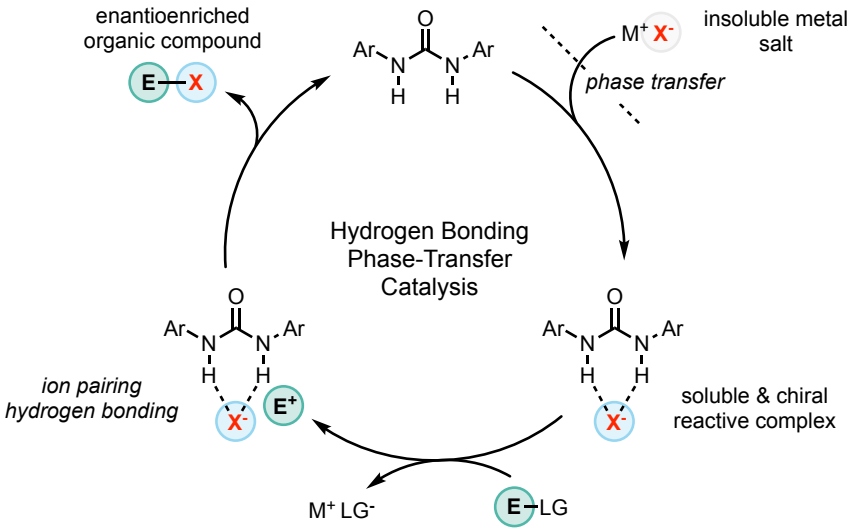


Figure 39: Hydrogen bonding phase transfer catalysis (HB-PTC). Figure adapted from Gouverneur *et al*.[478]

Traditionally, DFT modeling has been employed to characterize and understand the mechanisms of organocatalysis. However, the utilization of DFT poses significant challenges, including the high computational cost associated with accurately describing the complex reaction pathways. Additionally, accurately modeling solvation effects remains a challenging task. These limitations have motivated researchers to explore alternative methods for predicting selectivity and reactivity in such systems.[479] As discussed in §1.3, ML techniques have emerged as promising tools for tackling these challenges. ML models can capture complex relationships between reactant structures and reaction outcomes, enabling efficient prediction of selectivity in organocatalysis. The application of ML in the field offers a more cost-effective

and time-efficient approach compared to traditional DFT methods, opening new avenues for accelerating catalyst discovery and optimization.[479]

In this chapter, we will employ ML to investigate three distinct systems: the enantioselective formation of β-fluoramines, the Strecker synthesis of α-amino acids, and the Pictet-Spengler cyclization of hydroxylactams. Our objective is to demonstrate that ML can predict selectivity in these systems and provide valuable chemical insights.

4.2. Enantioselective formation of β-fluoroamines

Small molecules bearing fluorine on a stereogenic carbon are ubiquitous in the agrochemical and pharmaceutical industry, with as many as 35% of agrochemicals and 20-25% of marketed drugs, containing one or more fluorine atoms. C-F bonds can enhance the biological, chemical, and physical properties of organic molecules by increasing solubility, metabolic stability, and bioavailability.[480,481]

Incorporating fluorine into a molecule at a late stage, so-called "late-stage fluorination", allows an intermediate to be diversified by fluorination at various positions. An application where late-stage fluorination is essential is for ^{18}F positron emission tomography (PET).[482,483] PET is a medical imaging technique used to aid diagnosis. A radionuclide, most commonly an $^{18}F^-$ labeled tracer molecule, is injected to the patient and allows location of the PET tracer in the body from coincident gamma rays emitted by positron-electron annihilation events. Due to the relatively short half-life of ^{18}F (110 minutes), the radioisotope must be incorporated into an already elaborated tracer molecule, before purification and administration to the patient.[484] However, the selective introduction of a fluorine at this later stage remains a challenge due to the difficulties in controlling reactivity and selectivity.[485]

Current methods for fluorination can be divided into those that use electrophilic ("F^+") or nucleophilic ("F^-") sources of fluorine (Figure 40). The latter includes the low-cost alkali metal salts (CaF_2), which are promising alternatives to more reactive and less safe sources, but suffer from poor solubility.[486,487]

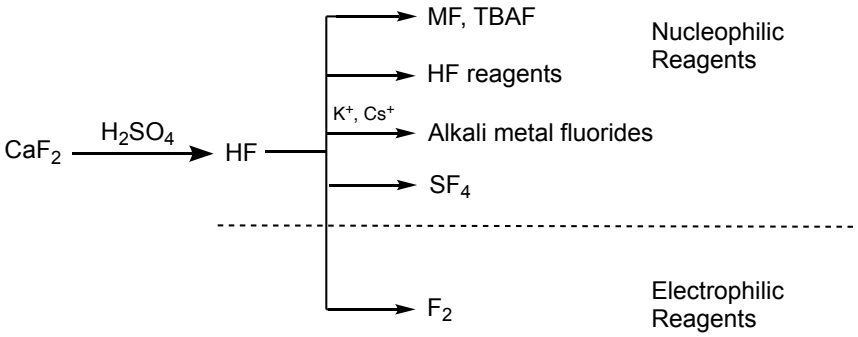
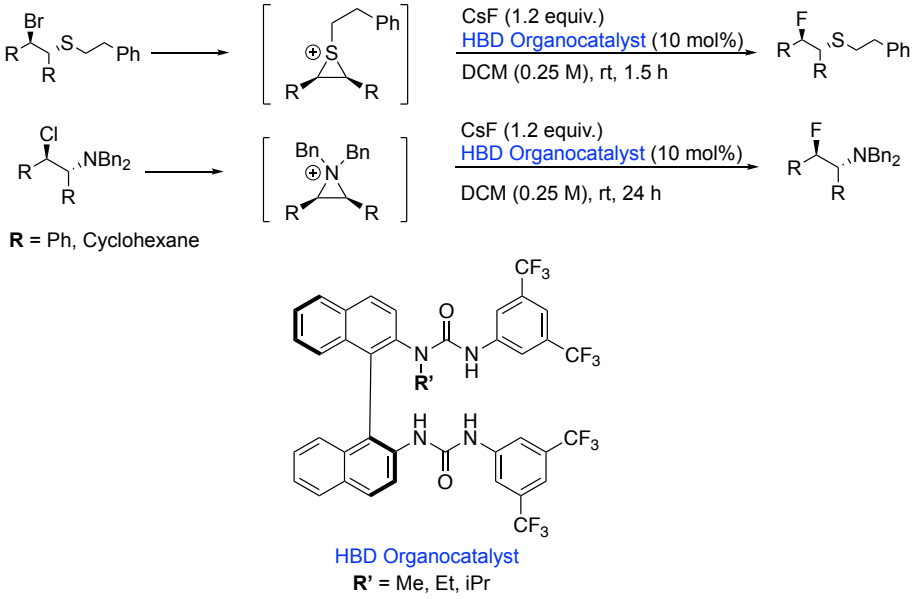


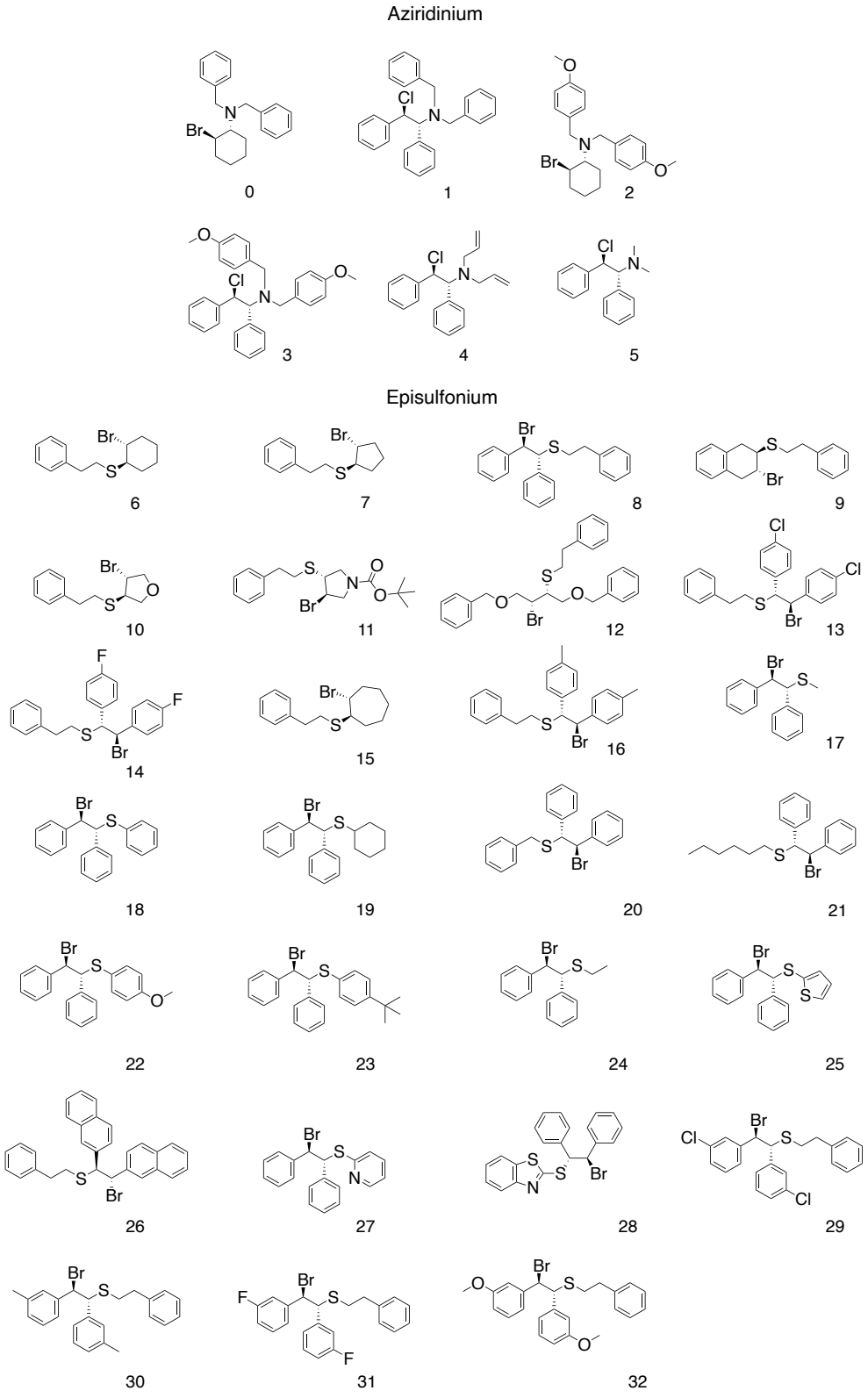
Figure 40: The origins of fluorine. Fluorine derives from fluorite, with varying degrees of processing.[486]

In 2018, Gouverneur and co-workers employed HB-PTC to afford enantioenriched fluorinated compounds using cheap and readily available CsF and KF sources. They developed urea-based catalysts which, through HB, enhanced the availability and controlled the reactivity of fluoride towards episulfonium and aziridinium intermediates (Scheme 3).[488,489] Through this process they converted racemic stilbene-derived *β*-haloamines and sulfides to the corresponding enantioenriched fluorinated products.

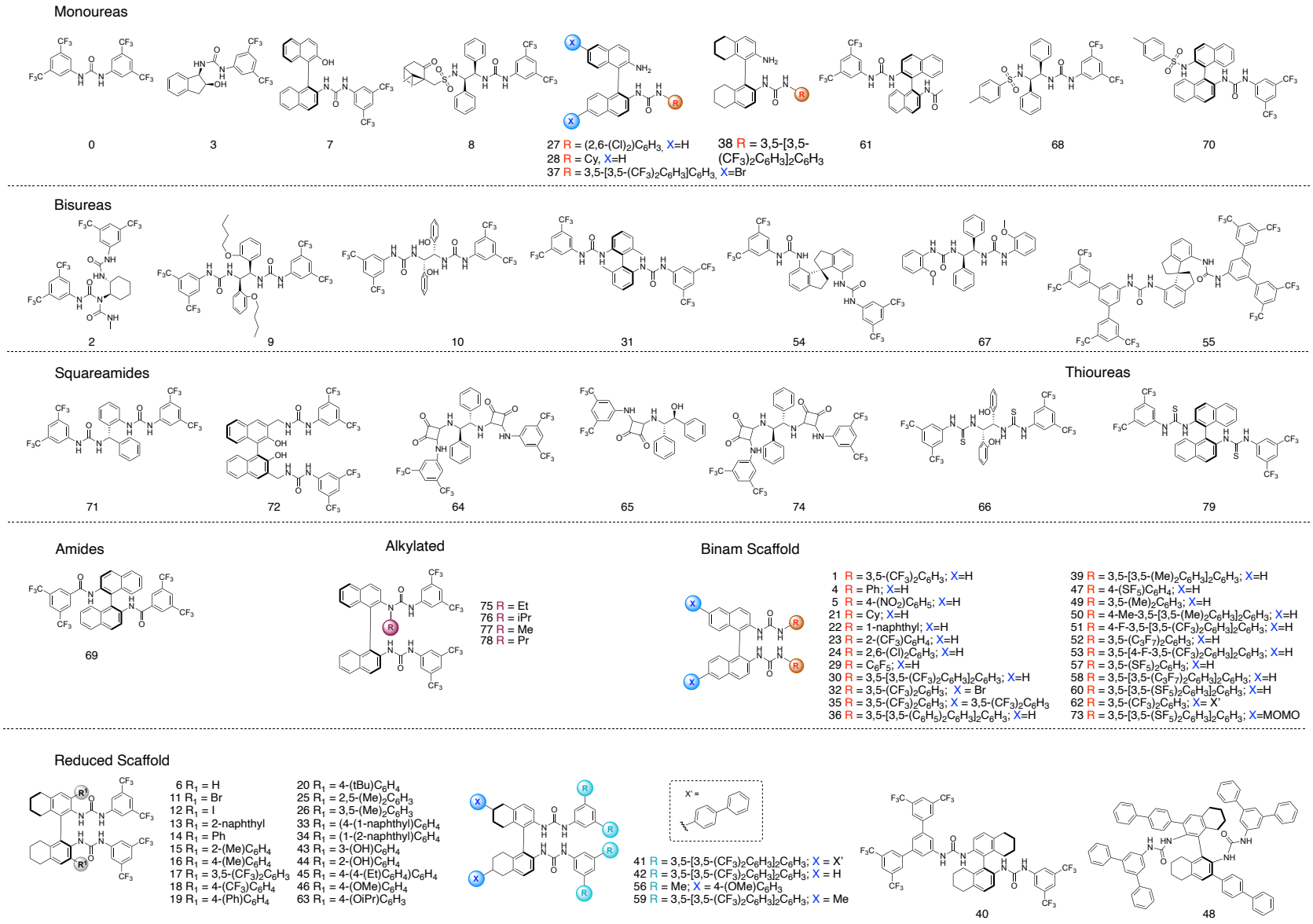


Scheme 3: The reaction under study is a urea-based catalyzed ring opening.

The Gouverneur group shared with us experimental data for 33 substrates and 80 catalysts (Scheme 4 – Scheme 5), leading to 257 experimentally tested reactions. The reactions were tested at various temperatures, ranging from -35 to 25 ˚C, and two different solvents, dichloromethane (DCM) and difluorobenzene (DFB). The experimental work in this chapter was performed by members of the Gouverneur group, Dr. Gabriele Pupo, Dr. Francesco Ibba, Dr. Anna Chiara Vicini and Dr. Lukas Pfeifer.



Scheme 4: Experimentally tested substrates used by the Gouverneur group.



Scheme 5: Experimentally tested catalysts.

Although the substrates and catalysts investigated share a common structural core, their diverse substitution results in a diverse chemical space, as evident by the average *Tanimoto* similarity value of 0.6. As shown in Eq. 2.11, this metric can range from 1 (identical molecules) to 0 (two dissimilar molecules). This can also be graphically seen by performing PCA on the Morgan fingerprints. Both for the catalysts (Figure 41 - left) and the substrates (Figure 41 - right), the plots exhibit distinct cluster formations; however, the clusters are not densely populated. This indicates that compounds within each cluster share certain similarities or common characteristics, facilitating their grouping. Nevertheless, diversity is observed within each cluster, suggesting that compounds within the same cluster may possess distinct structural features or variations in chemical properties.

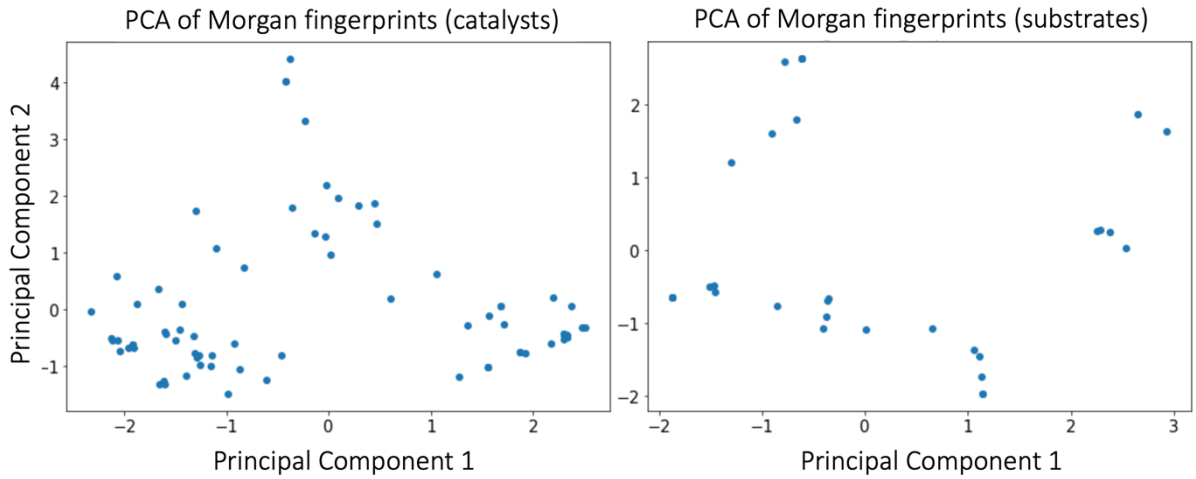


Figure 41: PCA analysis plotting the chemical space with Morgan fingerprints for the catalysts and the substrates. While clusters can be observed not all of them all densely populated implying structural differences within the dataset.

The experimental entioselectivity values (in enantiomeric ratio, *e.r.*), reported in literature[488,489] or shared to us by the Gouverneur group, are converted to their corresponding $\Delta\Delta G^{\ddagger}$ using the following relationship (Eq. 4.1)[490]:

$$\Delta\Delta G^{\ddagger}\left(\frac{kJ}{mol}\right) = -RT\ln(e.r.) \qquad (4.1)$$

where *e.r.* is the enantiomeric ratio, T is the temperature (K) at which the reaction was performed, and R is the gas constant (8.3145 J/K·mol). Upon converting the data to $\Delta\Delta G^{\ddagger}$, no further scaling is necessary, as the values already span in a range from 0 kJ/mol to 8.6 kJ/mol. However, as depicted in Figure 42, it is evident that the majority of the target values are situated in proximity to 0, while only a few extend beyond 7 kJ/mol. This observation indicates that our dataset includes a limited number of successful experiments, which may pose challenges when

predicting high values. Nevertheless, it suggests that predicting unfavorable values should not be a significant challenge at the very least.

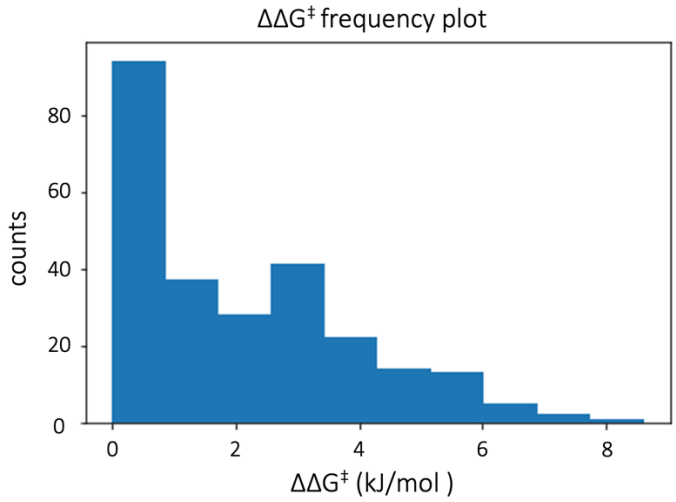


Figure 42: Histogram plot showing the occurrences of the target value. Only a few data points have a high $\Delta\Delta G^{\ddagger}$, while the majority is situated around 0.

4.2.1. Computational workflows

This section describes the process of generating a ML model to predict the $\Delta\Delta G^{\ddagger}$ of the nucleophilic fluorination for both episulfonium and aziridinium substrates. Moreover, it aims to infer underlying reactivity of these reactions by analyzing the ML model.

DFT calculated descriptors and Morgan fingerprints are used to characterize the data set. As discussed in §2.2, determining the most appropriate descriptors to accurately represent a reaction is not always straightforward. Hence, multiple models (nine) incorporating different descriptors, at different levels of theory, were constructed (Table 4). In all models, the substrates are treated independently of the catalyst-F^- complex, excluding any consideration of the substrate-catalyst-F^- system. The descriptors calculated for the substrates, and catalyst-F^- complex, and information regarding the solvent and the temperature for each reaction are combined into a unified reaction matrix (Figure 43). To represent the solvents, we employ a binary system, where DCM is denoted as 0, and DFB is denoted as 1. All models were run 100 times, each time the training set and the test set were split randomly (90% of the data consist the training set and 10% of the data consist the unseen test set). In all 100 runs LASSOCV and 10-fold cross validation was used. The important descriptors that arise after LASSOCV are calculated on the entire data set.

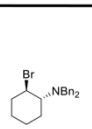
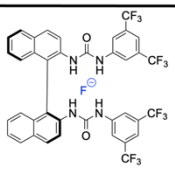
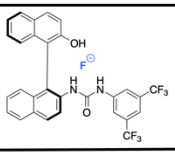
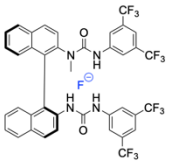
Reactant	Catalyst	Solvent	Temperature	Reactant	Catalyst	Solvent	Temperature	Reaction
		DCM	-35	[0,1,0,...0,1] +	[0,0,1,...0,0] +	0 +	-35	= [0,1,0,...0,0,0,-35]
		DCM	25	[0,1,0,...0,1] +	[0,0,0,...0,0] +	0 +	25	= [0,1,0,...0,0,0,25]
		DFB	25	[1,0,0,...0,1] +	[0,0,1,...1,0] +	1 +	25	= [1,0,0,...1,0,1,25]

Figure 43: Graphical representation of the generation of a reaction matrix. Information about the substrates and the catalysts-F^- complexes are combined into one reaction matrix, along with information for the solvent and temperature.

Table 4: Summary of the models investigated in this section. In the columns Substrates and Catalyst-F^- the level of DFT used is explained or the use of fingerprints. In the column Solvent model, it is identified if implicit solvent was used during the calculations. In column Complex generation we report the way the initial catalyst-F^- complex was generated and whether conformational sampling was performed. Finally in the column Descriptors we report if the descriptors used were averaged or not.

Model	*Substrates*	*Catalyst-F⁻*	*Solvent*	*Complex generation*	*Descriptors*
1	fingerprints	fingerprints	N/A	N/A	N/A
2a	fingerprints	**A**	Yes	fit to core	analytical
2b	fingerprints	**A**	Yes	conformational	analytical
3a	**B**	**A**	Yes	fit to core	average
3b	**B**	**A**	Yes	conformational	analytical
3c	**B**	**A**	Yes	conformational	average
4	**B**	**C**	Yes	conformational	average
5a	**B**	**B**	No	conformational	analytical
5b	**B**	**B**	No	conformational	average
A	PBE-D3BJ/def-TZVP//PBE-D3BJ/def2-SVP				
B	PBE-D3BJ/def2-SVP				
C	PBE-D3BJ/ma-def-TZVP//PBE-D3BJ/ma-def2-SVP				

Model 1

Model 1 was generated by utilizing Morgan fingerprints for the substrates and catalysts, resulting in a feature set of 2,048 variables. Subsequently, bits with a constant value of 0 across all reactions were excluded, resulting in a reduced set of 292 informative features.

It is important to note that there were some challenges with the recognition of certain catalysts (e.g., catalysts **57**) using RDKit. However, fingerprints were still generated for these catalysts. Therefore, the outcomes derived from this model should be interpreted with caution, considering the potential limitations associated with catalyst representation.

Model 2a

Model 2a was generated using Morgan fingerprints for the substrates, and physical chemical descriptors for the catalyst-F^- complexes. To calculate the catalyst descriptors, we follow the workflow shown in Figure 44.

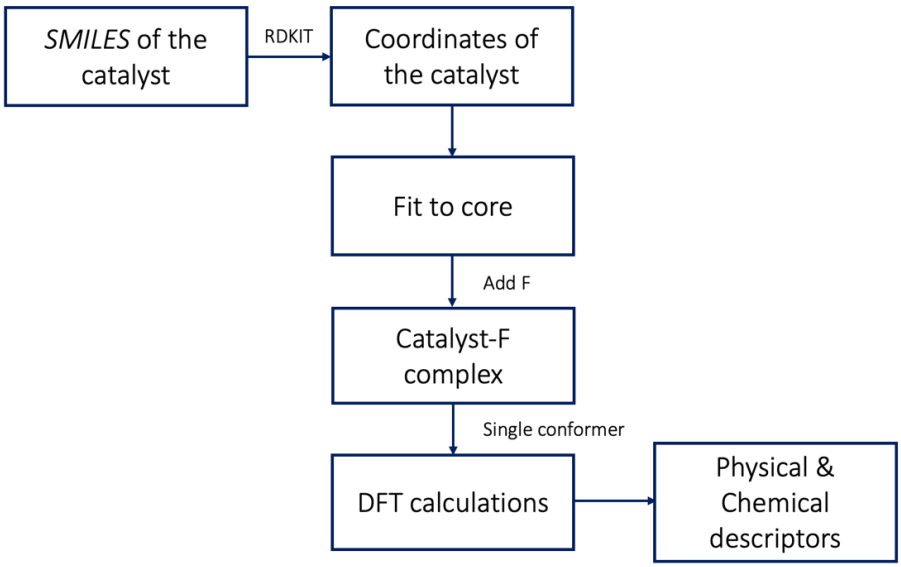


Figure 44: Workflow for obtaining parameters for each catalyst.

The crystal structure of catalyst **1** (Scheme 5) was used as a template, where the core is defined by the atoms marked in blue and salmon in Figure 45. These cores were used as a reference to fix the coordinates of the catalysts. In total, 48 catalysts were fitted to the two-urea core (Figure 45a) and 21 catalysts were fitted to the one-urea (Figure 45b). Catalysts that did not contain urea motifs or contained many atoms in their structures, making it difficult to fit to either core (11 catalysts), were handled manually. The fluoride ion was added to these structures, and the

complex formed was used as an input for electronic structure calculations using the ORCA (*v.4.1.1* package).[283]

Geometry optimizations were performed at the PBE-D3BJ/def2-SVP level of theory[255,278] and RI- JCOSX[282] was applied. Solvent effects were accounted for with the SMD solvent model[276] with parameters appropriate for DCM. Single point energy calculations on the optimized geometries were carried out at the SMD(DCM)-PBE-D3BJ/def-TZVP level of theory. This level of theory was chosen for its generalizability and good tradeoff between accuracy and computational cost.[264,265] Catalysts that did not converge were excluded from the models, leaving 247 reactions to be investigated further.

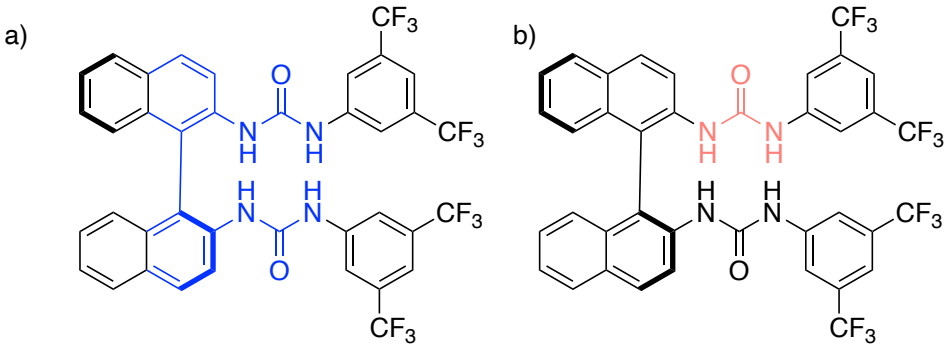


Figure 45: The coordinates of the catalysts were fitted to the coordinates of a template catalyst. a) Two-urea core marked in blue, b) One-urea marked in salmon.

From the electronic structure calculations, five types of descriptors were calculated, leading to up to 45 features per catalyst (Table 5). Those include 1) HOMO and LUMO energies, 2) dipole moment, 3) ^{1}H, ^{13}C and ^{19}F NMR shifts for the urea moiety and the fluoride, 4) BO of the HB interactions between the fluoride and the urea hydrogen atoms (BO_{HF}), BO of the nitrogen-urea hydrogen atoms (BO_{NH}), and BO of the nitrogen-urea carbon atoms (BO_{CN}) (Figure 46) and 5) atomic charges of the fluoride and the urea hydrogens.

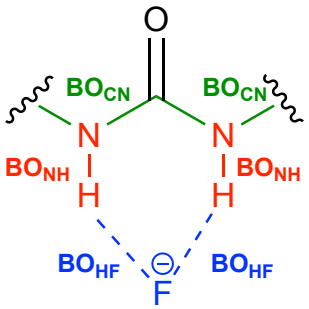


Figure 46: Representation of the calculated bond orders (BO). In blue the fluoride-urea hydrogen atoms (F^{-}-H(N)), referenced here as BO_{HF}. In red the urea N–H bonds (H-N(CONH)), references here as BO_{NH}. In green the urea N–C bonds (C-N(CONH)), referenced here as BO_{CN}.

Table 5: Catalyst descriptors extracted from ORCA.

Catalyst's Descriptors	*Method*	*Units*
NMR shifts	GIAO	ppm
Atomic charges	Hirshfeld	e
Bond Order (BO)	Mayer	-
Dipole Moment	-	Debye
HOMO/LUMO	-	eV
Steric parameters	*Sterimol*	Å

Finally, steric descriptors we added. They are calculated as explained in Appendix B2. For these catalysts the bonds scanned are shown in Figure 47, these include the N-C bonds of the urea moiety and the N-R bond of the urea (marked in maroon). In total, 15 sterimol type descriptors are used.

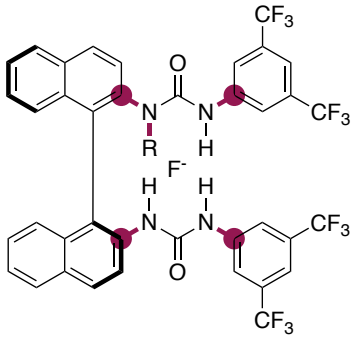


Figure 47: Bonds along which sterimol type descriptors are calculated.

As the numbers of atoms differ between the catalyst scaffolds, the number of descriptors varies for each system, for instance a monourea catalyst has only two BO_{HF} whereas a bisurea catalyst has four BO_{HF}. This inconsistency in the number of descriptors used for each complex, was solved by representing missing features by 0. In total 52 features describing the catalysts arise from this method. This alignment was performed manually, and it is an exceptionally time-consuming step. An alternative solution was to take into consideration the average values of the features (average NMR shifts for each atom, the average charges for the hydrogen atoms of the urea groups, and the summation of the BO descriptors). This latter approach is used for Models 3a, 3c, 4 & 5b and employs 27 descriptors for the catalysts.

Model 2b

This model follows the workflow described in Model 2a with the difference that conformational sampling is performed to ensure that the lowest energy catalyst-F^- complex is considered for the generation of the descriptors. This was done using GFN2-xTB (*v.6.2*).[285] Each complex was subjected to simulated annealing. A threshold of RMSD > 1 Å was used to

identify unique structures; conformers where the anion is not accessible to the electrophiles were also excluded. For the remaining conformers, single point energies at the PBE-D3BJ/def2-SVP level of theory were calculated, and Boltzmann weighted. Only the conformers that contribute up to 90% of the population were optimized at the SMD(DCM)-PBE-D3BJ/def2-SVP level of theory. For those systems, the lowest energy conformation was subjected to a single point energy calculation at the SMD(DCM)-PBE-D3BJ/def2-TZVP level of theory (Figure 48). The descriptors extracted are the same as in Model 2a (Table 5).

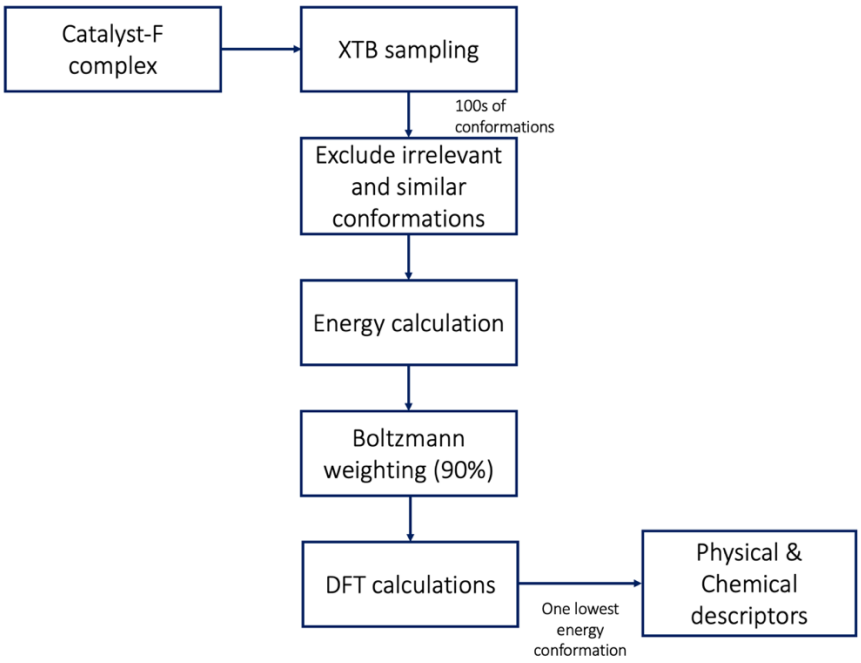


Figure 48: Workflow for obtaining parameters for the lowest energy conformation of each catalyst.

Model 3a, 3b, and 3c

Model 3a relies fully on physical chemical descriptors for both substrates and catalyst-F^- complexes. The geometries for the catalyst were taken from Model 2a, where the catalysts were fitted to a core. Descriptors for the lowest energy substrates, after simulated annealing, were calculated at the SMD(DCM)-PBE-D3BJ/def2-SVP level of theory. The descriptors extracted from the substrates are presented in Table 6. For the catalysts the same types of descriptors were extracted as before however this time we used the average NMR shifts for each atom type, the average charges for the hydrogen atoms of the urea groups, and the summation of the BO descriptors.

Table 6: Substrate descriptors extracted from ORCA.

Descriptors	*Method*	*Units*
Atomic Charges	Hirshfeld	e
Dipole Moment	-	Debye
HOMO/LUMO	-	eV
Steric parameters	*Sterimol*	Å

Model 3b uses the catalyst geometries from Model 2b and calculates analytical descriptors (not averaged). The previously calculated descriptors from Model 3a are used for the substrates. Model 3c uses the catalysts and substrates geometries from Model 3b however it uses average descriptors for the catalysts.

Model 4

Model 4 follows Model 2b and performs simulated annealing, however it uses the SMD(DCM)-PBE-D3BJ/ma-def2-SVP level of theory to optimize the catalyst conformers that contribute up to 90% of the population after the Boltzmann weighting. For those systems, the lowest energy conformation was subjected to a single point energy calculation at the SMD(DCM)-PBE-D3BJ/ma-def-TZVP level of theory. In this model, the average values of the NMR shifts (per atom type), the average values of charges (per atom type) and the summation of the values of BOs (per bond type), are extracted. Physical chemical descriptors for the substrates are included as described above.

Model 5a - Model 5b

Model 5a was generated as a faster alternative compared to the methodologies described above. In this model, conformers for the catalyst-F^- complexes are obtained as described in Model 2b, performing conformer sampling, and single point energies at the PBE-D3BJ/def2-SVP level of theory were calculated. Descriptors are extracted from the lowest energy conformation of each catalyst, after this energy calculation. No geometry optimization takes place. Physical chemical descriptors for the substrates are included as described above. Model 5a, uses analytical values of the NMR shifts, the charges, and the BOs. Model 5b uses the same geometries as Model 5a however it uses average descriptors for the catalysts.

4.2.2. Results and discussions

In the following paragraphs, we evaluate the performance of the different models. Our focus is on Model 5b, for which we analyze the most important descriptors derived from LASSOCV. For the remaining models, we do not provide chemical interpretation, as they are not explored further. Their data are included in the supplementary material (subdirectory Chapter 4.2). A comprehensive summary of the performance metrics for all models can be found in Table 7. By thoroughly investigating multiple models and considering both the interpretability and predictive performance, our study highlights the trade-offs and strengths associated with different modeling approaches.

Table 7: A summary of all the models and their metrics presented in this section.

	Train set		*Test set*	
Model	RMSE (kJ/mol)	R^2	RMSE (kJ/mol)	R^2
1	0.41	0.95	0.75	0.80
2a	0.78	0.83	0.89	0.75
2b	0.77	0.83	0.85	0.76
3a	0.93	0.75	1.04	0.67
3b	0.97	0.74	1.03	0.70
3c	0.94	0.75	1.01	0.69
4	0.95	0.74	1.05	0.66
5a	1.00	0.71	1.04	0.68
5b	1.01	0.71	1.03	0.67

It is important to note that while the correlations achieved by the models are not particularly high, the corresponding errors are remarkably low. Evaluating the robustness of a model is better determined by examining the error metrics rather than relying solely on correlation. The R^2 value serves as a relative measure of fitness, whereas RMSE provides an absolute measure of fitness. In this regard, RMSE proves to be a more reliable metric for comparing and assessing the performance of different models. This perspective is shared by Winkler *et al.*,[491] who emphasize that the usefulness of a model is better determined by its RMSE rather than relying solely on the value of R^2. The lower the RMSE, the better the model's predictive accuracy and its ability to capture the underlying trends and patterns in the data.

Model 1

Model 1 yields high correlations and low errors both for the training ($R^2 = 0.95$, RMSE = 0.41 kJ/mol) and the test set ($R^2 = 0.80$, RMSE = 0.75 kJ/mol, Figure 49). While models based on fingerprint descriptors can be generated within minutes and with minimal computational cost, they can be hard to interpret. Often the same bits of a fingerprint represent different sets of atoms. In our data set where most of the catalysts are highly symmetric, it would be impossible to identify which part of the catalyst is represented by specific bits. This led us to the conclusion that Model 1 can be a cheap and easy approach to identify the behavior of a data set if the desired outcome is a prediction. However, if one would like to explain and interpret the underlying chemistry, we do not suggest a model based solely on fingerprints.

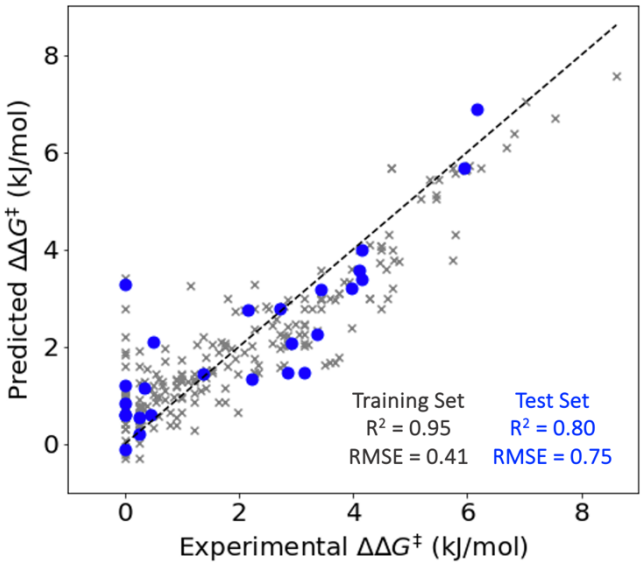


Figure 49: Model 1 – Fingerprints for both the substrates and the catalysts. In grey the training set (90% of the data) shows good correlation (R^2) and low error (RMSE < 1kJ/mol). In blue the unseen test set (10% of the data) confirms the predictive power of the model.

Model 2a

This model consists of 156 descriptors of those only 50 descriptors were selected by LASSOCV. Model 2a yields good correlations and low errors both for the training set ($R^2 = 0.83$, RMSE = 0.78 kJ/mol) and the test set ($R^2 = 0.75$, RMSE = 0.89 kJ/mol, Figure 50).

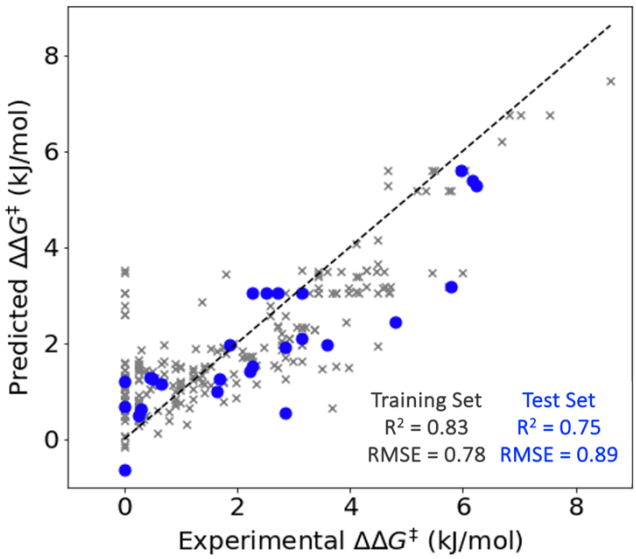


Figure 50: Model 2a - Fingerprints for the substrates and physical chemical descriptors of a fitted conformation for the catalysts. In grey the training set (90% of the data) shows good correlation (R^2) and low error (RMSE < 1kJ/mol). In blue the unseen test set (10% of the data) confirms the predictive power of the model.

Model 2b

The same descriptors are considered for this model as for Model2a; and it yields similar results for the training (R^2 = 0.83, RMSE = 0.77 kJ/mol) and the test set (R^2 = 0.76, RMSE = 0.85 kJ/mol, Figure 51). This is something we expected to see, as the conformations of the catalysts are not that different and it indicates that conformational sampling is not necessary, as fitting the catalyst to a crystal structure already provides the most chemically relevant information.

Model 2a has the advantage that no additional computational time is needed to identify the lowest energy conformation. However, it is general practice to use the lowest energy conformation of a given structure, additionally crystal structures are not always available. As the purpose of this project is to identify the most cost effective and accurate model, we suggest that if the crystal structure is known, fitting the structures to this geometry can serve as an alternative to searching for the lowest energy conformation, as the two models perform equally well.

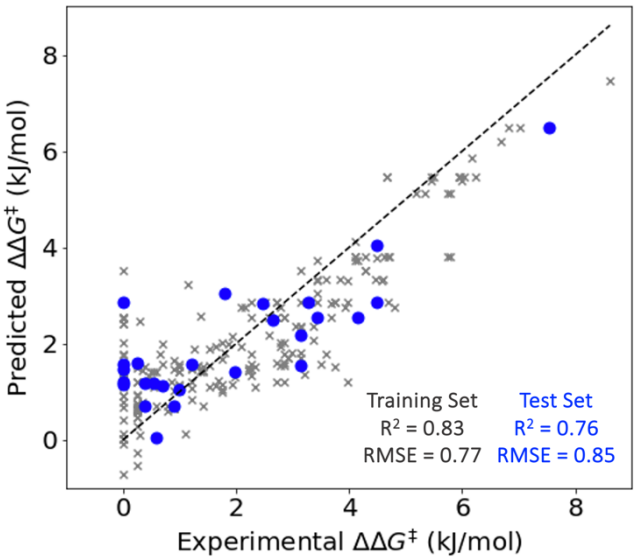


Figure 51: Model 2b - Fingerprints for the substrates and physical chemical descriptors of the lowest energy conformation for the catalysts. In grey the training set (90% of the data) shows good correlation (R^2) and low error (RMSE < 1kJ/mol). In blue the unseen test set (10% of the data) confirms the predictive power of the model.

Model 3a

From the 42 descriptors generated for this model, following LASSOCV, only 15 remain. Model 3a yields moderate correlations and low errors both for the training set (R^2 = 0.75, RMSE = 0.93 kJ/mol) and the test set (R^2 = 0.67, RMSE = 1.04 kJ/mol, Figure 52). We see that as LASSOCV considers less features the model straggles to keep high correlations, yet the errors are still small.

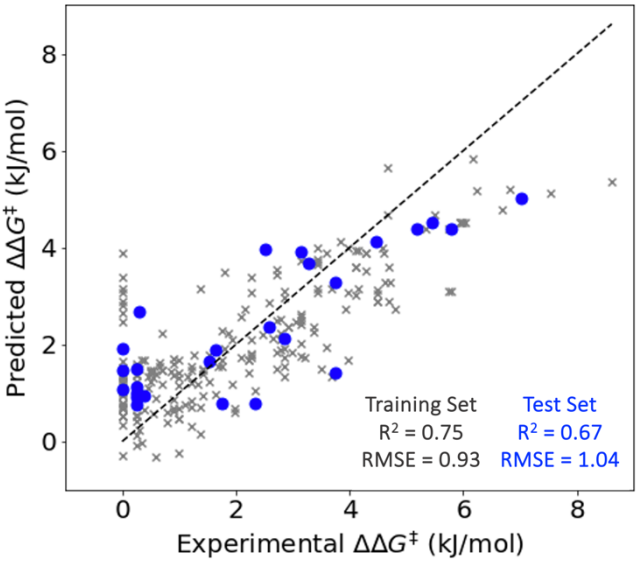


Figure 52: Model 3a - Physical chemical descriptors of the lowest energy conformation for the substrates and physical chemical descriptors of a fitted conformation for the catalysts. In grey the training set (90% of the data) shows good correlation (R^2) and low error (RMSE ≈ 1kJ/mol). In blue the unseen test set (10% of the data) confirms the predictive power of the model.

Model 3b

From the 67 descriptors generated for this model, following LASSOCV only 18 remain. Model 3b yields very similar results to Model 3a, with good correlations and low errors both for the training set ($R^2 = 0.73$, RMSE = 0.97 kJ/mol) and the test set ($R^2 = 0.70$, RMSE = 1.03 kJ/mol, Figure 53). It would be best to not draw any definitive conclusions with such small differences in the R^2 and RMSE. In principle the small differences could be attributed to the features considered by LASSOCV.

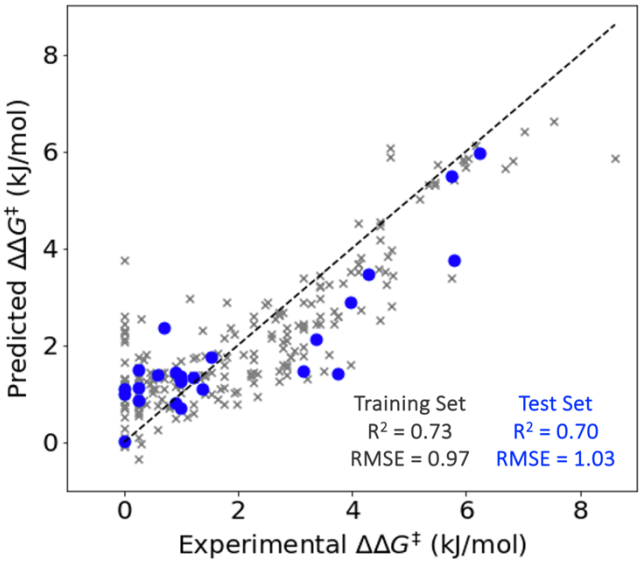


Figure 53: Model 3b - Physical chemical descriptors of the lowest energy conformation for the substrates and physical chemical descriptors of the lowest energy conformation for the catalysts. In grey the training set (90% of the data) shows good correlation (R^2) and low error (RMSE ≈ 1kJ/mol). In blue the unseen test set (10% of the data) confirms the predictive power of the model.

Model 3c

From the 42 descriptors generated for this model, following LASSOCV only 15 remain. Model 3c yields good correlations and low errors both for the training set ($R^2 = 0.75$, RMSE = 0.94 kJ/mol) and the test set ($R^2 = 0.69$, RMSE = 1.01 kJ/mol, Figure 54).

At this stage, it is evident that all three variations of Model 3 yield similar results. This finding is encouraging because it suggests that there is no need to calculate analytical descriptors, which require much manual work for their alignment, as the average descriptors suffice and perform equally well.

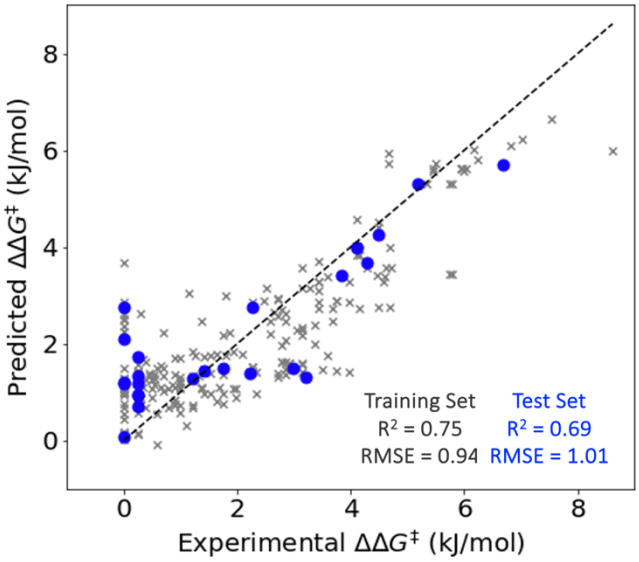


Figure 54: Model 3c - Physical chemical descriptors of the lowest energy conformation for the substrates and average physical chemical descriptors of the lowest energy conformation for the catalysts. In grey the training set (90% of the data) shows good correlation (R^2) and low error (RMSE ≈ 1kJ/mol). In blue the unseen test set (10% of the data) confirms the predictive power of the model.

Model 4

From the 42 descriptors generated for this model, following LASSOCV only 14 remain. Model 4 yields good correlation and low error for the training set (R^2 = 0.74, RMSE = 0.95 kJ/mol) but slightly lower correlation for the test set (R^2 = 0.66, RMSE = 1.05 kJ/mol) compared to previous models (Figure 55). This result shows that descriptors calculated at a higher level of theory do not necessarily yield better results.

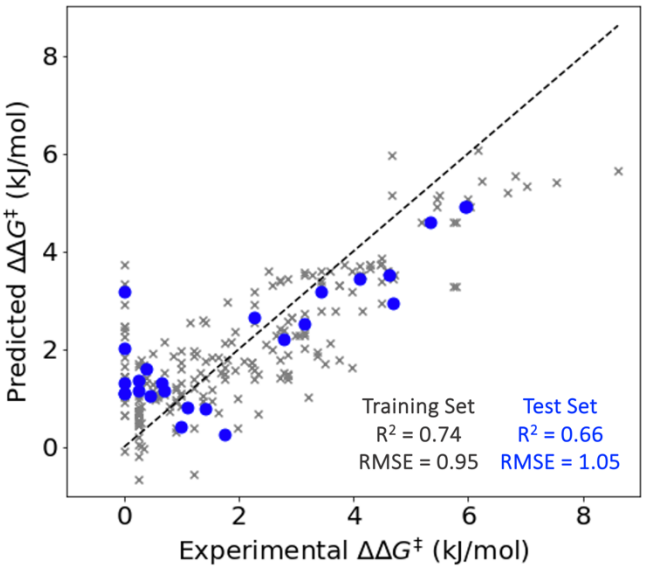


Figure 55: Model 4 - Physical chemical descriptors of the lowest energy conformation for the substrates and average physical chemical descriptors of the lowest energy conformation for the catalysts. In grey the training set (90% of the data) shows good correlation (R^2) and low error (RMSE ≈ 1kJ/mol). In blue the unseen test set (10% of the data) performs slightly worse, with lower correlation and higher error.

Model 5a

From the 67 descriptors generated for this model, following LASSOCV only 17 remain. Model 5a yields moderate correlations and low errors both for the training (R^2 = 0.71, RMSE = 1.00 kJ/mol) and the test set (R^2 = 0.68, RMSE = 1.04 kJ/mol, Figure 56). This model is considerably cheaper than the previous models and still yields similar accuracies.

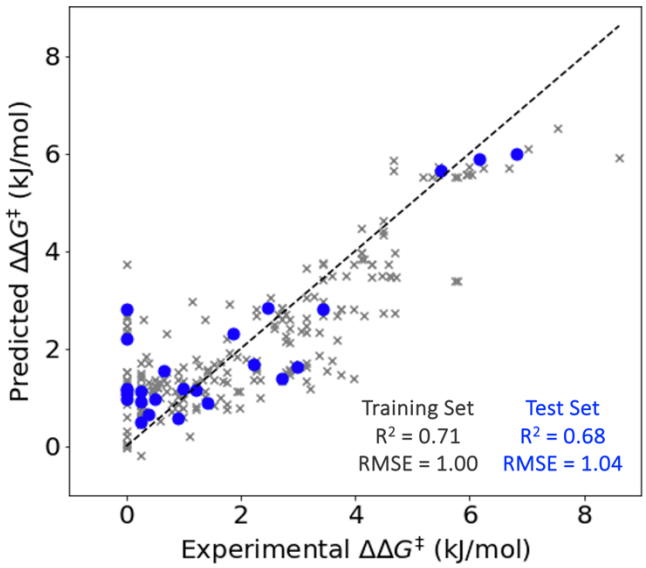


Figure 56: Model 5a - Physical chemical descriptors of the lowest energy conformation for the substrates and physical chemical descriptors of the lowest energy conformation without optimization for the catalysts. In grey the training set (90% of the data) shows moderate correlation (R^2) and low error (RMSE ≈ 1kJ/mol). In blue the unseen test set (10% of the data) confirms the predictive power of the model.

Model 5b

In contrast to Model 5a, this model uses average descriptors, 13 of which represent the substrates, 27 represent the catalysts and one binary descriptor is used to represent the solvent and one for the temperature. LASSOCV takes under consideration only 15. Model 5b yields moderate correlations and low errors both for training (R^2 = 0.71, RMSE = 1.01 kJ/mol) and the test set (R^2 = 0.67, RMSE = 1.03 kJ/mol, Figure 57), showing that the average descriptors are sufficient to describe the reaction.

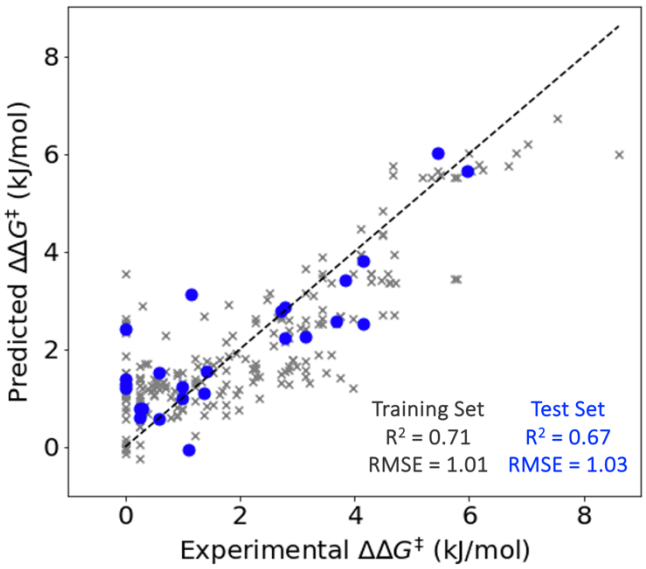


Figure 57: Model 5b - Physical chemical descriptors of the lowest energy conformation for the substrates and average physical chemical descriptors of the lowest energy conformation for the catalysts. In grey the training set (90% of the data) shows moderate correlation (R^2) and low error (RMSE ≈ 1kJ/mol). In blue the unseen test set (10% of the data) confirms the predictive power of the model.

Based on the low computational cost of this model and most importantly the low error, we selected it as our final model. We therefore investigated it further and we embarked on identifying the chemical features that may affect selectivity. To do so we analyzed the most relevant coefficients arising from LASSOCV (Eq. 4.2). Here, the descriptors in blue represent the substrates, in black the catalysts:

$$\Delta\Delta G^{\ddagger} = 0.66 + 1.25\mathrm{Solv} + 0.01\mathrm{Temp} + 0.59N_2R(L) + 0.54NC_1(B_1) + 0.28N_4C(B_1) + 0.25N_2R(B_1) + 0.12N_2C(B_5) + 0.09N_1C(B_1) + 0.07N_2R(B_5) + 0.06NC_1(L) + 0.02NC_2(L) - 0.08LUMO_{\mathrm{sub}} - 0.11N_2C(B_1) - 0.5HOMO_{\mathrm{sub}} - 0.61H_{\mathrm{charge}} \quad (4.2)$$

Eq. 4.2 illustrates the inclusion of the most significant descriptors and their corresponding coefficients. It is important to note that these coefficients are dependent on the actual values that the descriptors can assume. While the model's training process involves normalization of the descriptor values, the subsequent analysis is conducted using the real values. This approach allows for a more comprehensive understanding of the importance and impact of the descriptors, as a large coefficient alone does not necessarily imply a highly influential descriptor.

Firstly, the $Solv$ parameter is present, we remind the reader that DFB is represented by 1 (DCM is represented by 0) and corresponds to larger $\Delta\Delta G^{\ddagger}$ according to the optimized experimental conditions reported by the Gouverneur group, our model seems to be able to capture this

information. The temperature is also present, with a small coefficient showing that the model recognizes the changes in temperature.

The steric descriptors L and B_1 around the C2 and C1 carbons of the substrates are found to affect the selectivity positively (Figure 58a). These descriptors indicate the relative sizes of each substituent, with bulkier moieties leading to higher *ee*s. Furthermore, the steric parameters at the C2 position differentiate between cyclohexyl and stilbene derived, with the later outperforming cyclohexyl systems. This can be explained by conformational flexibility in stilbene substrates, which reduce steric repulsion between the protecting groups and the backbone. Furthermore, the presence of stabilizing pi-pi interactions between stilbene substrate and catalyst (shown in Figure 58b-c) are absent in the cyclohexyl substrate, which unequivocally lacks the two aromatic cores necessary to establish them.

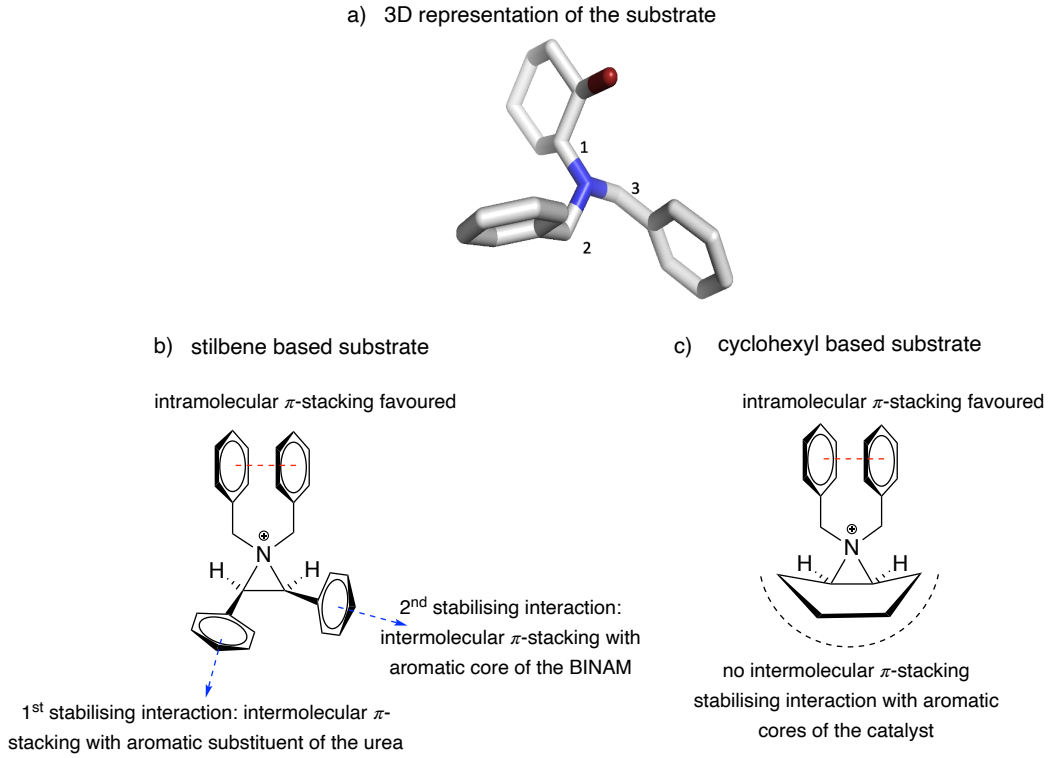


Figure 58: Representatives of the substrates under study. a) A 3D representation noting the carbon numbering according to Eq. 4.2. b) Interactions between stilbene and c) cyclohexyl based substrates.

The HOMO and LUMO of the substrates are two other descriptors that affect selectivity according to our model. As they decrease, the selectivity increases. Indeed, stilbene substrates, which present lower HOMO and LUMO compared to their cyclohexane counterparts, generally outperform their cyclohexyl equivalent in terms of selectivity. Therefore, we believe HOMO and LUMO to be a proxy for the type of substrate backbone.

In Figure 59 we show a representative catalyst and the numbering around the carbon, nitrogen, and hydrogen atoms for clarity. The BOs calculated between the atoms of interest, are shown with the magenta arrows. In spheres the sterimol type values are marked, to calculate them we follow the direction of the magenta arrows as well.

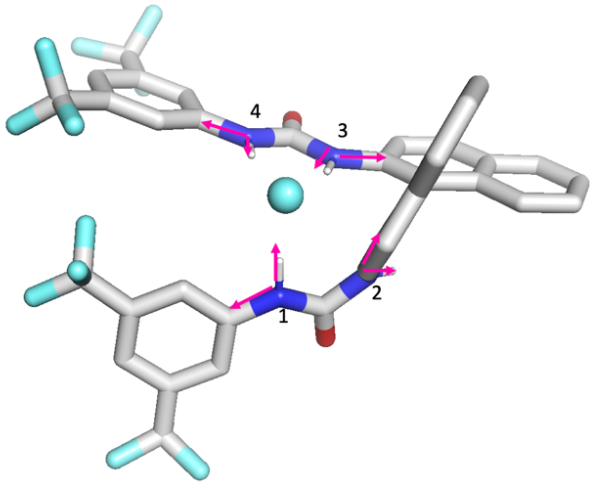


Figure 59: Representative catalyst and atom numbering for reference according to Eq. 4.2. In pink the direction for the calculation of steric descriptors and the BOs.

Eight catalyst descriptors are found to be important, seven steric descriptors and the average H_{charge} of the NH urea groups. The coefficient of the H_{charge} is negative, keeping in mind that the actual value of an H_{charge} is negative, higher absolute values of H_{charge} lead to increased selectivity. Catalysts forming three or four HBs with the fluoride show higher selectivities, compared to the monourea catalysts where only two HBs are formed. At the same time the values of L, B_1 and B_5 for the N_2R steric descriptor (R being alkyl groups) show that bulkier alkyl groups in this position are more advantageous for selectivity, allowing only three HBs to coordinate with the fluoride. This substitution pattern is particularly important as it affects the geometry of the catalyst and the way the orbitals interact during the nucleophilic addition of the fluoride to the substrate. At the same time the charge of the F^- is regulated with HB interactions making it a better nucleophile. All the above are consistent with experimental findings.[492]

Lastly larger values for B_1 steric descriptor for the N_1C and N_4C, suggest that bulky groups on the left-hand side of the molecule favor selectivity (Figure 59). At the same time the presence of the B_1 steric descriptor for both the N_1C and N_4C shows that symmetric catalysts are preferred, which is consistent with experimental findings as catalysts that are not symmetric perform poorly. The negative sign of the $N_2C(B_1)$ descriptor suggests that substitutions on the 3-3' position will reduce selectivity. The prevalence of steric descriptors suggests that a model

generated solely from steric descriptors could perform equally well. Therefore, we explored an additional model that only uses steric descriptors.

Steric Descriptors Model

As expected, the predictive power of this model is equivalent to Model 5b (Figure 60). Here the LASSOCV analysis considers 14 descriptors, including 7 steric descriptors for the catalysts, 5 descriptors for the substrates, the solvent and temperature.

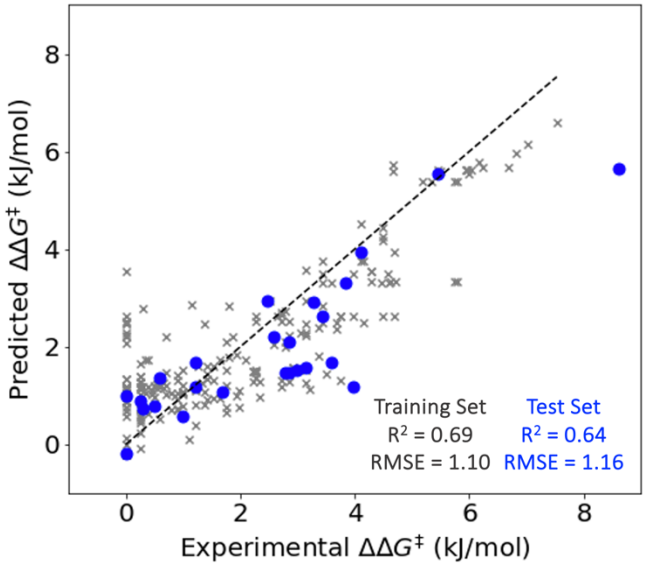


Figure 60: Model 5b-sterics. Physical chemical descriptors of the lowest energy conformation for the substrates and steric descriptors for the lowest energy conformation for the catalysts. In grey the training set (90% of the data) shows moderate correlation (R^2) and low error (RMSE ≈ 1kJ/mol). In blue the unseen test set (10% of the data) confirms the predictive power of the model.

Descriptors associated with the bulkiness of alkyl groups around the urea moiety $N_2R(L)$, $N_2R(B_1)$, $N_2R(B_5)$ have positive sign, consistent with the findings in Model 5b (Figure 61a). $N_1C(B_1)$ and $N_4C(B_1)$, which have the same value for symmetric systems, represent substitutions on the left-hand side of the molecule (Figure 61b). These descriptors indicate the bulkiness of the substituent group at this position and agree with the $N_2C(B_5)$ descriptor, which represents the overall size of the catalyst. It suggests that bulky groups on that side of the catalyst will increase $\Delta\Delta G^{\ddagger}$. Finally, the negative sign of the $N_2C(B_1)$ descriptor suggests that substitutions on the 3-3' position will reduce selectivity as Model 5b suggested (Figure 61c). Experimental findings confirm a significant drop in selectivity when bulky groups are used as substituents on the 3-3' position.

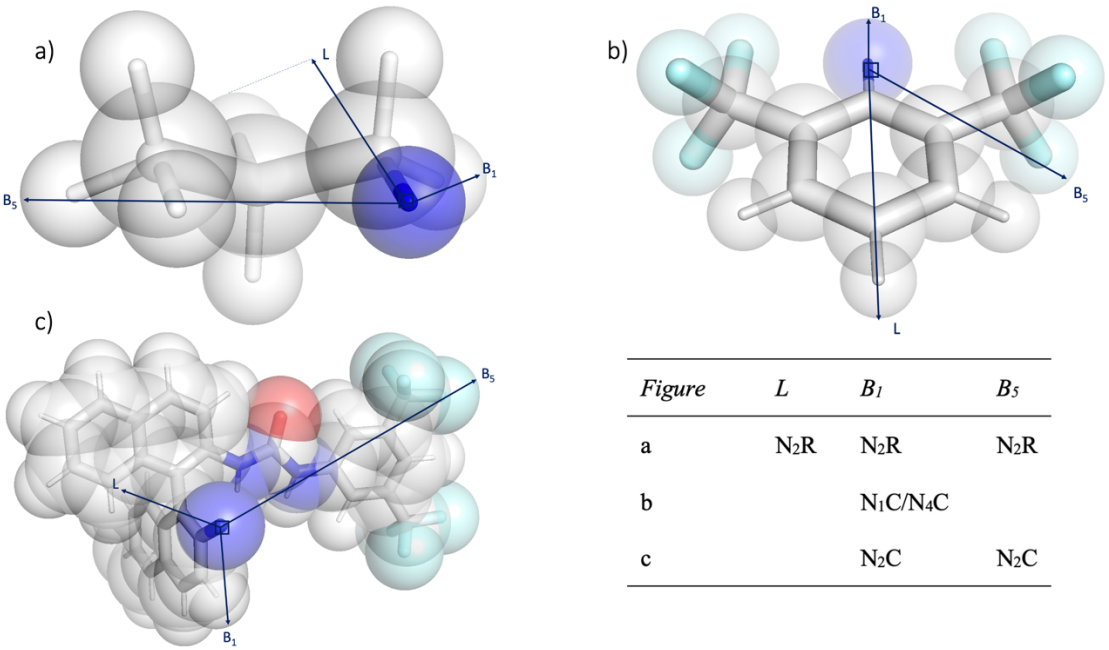
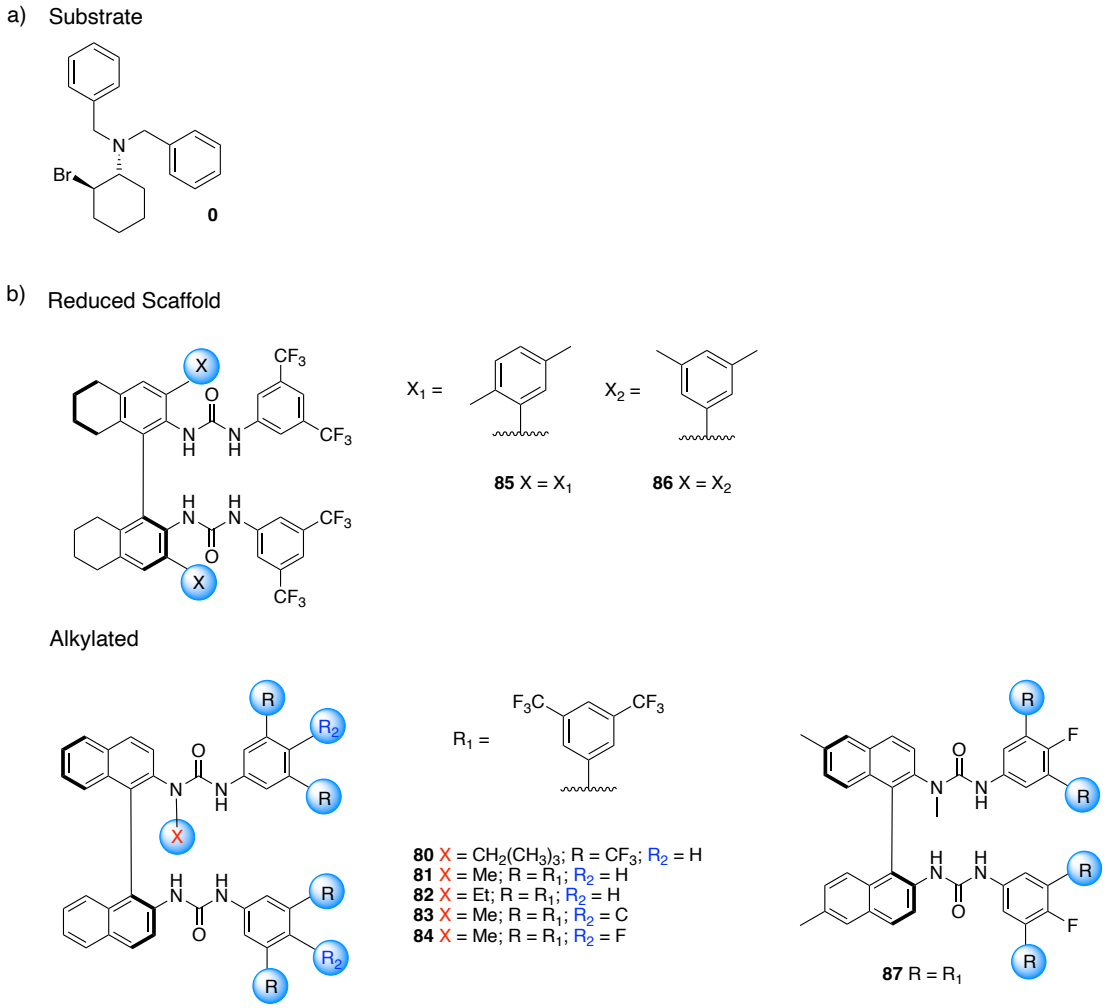


Figure	*L*	*B1*	*B5*
a	N_2R	N_2R	N_2R
b		N_1C/N_4C	
c		N_2C	N_2C

Figure 61: Important steric descriptors identified for the catalysts. a) Sterics for the alkyl groups at the urea moiety. b) Sterics for the groups used to substitute N1 and N4, when the catalysts are symmetric the values of L, B1 and B5 are the same, which favors selectivity. c) Sterics when the 3-3' position is substituted, when the B1 value is big (therefore the position is substituted) the selectivity drops.

To conclude this investigation, we examined the extrapolation ability of Model 5b to a completely unseen dataset of eight catalysts and one substrate (Scheme 6). The experimental study was conducted by Dr. Anna Vicini, after generating the ML models. The reactions were performed in DCM and room temperature. In Figure 62 we present the performance of the test set. Model 5b successfully extrapolate to this unseen dataset, demonstrating improved performance with a high correlation coefficient and significantly reduced errors (R^2 = 0.80, RMSE = 0.76 kJ/mol). One key factor contributing to the improved performance of the model is the utilization of the entire original dataset. By incorporating all available data points into the training set we were able to provide the models with a robust foundation for learning patterns and making accurate predictions.



Scheme 6: External data set. a) Substrate under study. b) Catalysts under study.

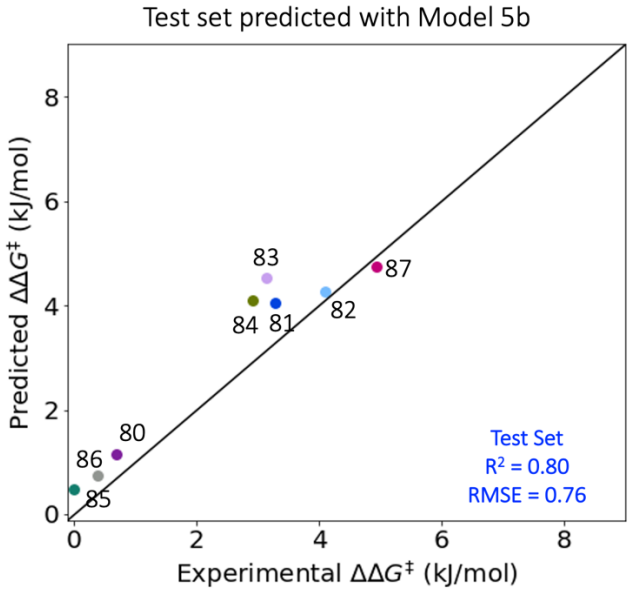


Figure 62: The predictive model that arises from Model 5b for the external validation set. Each data point is represented by a different color. In order the colors that correspond to the catalysts are: 80: purple, 81: blue, 82: sky-blue, 83: levander, 84: olive-green, 85: blue-green, 86: grey, 87: magenta.

To assess the potential over-fitting of our model, we performed y-randomization as a final step. This technique ensures that the model's predictive power is unbiased. Similarly, to the models before, 100 runs were conducted and the average metrics were analyzed, with the model exhibiting no predictive power for the test sets (R^2 = 0.39, RMSE = 1.60 kJ/mol). This decline in predictive capabilities during y-randomization further supports our confidence in the robustness of the original model.

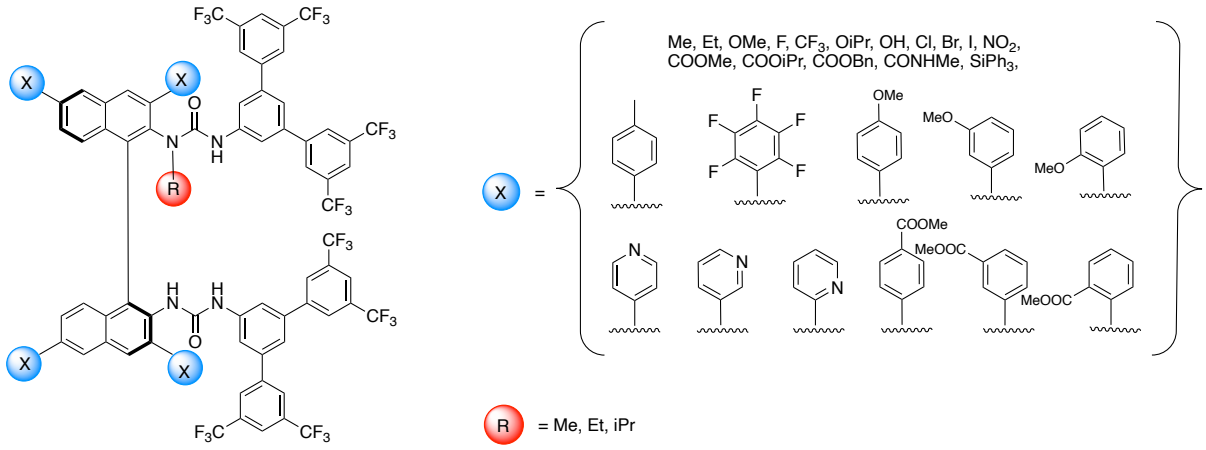
Conclusions

In summary, we systematically developed and evaluated nine ML models to predict selectivity in the enantioselective formation of β-fluoroamines. The models differ in the type of descriptors and levels of theory employed. Fingerprint-based approaches were computationally efficient to generate and exhibited greater predictive power. However, their limited interpretability posed challenges in gaining insights into the underlying factors driving the predictions. Notably, the predictive performance of Model 4, which employed a higher level of theory DFT calculated descriptors, was not better than the other models. On the other hand, Model 5b, which is computationally more efficient, was found to be robust, despite displaying moderate correlation during training and it was able to extrapolate to unseen data points. These findings emphasize that models trained at a higher level of theory do not necessarily translate into superior predictive capabilities.

4.2.3. Investigating novel catalysts

The results obtained using Model 5b with a completely new dataset, sparked our interest in taking this study a step further. Our aim was to identify a novel catalyst that in combination with substrate **0** (Scheme 6), would result to higher *ee*, under the conditions considered before, room temperature and DCM solvent.

We generated 244 catalysts by substituting the 3-3’ (162 catalysts) and 6-6’ positions (82 catalysts) with the X groups in Scheme 7, and the urea N with alkyl (R) groups in Scheme 7. 3,5-bis(trifluoromethyl)phenyl were used on the opposite side of the BINAM as according to the important descriptors that arise from LASSOCV, they are preferred for higher *ee* (Scheme 7).



Scheme 7: Modifications made to generate a dataset of 244 novel catalysts.

These catalysts were generated using the `molfunc` function in autodE,[493] where the functionalization is performed by replacing a monovalent atom in the xyz input file for a fragment (provided in a SMILES format). For each of the generated structures, energy minimization is performed with purely rigid body rotations, to avoid clashes. After all the structures were generated, the *ee* was predicted using Model 5b.

The 3-3' modifications did not yield high $\Delta\Delta G^{\ddagger}$ values (highest $\Delta\Delta G^{\ddagger}$ = 5.51 kJ/mol, Figure 63a). This is consistent with our previous discussion where we explained that the negative sign of the $N_2C(B_1)$ descriptor suggests that substitutions on the 3-3' position will reduce selectivity (Figure 61a).

The 6-6' modification proved more promising with the highest $\Delta\Delta G^{\ddagger}$ = 9.06 kJ/mol (Figure 63b). Catalysts with electron withdrawing groups on the BINAM are predicted as the highest scoring catalysts (Figure 64). Indeed, electron withdrawing groups have been experimentally proven to be better catalysts, as they increase the acidity of the urea and consequently the strength of the HBs. Different levels of alkylation at the urea nitrogen yield negligible changes on the $\Delta\Delta G^{\ddagger}$. Such differences are unfortunately within our model's error and therefore strict assumptions as to which modification should be preferred should not be made.

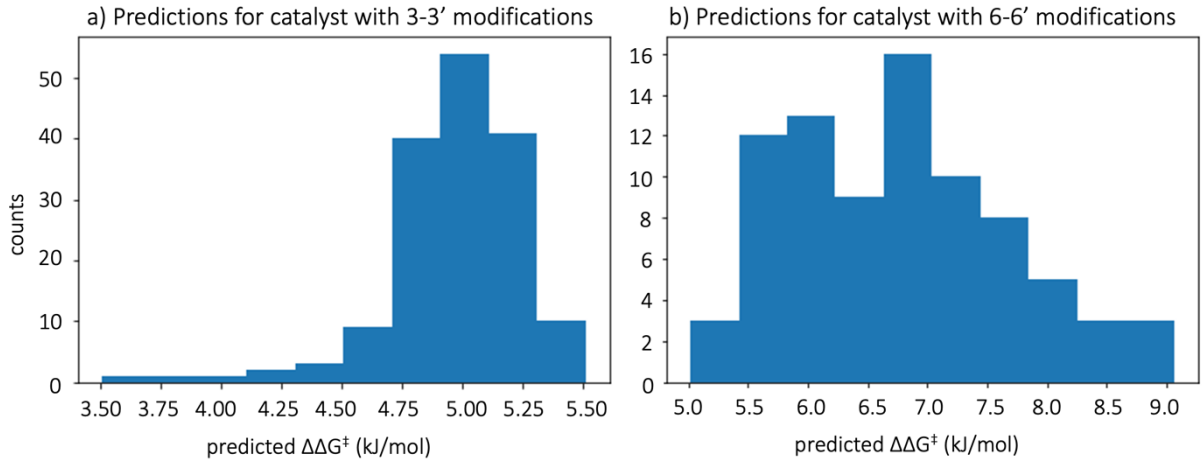


Figure 63: Histograms showing the occurrences of the predicted $\Delta\Delta G^{\ddagger}$ values; a) Catalysts with modifications on the 3-3' position show low $\Delta\Delta G^{\ddagger}$ values, b) Catalysts with modifications on the 6-6' position show high $\Delta\Delta G^{\ddagger}$values.

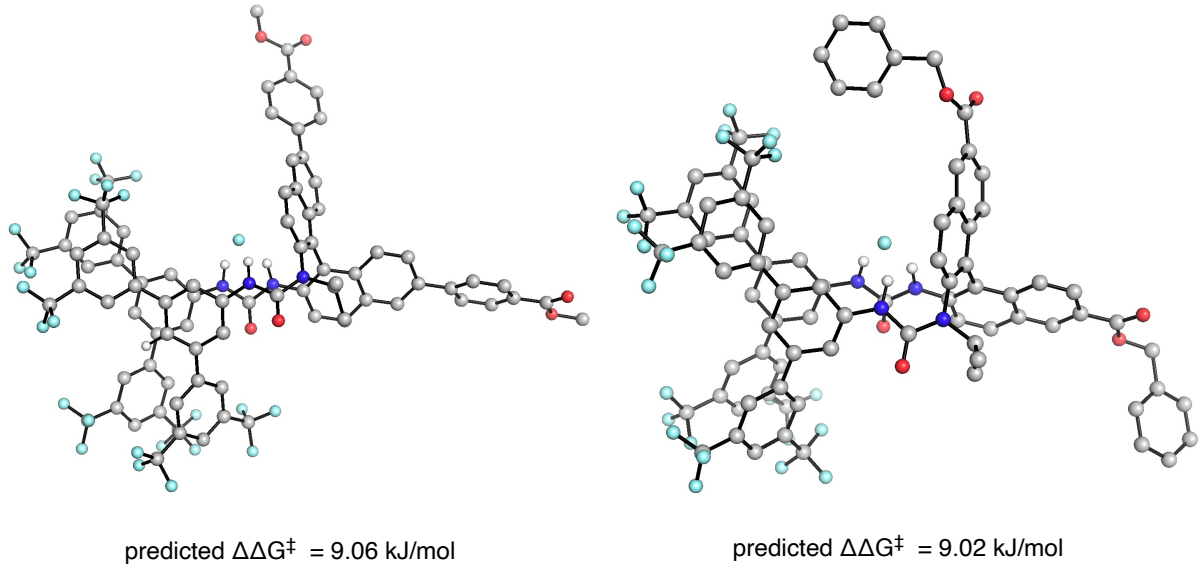


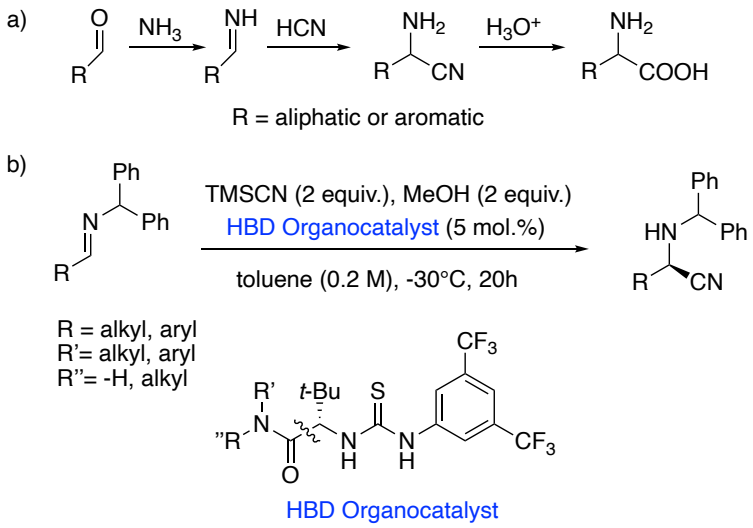
Figure 64: Top two high scoring catalysts with 6-6' modifications presenting electron withdrawing groups on the BINAM.

Due to the COVID-19 pandemic, experimental validation of our predictions was not feasible at the time. I believe that combining computational and experimental approaches, could further explore the underlying mechanisms and provide a more comprehensive understanding of the suggested catalytic structures.

4.3. Enantioselective Strecker synthesis of α-amino acids

To demonstrate the applicability of Model 5b, as well as our automated tool *Pythia*, we embarked on predicting the selectivity of the asymmetric variant of the Strecker synthesis of α-amino acids, first reported by Jacobsen and co-workers in the early 2000s.[434–438] This reaction remains one of the most widely used approaches to generate α-amino acids motifs. It involves the addition of hydrogen cyanide to imines, resulting in the formation of an α-aminonitrile intermediate, that upon hydrolysis yields the desired the α-amino acid derivative (Scheme 8a).[494]

Achieving control over the stereochemical outcome of these reactions is challenging in synthetic processes; especially when considering a broader range of unnatural α-amino acids that are not easily accessible through conventional chemo-enzymatic methods. To address this issue, Jacobsen and co-workers developed an asymmetric variant utilizing chiral (thio)ureas as HB donors that coordinate the cyanide anion, facilitating its nucleophilic addition to the iminium cation and lead to the formation of a stereochemically defined C-C bond (Scheme 8b).[435,494] Since the early 2000s, Jacobsen and his team have extensively employed this reaction with high yields and *ee* values exceeding 98%.[434–438]



Scheme 8: Strecker synthesis of α-amino acids. a) Uncatalyzed and b) Thiourea-catalyzed variant.

Computational studies showed two different pathways: the direct imine activation by the thiourea and the cyanide/isocyanide binding by the thiourea Figure 65.[435,495] The first one was found to be the less favored (23.2 kcal/mol higher than the second one). In the favored pathway, a proton transfer from the thiourea-bound HCN (or HNC) to imine occurs by generating a catalyst-bound cyanide·iminium ion pair.[495] The resulting ion pair undergoes a rearrangement

that determines the enantioselectivity of the reaction. The separation of the two charged species occurs through the transfer of the HB interaction of the iminium ion from the cyanide to the carbonyl of the catalyst amide. The subsequent step is a simultaneous stereospecific collapse to form the α-aminonitrile product.

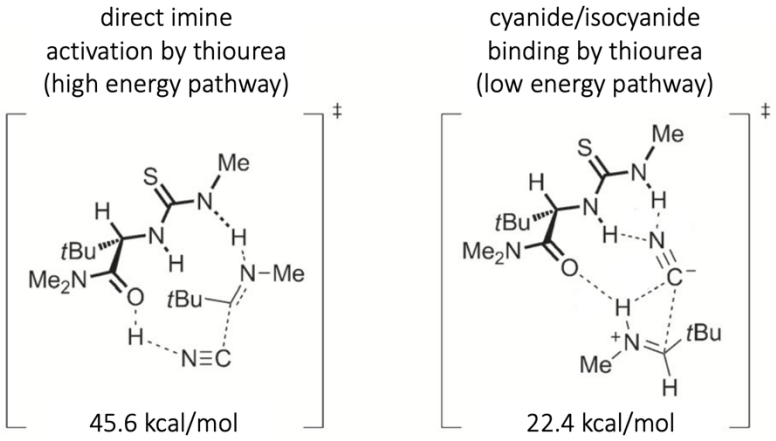
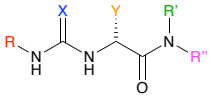


Figure 65: Mechanistic studies to explain the observed enantioselectivity: a) two ways of potential activation mechanism. Figure from Jacobsen et al.494

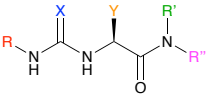
4.3.1. Computational workflows

Data were extracted from five publications by Jacobsen and co-workers, consisting of 63 substrates and 14 catalysts (Figure 66a, c).[434–438] The reactions were reported in four different temperatures -78, -75, -70 and -30 °C in all cases using toluene as a solvent. *ee* was converted to $\Delta\Delta G^{\ddagger}$ as described in Eq. 4.1. Overall, these publications lack negative results, for this reason we decided to include four catalysts with no chiral centers or with symmetric substitutions on both sides of the thiourea or on the amide group, which are expected to show no selectivity (Figure 66b). This resulted in a final data set consisting of 119 reactions, with 63 substrates and 18 catalysts.

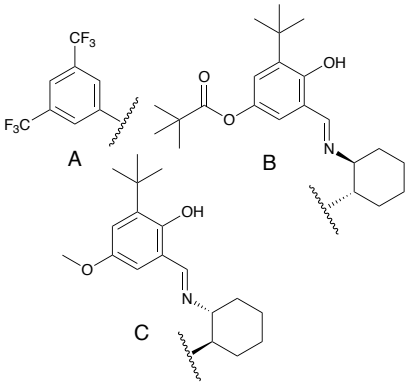
a) Experimentaly tested catalysts



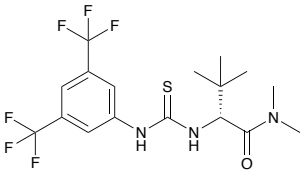
1J R = A; X = S; Y = *t*-Bu; R' = Me; R'' = Me
2J R = A; X = S; Y = *t*-Bu; R' = Me; R'' = Bn
3J R = A; X = S; Y = *t*-Bu; R' = Me; R'' = (*S*)-1-phenylethyl
4J R = A; X = S; Y = *t*-Bu; R' = Me; R'' = (*R*)-1-phenylethyl
5J R = A; X = S; Y = *t*-Bu; R' = Me; R'' = 1,1-bisphenylmethyl
6J R = A; X = O; Y = *t*-Bu; R' = Me; R'' = 1,1-bisphenylmethyl
7J R = A; X = S; Y = *t*-Bu; R' = Me; R'' = *i*-Pr
8J R = Ph; X = S; Y = Me; R' = Me; R'' = 1,1-bisphenylmethyl
9J R = B; X = O; Y = *t*-Bu; R' = H; R'' = Bn



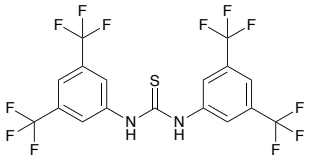
10J R = Ph; X = S; Y = *t*-Bu; R' = H; R'' = 1,1-bisphenylmethyl
11J R = Ph; X = S; Y = *t*-Bu; R' = Me; R'' = 1,1-bisphenylmethyl
12J R = Cy; X = S; Y = *t*-Bu; R' = Me; R'' = 1,1-bisphenylmethyl
13J R = C; X = S; Y = *t*-Bu; R' = H; R'' = Bn
14J R = B; X = S; Y = *t*-Bu; R' = H; R'' = Bn



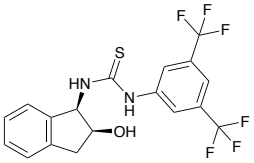
b) Unreactive catalysts



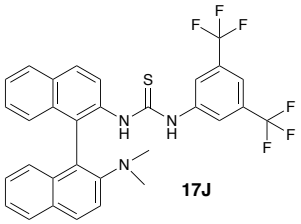
1J



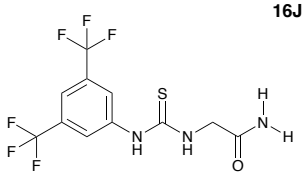
15J



16J

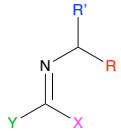


17J



18J

c) Experimentaly tested substrates



1J-s R = Ph; R' = Ph; X = H; Y = *t*-Bu
2J-s R = Ph; R' = Ph; X = H; Y = Et_2MeC
3J-s R = Ph; R' = Ph; X = H; Y = $PhMe_2C$
4J-s R = Ph; R' = Ph; X = H; Y = (1-methyl)cyclohexyl
5J-s R = Ph; R' = Ph; X = H; Y = 1-adamantly
6J-s R = Ph; R' = Ph; X = H; Y = cyclohexyl
7J-s R = Ph; R' = Ph; X = H; Y = *p*-$OMeC_6H_4$
8J-s R = Ph; R' = Ph; X = H; Y = *p*-MeC_6H_4
9J-s R = Ph; R' = Ph; X = H; Y = C_6H_5
10J-s R = Ph; R' = Ph; X = H; Y = *p*-ClC_6H_4
11J-s R = Ph; R' = Ph; X = H; Y = *p*-$CF_3C_6H_4$
12J-s R = Ph; R' = Ph; X = H; Y = *p*-CNC_6H_4
13J-s R = Ph; R' = Ph; X = H; Y = *p*-BrC_6H_4
14J-s R = Ph; R' = Ph; X = H; Y = *o*-$OMeC_6H_4$
15J-s R = Ph; R' = Ph; X = H; Y = *o*-BrC_6H_4
16J-s R = Ph; R' = Ph; X = H; Y = *m*-BrC_6H_4
17J-s R = Ph; R' = Ph; X = H; Y = 2-furyl
18J-s R = Ph; R' = Ph; X = H; Y = 3-furyl
19J-s R = Ph; R' = Ph; X = H; Y = 2-thiophenyl
20J-s R = Ph; R' = Ph; X = H; Y = 1-cyclohexenyl
21J-s R = Ph; R' = Ph; X = H; Y = (*E*)-1-methyl-pent-1-enyl
22J-s R = Ph; R' = Ph; X = H; Y = (*E*)-cinnamyl
23J-s R = Ph; R' = Ph; X = H; Y = (*E*)-hexenyl
24J-s R = vinyl; R' = H; X = Me; Y = C_6H_5
25J-s R = vinyl; R' = H; X = Et; Y = C_6H_5
26J-s R = vinyl; R' = H; X = Me; Y = napthalen-2-yl
27J-s R = vinyl; R' = H; X = Me; Y = 4-phenylpropan-1-yl
28J-s R = Ph; R' = H; X = Me; Y = C_6H_5
29J-s R = Ph; R' = H; X = Me; Y = *p*-MeC_6H_4
30J-s R = Ph; R' = H; X = Me; Y = *p*-BrC_6H_4
31J-s R = Ph; R' = H; X = Me; Y = *p*-$NO_2C_6H_4$
32J-s R = Ph; R' = H; X = Me; Y = *p*-$OMeC_6H_4$
33J-s R = Ph; R' = H; X = Me; Y = *p*-$CF_3C_6H_4$
34J-s R = Ph; R' = H; X = Me; Y = *m*-BrC_6H_4
35J-s R = Ph; R' = H; X = Me; Y = *o*-BrC_6H_4
36J-s R = Ph; R' = H; X = Me; Y = *t*-Bu
37J-s R = *p*-$OMeC_6H_4$; R' = H; X = Me; Y = C_6H_5
38J-s R = *p*-$CF_3C_6H_4$; R' = H; X = Me; Y = C_6H_5
39J-s R = *p*-*t*-BuC_6H_4; R' = H; X = Me; Y = C_6H_5
40J-s R = *p*-BrC_6H_4; R' = H; X = Me; Y = C_6H_5
41J-s R = vinyl; R' = H; X = H; Y = C_6H_5
42J-s R = vinyl; R' = H; X = H; Y = *p*-$OMeC_6H_4$
43J-s R = vinyl; R' = H; X = H; Y = *p*-BrC_6H_4
44J-s R = vinyl; R' = H; X = H; Y = napthalen-2-yl
45J-s R = vinyl; R' = H; X = H; Y = cyclohexyl
46J-s R = vinyl; R' = H; X = H; Y = *t*-Bu
47J-s R = vinyl; R' = H; X = H; Y = *m*-$OMeC_6H_4$
48J-s R = vinyl; R' = H; X = H; Y = *o*-$OMeC_6H_4$
49J-s R = vinyl; R' = H; X = H; Y = *p*-MeC_6H_4
50J-s R = vinyl; R' = H; X = H; Y = *m*-MeC_6H_4
51J-s R = vinyl; R' = H; X = H; Y = *o*-MeC_6H_4
52J-s R = vinyl; R' = H; X = H; Y = *m*-BrC_6H_4
53J-s R = vinyl; R' = H; X = H; Y = *o*-BrC_6H_4
54J-s R = vinyl; R' = H; X = H; Y = *p*-*t*-BuC_6H_4
55J-s R = Ph; R' = H; X = H; Y = *t*-Bu
56J-s R = Ph; R' = H; X = H; Y = cyclohexyl
57J-s R = Ph; R' = H; X = H; Y = 1-cyclohexenyl
58J-s R = Ph; R' = H; X = H; Y = $(CH_3)_3CCH_2$
59J-s R = Ph; R' = H; X = H; Y = $CH_3(CH_2)_4$
60J-s R = Ph; R' = H; X = H; Y = *i*-Pr
61J-s R = Ph; R' = H; X = H; Y = cyclopropyl
62J-s R = Ph; R' = H; X = H; Y = cyclooctyl
63J-s 3,4-dihydroisoquinoline

Figure 66: Data set under study. a) Experimentally tested catalysts extracted from literature. b) Unreactive catalysts made computationally. c) Experimentally tested substrates extracted from literature.[434–438]

In this section, three types of descriptors were investigated, Mordred descriptors, Morgan fingerprints, and DFT descriptors generated following Model 5b (§ 4.2.2). To interpret the DFT model, LASSOCV was employed to identify the most important descriptors.

For the Mordred descriptors, a correlation analysis was performed to identify descriptors that exhibited a significant correlation with $\Delta\Delta G^{\ddagger}$. Pearson correlation coefficients were computed between the Mordred descriptors of each substrate and catalyst and the respective $\Delta\Delta G^{\ddagger}$ value. Descriptors with a correlation coefficient of 0.36 or higher were considered for the substrates, while descriptors with a correlation coefficient of 0.55 or higher were considered for the catalysts. The difference in the criteria was based on the number of descriptors correlating. We further analyzed these features for significance using a two-tailed p-test over 5,000 random sample permutations using the Pearson's correlation coefficient as the test statistic. Following feature generation, we applied one hot encoding for categorical features and min-max scaling for continuous features, as described in *Pythia* (§3.3 – Notebooks 3&6).

A range of ML algorithms were employed to determine the most predictive model. They included, regressors such as Bayesian, DT, kNN, Support Vector (SV), GP, and LASSOCV, and classifiers such as Ada Boost, ET, LR, DT, GP, and kNN. Regression models were evaluated based on R^2 and RMSE, while classification models were assessed based on accuracy, sensitivity, specificity, ROC curves, MCC and g-mean (please refer to §2.2.3 for explanation of the metrics).

In the case of classification, a binary approach was employed, with class 1 representing $\Delta\Delta G^{\ddagger}$ >= 4 kJ/mol and class 0 representing $\Delta\Delta G^{\ddagger}$ < 4 kJ/mol. For the 119 reactions in our data set, 44 fell into class 0, and 75 fell into class 1. To address class imbalance, additional sampling points were generated for the minority class using SMOTE for non-categorical features and SMOTEN for categorical features. Once the synthetic data points have been added, there is a total of 148 data points, with 29 synthetic samples used.

As previously described in *Pythia* (§3.2), the entire dataset was cross-validated for training, while 10% of the data points (12 reactions) with the lowest *Tanimoto* similarity were reserved for testing. Among the 107 remaining reactions, 38 are classified as 0 and 69 as 1. Additionally, 29 synthetic data are generated, resulting in a training set of 136 data points.

Classifiers are expected to outperform regressors as classification problems are generally more tractable. Additionally, synthetic data oversampling was applied to class 0, enabling classifiers to better predict low reactivity catalysts compared to regressors. It's worth noting that utilizing

classification models for predicting continuous endpoint values, such as $\Delta\Delta G^{\ddagger}$, deviates from standard practice and does not align with conventional methodologies. However, in the case of *Pythia*, these models were constructed based on the available data. Therefore, the inclusion of classification models in this analysis is primarily intended to showcase *Pythia's* versatility and capabilities. While we provide a summary of the results, we refrain from engaging in further critical discussion, as this study is primarily focused on regression tasks rather than classification.

4.3.2. Results and discussion

As a starting point, a LASSOCV model was trained on the original 99 reactions extracted from literature. However, the model's accuracy was poor (R^2 = 0.55 and RMSE = 1.46 kJ/mol) and it was not considered further (please refer to the supplementary material, subdirectory Chapter 4.3). Therefore, only the models generated from the 119 reactions were considered further. These are discussed based on the type of descriptor employed (Mordred, fingerprints and DFT). We present first the results for the classifiers and then the results for the regressors. A summary of the best performing models can be found in Table 8.

Table 8: Summary of the best performing algorithms, for the unseen dataset, according to the different descriptor models.

Descriptors	*Model*	*Metrics*
Mordred	GP	accuracy = 0.92
	kNN	R^2 = 0.70, RMSE = 1.64 kJ/mol
Fingerprints	ET	accuracy = 0.83
	LASSOCV	R^2 = 0.80, RMSE = 1.34 kJ/mol
DFT	ET	accuracy = 0.83
	LASSOCV	R^2 = 0.76, RMSE = 1.44 kJ/mol

Mordred descriptors

When considering the entire data set, 55 Mordred features are shown to be statistically significant, two of which are considered as categorical. After the one hot encoding the feature set extends to 61 as each unique value of the categorical features becomes a binary feature array. When the 12 reactions are kept outside the dataset, 38 Mordred features are considered. The list of features considered in each case is given in the supplementary material (subdirectory Chapter 4.3).

Table 9 shows that all classifiers achieve high accuracies (>0.8). GP gives the highest accuracy, while Ada Boost and ET perform similarly. Additionally, all three models have high sensitivities and specificities. The MCC values demonstrate that all models perform better than random (values well above 0). Finally, all classifiers tested were found to have a g-mean > 0.8 and therefore are considered to perform well. We remind the reader that g-mean penalizes classifiers that have imbalanced performance across classes. A high g-mean indicates that the classifier performs well in terms of both positive and negative class predictions.

Analysis of the confusion matrices shows that each of the top three performing classifiers predicts a slightly different amount of FN and FP predictions (Appendix C - Figure 88). This suggests that GP will predict more FN, while ET and Ada Boost will predict more FP. Generally, the latter is more desirable as it means that we might perform more unnecessary experiments, but we will not lose a good candidate. On the other hand, if a reaction is predicted falsely negative, it is likely that it will not be tested experimentally. Analysis of the ROC curves indicates that both GP and Ada Boost exhibit high classification performance (AUC of 0.94 and 0.93 respectively). This demonstrates that both models possess high discriminatory abilities and rapidly improve their classification skills as the decision threshold varies. Both models are well-suited for the binary classification task, and the small difference in AUC values highlights their comparable abilities. ET follows with an AUC of 0.88 suggesting a slightly weaker ability to discriminate between classes (Appendix C - Figure 88).

Table 9: Performance of the different classifiers when using Mordred descriptors for the training set.

Classifier	*Accuracy*	*Sensitivity*	*Specificity*	*MCC*	*Precision*	*g-mean*
kNN	0.82	0.85	0.78	0.64	0.80	0.82
GP	0.89	0.88	0.91	0.78	0.90	0.89
DT	0.81	0.73	0.89	0.63	0.87	0.81
ET	0.86	0.93	0.78	0.72	0.81	0.85
Ada Boost	0.86	0.88	0.85	0.73	0.86	0.86
LR	0.84	0.92	0.76	0.68	0.79	0.83

For the test set, most classifiers perform well with high accuracies and perfect sensitivity. With the exception of Ada Boost, which surprisingly underperforms in terms of accuracy (0.67), specificity (0.33), MCC (0.45), precision (0.60) and g-mean (0.58), especially when compared to GP, which performs the best in all metrics showing the highest accuracy (0.92), specificity (1.00), MCC (0.85), precision (0.86) and g-mean (0.91) (Table 10). GP's perfect AUC (1.00) further validates its effectiveness in distinguishing between reaction classes. From the confusion matrices (Appendix C - Figure 89) it is evident that Ada Boost has more FP predictions, which is in accordance with the metrics discussed above, however it shows better

classification skill (AUC = 0.78), compared to ET (AUC = 0.67), which interestingly performs better in every metric, as shown in Table 10. This disagreement in the metrics could be attributed to the fact these endpoints ($\Delta\Delta G^{\ddagger}$) are inherently continuous and not explicitly designed for classification tasks.

Table 10: Performance of the different classifiers when using Mordred descriptors for the test set.

Classifier	*Accuracy*	*Sensitivity*	*Specificity*	*MCC*	*Precision*	*g-mean*
kNN	0.83	1.00	0.67	0.71	0.75	0.82
GP	0.92	1.00	0.83	0.85	0.86	0.91
DT	0.75	1.00	0.50	0.58	0.67	0.71
ET	0.83	1.00	0.67	0.71	0.75	0.82
Ada Boost	0.67	1.00	0.33	0.45	0.60	0.58
LR	0.83	1.00	0.67	0.71	0.75	0.82

Based on these findings, it can be concluded that while the classifiers can offer some insights, it's essential to interpret their performance in the context of the specific task at hand and consider the limitations inherent in applying classification methodologies to continuous endpoint prediction. Despite the challenges GP stood out as the top-performing model across multiple metrics, exhibiting the highest accuracy, specificity, MCC, precision, g-mean, and AUC.

When the same features are considered to train the regressors, the models perform moderately except for GP, which is unable to make any predictions (RMSE = 161.52 kJ/mol). SV performs slightly better than the rest, with $R^2 = 0.71$ and RMSE = 1.52 kJ/mol (Appendix C - Figure 90). Based on these results we do not expect the models to perform adequately when used to predict unseen data. As expected, all regressors except for kNN ($R^2 = 0.70$ and RMSE = 1.64 kJ/mol) lack substantial predictive capability (Appendix C - Figure 91). We therefore conclude that Mordred descriptors are weak when used in combination with regression models for this type of predictions and are not discussed further.

Morgan fingerprints

After removing features represented by 0 along all reactions, 227 features remain to describe each reaction. Among the classifiers ET performs better in all metrics, while LR outperforms GP in sensitivity, but not in specificity (Table 11). Analysis of the confusion matrices of the top three performing classifiers, reveals that while both LR and GP perform well, LR gives fewer FN predictions and GP gives fewer FP predictions (Appendix C - Figure 92). Interestingly, ET only misclassifies two reactions as FN.

Table 11: Performance of the different classifiers when using Morgan fingerprints for the training set.

Classifier	*Accuracy*	*Sensitivity*	*Specificity*	*MCC*	*Precision*	*g-mean*
kNN	0.86	0.92	0.80	0.72	0.82	0.86
GP	0.91	0.93	0.88	0.81	0.88	0.91
DT	0.85	0.81	0.89	0.71	0.88	0.85
ET	0.92	0.97	0.86	0.84	0.88	0.92
Ada Boost	0.89	0.93	0.84	0.77	0.85	0.88
LR	0.91	0.95	0.86	0.81	0.88	0.90

For the test set, all classifiers, except DT, perform similarly across all metrics Table 12. However, analysis of the ROC curves indicates that GP gains skill faster (AUC = 0.97), followed by ET (AUC = 0.86) and then LR (AUC = 0.81) (Appendix C - Figure 93). Interestingly, Ada Boost, which showed lower accuracy when using Mordred descriptors for the test set, now performs well. The striking similarity in performance across the majority of classifiers may suggest feature redundancy. It's plausible that the 227 binary features used for constructing the models contain considerable redundancy or irrelevance, leading to all models learning similar decision boundaries and achieving comparable performance levels.

Table 12: Performance of the different classifiers when using Morgan fingerprints for the test set.

Classifier	*Accuracy*	*Sensitivity*	*Specificity*	*MCC*	*Precision*	*g-mean*
kNN	0.83	1.00	0.67	0.71	0.75	0.82
GP	0.83	1.00	0.67	0.71	0.75	0.82
DT	0.75	1.00	0.50	0.58	0.67	0.71
ET	0.83	1.00	0.67	0.71	0.75	0.82
Ada Boost	0.83	1.00	0.67	0.71	0.75	0.82
LR	0.83	1.00	0.67	0.71	0.75	0.82

All regressors perform relatively well ($R^2 > 0.7$, RMSE < 1.55 kJ/mol, Appendix C - Figure 94) for the training set, with the exception of GP. GP exhibits high errors and low correlation (RMSE = 2.9 kJ/mol, $R^2 = 0.3$), probably due to the small amount of data and large number of descriptors employed. When predicting on the unseen dataset, GP is still the least predictive model. LASSOCV shows its robustness even with a small training set, avoiding over-fitting ($R^2 = 0.80$, RMSE = 1.34 kJ/mol, Appendix C - Figure 95). Despite achieving predictive performance, employing regression tasks with Morgan fingerprints allows limited room for improvement in terms of accuracy and interpretability.

Generally, when employing Morgan fingerprints, both the classifiers and the regressors perform better than when utilizing Mordred descriptors, however one should not ignore the difference in the feature vector (61 Mordred feature *vs* 277 fingerprint bits). The number of features in the vector affects the model's complexity, information representation, and ability to

capture relevant patterns. A larger feature vector can potentially provide a more comprehensive representation of the data, allowing the model to extract more nuanced patterns and make better predictions. Based on the above we conclude that Morgan fingerprints describe this reaction better than Mordred descriptors. However, relying solely on fingerprints presents challenges when attempting to extract chemical insights from them.

DFT descriptors

To describe the reactions with DFT descriptors, 43 features are employed. For the substrates,10 features including HOMO, LUMO, N charge, dipole moment, and steric descriptors calculated along the maroon arrows in Figure 67a are used. Additionally, for the catalysts, 33 features are included such as HOMO, LUMO, dipole moment, charges for the (thio)urea Hs and the CN^-, average NMR shifts for the Hs and Cs of the (thio)urea moiety and the C of the CN^-, BO and steric descriptors along the highlighted bonds in Figure 67b.

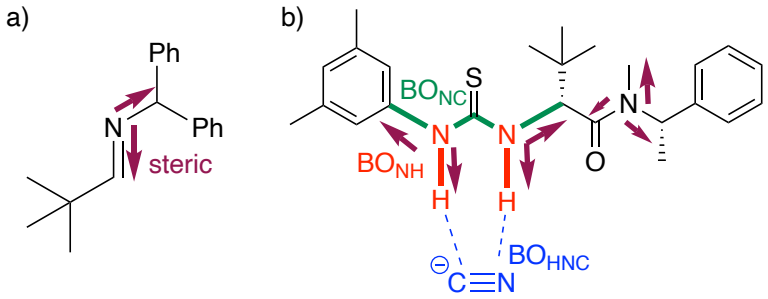


Figure 67: Representation of the structural descriptors calculated for this dataset; a) for the substrates and b) for the catalysts. In green, red, and blue the different BOs considered. In maroon the bonds scanned for sterics starting from the Ns.

Analysis of the different metrics across the classifiers demonstrates they all have high accuracies (> 0.82), sensitivities, specificities and g-mean (Table 13). The MCC values show that all models perform better than random (values well above 0). LR and ET perform similarly well, with LR outperforming ET in sensitivity (better at identifying TP), while ET outperforms LR in specificity (better at identifying TN). Ada Boost and GP follow. Furthermore, the ROC curves show AUC close to 0.9 for the three top performing classifiers indicating that all of them gain skill fast (Appendix C - Figure 96).

Table 13: Performance of the different classifiers when using DFT descriptors for the training set.

Classifier	*Accuracy*	*Sensitivity*	*Specificity*	*MCC*	*Precision*	*g-mean*
kNN	0.82	0.84	0.80	0.64	0.81	0.82
GP	0.85	0.91	0.80	0.71	0.82	0.85
DT	0.84	0.93	0.76	0.70	0.79	0.84
ET	0.86	0.89	0.84	0.73	0.85	0.86
Ada Boost	0.85	0.88	0.82	0.70	0.83	0.85
LR	0.86	0.91	0.82	0.73	0.84	0.86

For the test set all classifiers accurately predict the 12 unseen data points (Table 14). kNN, DT and ET perform similarly well however, DT gains skill faster (AUC = 0.81), suggesting that the model can reliably distinguish between the classes (Appendix C - Figure 97). At the same time GP, Ada Boost and LR follow, performing similarly in all metrics (Table 14). Most importantly none of the classifiers has data points classified as FN. In the context of chemical reactions and $\Delta\Delta G^{\ddagger}$ prediction, FN occur when the model fails to identify reactions resulting in high $\Delta\Delta G^{\ddagger}$ values. This can lead to missed opportunities for further investigation or optimization of reactions, which here is avoided.

Conversely, the low number of FP signifies a particularly positive outcome. In chemical experimentation, FP would correspond to predicted reactions with high $\Delta\Delta G^{\ddagger}$ values that do not actually exhibit such energetics when tested experimentally. Having two FP out of 12 test points, indicates that the model is effectively filtering out reactions that are unlikely to have the predicted $\Delta\Delta G^{\ddagger}$ values. This is especially valuable, where experimental resources can be limited and costly. By minimizing FP, it is ensured that experimental efforts are focused on reactions with a higher likelihood of exhibiting the predicted $\Delta\Delta G^{\ddagger}$.

Table 14: Performance of the different classifiers when using DFT descriptors for the test set.

Classifier	*Accuracy*	*Sensitivity*	*Specificity*	*MCC*	*Precision*	*g-mean*
kNN	0.83	1.00	0.67	0.71	0.75	0.82
GP	0.75	0.83	0.67	0.51	0.71	0.75
DT	0.83	1.00	0.67	0.71	0.75	0.82
ET	0.83	1.00	0.67	0.71	0.75	0.82
Ada Boost	0.75	0.83	0.67	0.51	0.71	0.75
LR	0.75	0.83	0.67	0.51	0.71	0.75

When the regressors are employed for the training set, kNN performs well with high correlation and low error (R^2 = 0.77, RMSE = 1.35 kJ/mol) and LASSOCV follows (R^2 = 0.74, RMSE = 1.44 kJ/mol). Not surprisingly, and in accordance with what we observed previously (Mordred and Morgan fingerprints regression models) GP does not perform well, showing high errors (Appendix C - Figure 98). When asked to predict the test set, the models exhibit good predicting

power, achieving high R^2 and low RMSE across all regressors (Appendix C - Figure 99). kNN and LASSOCV show the highest correlation and lowest error ($R^2 = 0.76$, RMSE ≈ 1.45 kJ/mol). We remind the reader that kNN makes predictions based on the similarity of data points in the feature space. It does not learn explicit relationships between features and the target variable but instead relies on the proximity of neighboring data points. As a result, interpreting the predictions of a kNN model can be challenging, especially when considering the contribution of individual features to the predictions. On the other hand, LASSOCV provides interpretable models due to its regularization and feature selection properties. Therefore, LASSOCV is chosen as our final model for the prediction of $\Delta\Delta G^{\ddagger}$ in this set of reactions (Figure 68).

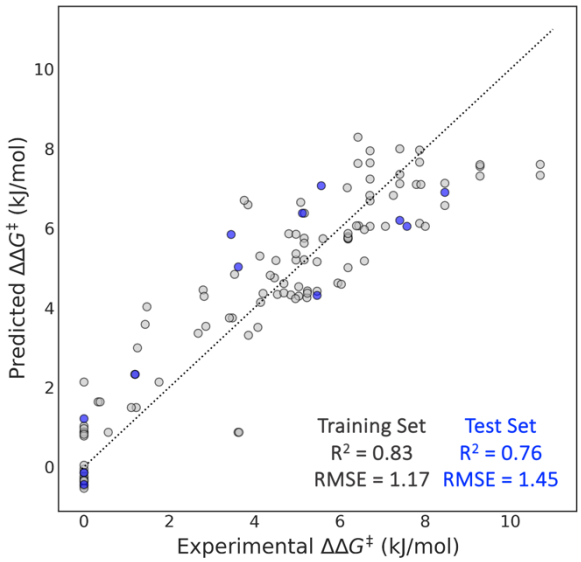


Figure 68: Final model to predict $\Delta\Delta G^{\ddagger}$ with DFT features and LASSOCV. Lowest energy conformation for the substrates and catalysts at the PBE-D3BJ/def2-SVP level of theory - Model 5b. Both the training (in grey) and the test set (in blue) show high R^2 and low RMSE.

To identify the chemical features that may affect selectivity, we analyzed the important coefficients obtained from LASSOCV (Eq. 4.3). Here, descriptors for the substrates are shown in blue and descriptors for the catalysts in black:

$$\begin{aligned}\Delta\Delta G^{\ddagger} = 1.05 &+ 10.5 N_1\text{charge} + 0.6 N_1C_0(B_1) + 0.1 N_1C_0(L) - 0.3 N_1C_2(B_1) - 0.8\,\text{LUMO} \\ &+ 4.6 N_8C_9(B_1) + 1.1 N_1C_{42}(B_1) + 0.6\, N_5C_4(L) + 0.4\, N_5C_4(B_5) - 0.3\, H\text{charge} \\ &- 0.8\, CN\text{charge} - 0.7 N_5H_{27}(B_5) - 0.4 N_8H_{26}(L) - 2.5 N_8H_{26}(B_1) \qquad (4.3)\end{aligned}$$

In Figure 69 the atoms Eq. 4.3 refers to are shown, both for the substrates and the catalysts. The analysis of the substrate parameters indicates that the charge on the N_1 contributes the most (Figure 69a). This highlights the importance of stabilizing the iminium ion during the reaction especially when electron withdrawing, and electron poor groups are present. Additionally, aldimines, which have a lower charge on the nitrogen, exhibit lower $\Delta\Delta G^{\ddagger}$ compared to ketoimines. We believe that the LUMO is a proxy for differentiating between aldimines and

ketoimines. Based on Eq. 4.3 LUMO has a negative coefficient, as the actual value of LUMO is negative, substrates with higher absolute LUMO values (ketoimines) outperform substrates with lower LUMO values (aldimines).

The presence of positive coefficients in $N_1C_0(B_1)$ and $N_1C_0(L)$ suggest the need for steric bulk to achieve high selectivity; however, caution is needed when adding bulky groups as indicated by the small coefficient for $N_1C_0(L)$. From literature we know that bulky ketoimines are less reactive and selective due to steric congestion at the reaction center. Feature $N_1C_2(B_1)$ represents groups where steric bulk on the imine protecting group can cause steric congestion, as indicated by the negative coefficient, which plays an important role in determining the point of attack of the nucleophile.

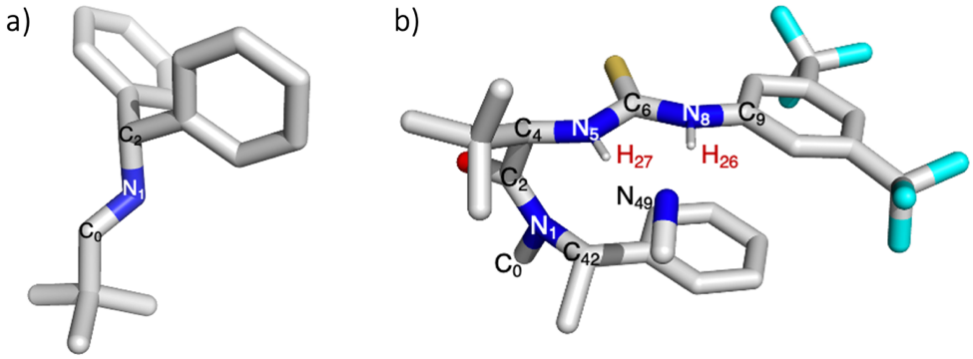


Figure 69: Representation of the important descriptors that influence the $\Delta\Delta G^{\ddagger}$. The atoms of interest are marked a) For the substrate the and b) For the catalyst according to Eq. 4.3.

For the catalysts, based on Eq. 4.3 and Figure 69b, we propose that bulkier groups are desired when substituting on the N_8C_9 as indicated by the positive coefficient of B_1 which suggests the presence of an aromatic group with further substitutions, in agreement with the experimental data. At the same time, descriptors $N_8H_{26}(B_1)$, N_8H_{26} (L) and N_5H_{27} (L) have a negative coefficient implying that the area around the H-CN^- needs to be free of steric bulk. Steric information from the N_1 accounts for the different substitutions around the amidic nitrogen. Here, $N_1C_{42}(B_1)$ is present with a positive coefficient, and it depicts the overall size of the functional groups substituted on the N_1, indicating that larger functional groups, such as phenyls, lead to higher selectivity. DFT analysis conducted by the Jacobsen group, supports the idea that substitutions on the amide core of the catalyst play a significant role in the *ee.* Indeed they demonstrated that the origin of the observed enantioselectivity is the variation in the distances of the iminium cation from the carbonyl moiety of the catalyst and the cyanide ion.[496]

Features N_5C_4(L) and $N_5C_4(B_5)$ have positive coefficients and are directly related with the chiral information expressed on carbon C_4 and on the size of the substituent selected on that position (t-butyl more advantageous than Me). Finally, $\Delta\Delta G^{\ddagger}$ is influenced by the charge of the two thiourea hydrogens, which are directly involved in the coordination of the anion, and the charge of the cyanide ion itself. Indeed, a possible deprotonation of the catalyst promoted by the anion could result in loose protons H_{26} and H_{27}, and in a corresponding decrease in selectivity. Contemporarily, if the cyanide is not properly coordinated, the high charge on it can induce a similar result.

Our LASSOCV analysis has revealed a set of important descriptors with their corresponding coefficients. These descriptors not only contribute to the predictive power of our model but also provide valuable insights into the underlying factors governing the selectivity of the reaction under study. This demonstrates that our model is not only predictive but also informative and highly interpretable. By aligning the identified descriptors with previous experimental and computational studies, we can establish meaningful connections between the model's findings and the existing knowledge in the field.

Conclusions

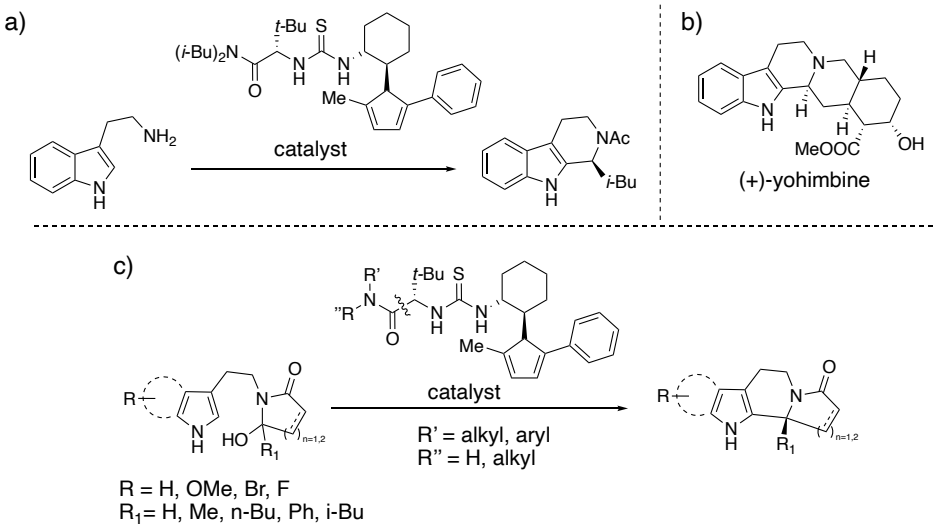
Here, we developed ML models for the prediction of selectivity for the Strecker synthesis of α-amino acids. A careful comparative study was conducted comparing the performance of three distinct types of descriptors and various ML algorithms. Initially we investigated how Mordred descriptors and Morgan fingerprints perform and we concluded that while Mordred descriptors perform adequately with classifiers, they do not perform as well with regressors. On the other hand, Morgan fingerprints perform well both with regressors and classifiers.

Following this analysis, we employed DFT descriptors (calculated based on Model 5b). LASSOCV demonstrated its robustness despite its relative simplicity. The utilization of DFT descriptors also facilitated interpretability, enabling us to gain insights into the underlying factors influencing selectivity, for example the importance of stabilizing the iminium ion during the reaction. This study can serve as a foundation for future endeavors, in designing novel catalysts for this set of reactions, positioning our model as a valuable tool that can guide researchers in developing more selective catalysts.

4.4. Pictet-Spengler cyclisations of hydroxylactams

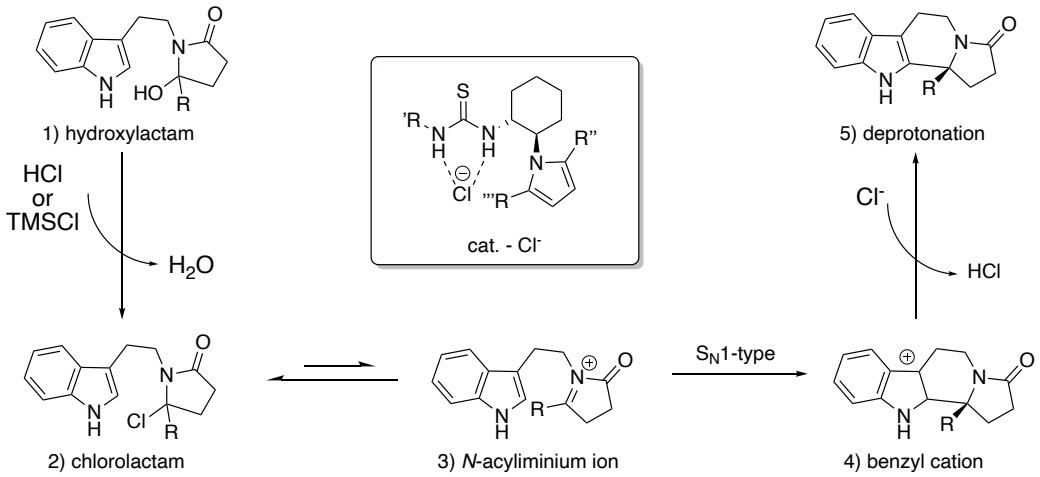
To further challenge *Pythia* and Model 5b (§ 4.2.2) and detect its limitations, we investigated its ability to predict selectivity in the Pictet-Spengler cyclisation of hydroxylactams reaction, which plays a significant role in the synthesis of six-membered heterocyclic, including various natural products.[497]

This reaction involves the cyclization of electron-rich aryl groups onto iminium electrophiles. The asymmetric variant of this reaction, originally developed by Jacobsen and Taylor, utilized chiral thioureas as catalysts to facilitate the cyclization of indoles onto N-acyliminium ions (Scheme 9a). Using this methodology, the authors successfully achieved the total synthesis of (+)-yohimbine (Scheme 9b). To expand the applicability of this reaction, the authors explored the possibility of generating N-acyliminium ions through the *in situ* dehydration of hydroxylactams (Scheme 9c).[495]



Scheme 9: a)Thiourea catalysed acyl-Pictet-Spengler reaction, b)(+)-yohimbine structure, c)Thiourea-catalysed enantioselective Pictet-Spengler cyclisation of hydroxylactams.[495]

In the proposed mechanism by the Jacobsen *et al*., the hydroxylactam undergoes an exchange of the alcoholic group with the chloride anion deriving from the HCl or TMSCl, which are used as *in situ* chlorinating agents (Scheme 10 – steps 1 & 2). After an equilibration step, where the Cl^- is removed by the catalyst's thiourea moiety (Scheme 10 – step 3), the resulting iminium ion undergoes intramolecular cyclisation promoted by the nearby indolic/pyrrolic group (Scheme 10 – step 4). The resulting cyclized product then deprotonates and restores the aromaticity of the system (Scheme 10 – step 5).[497]



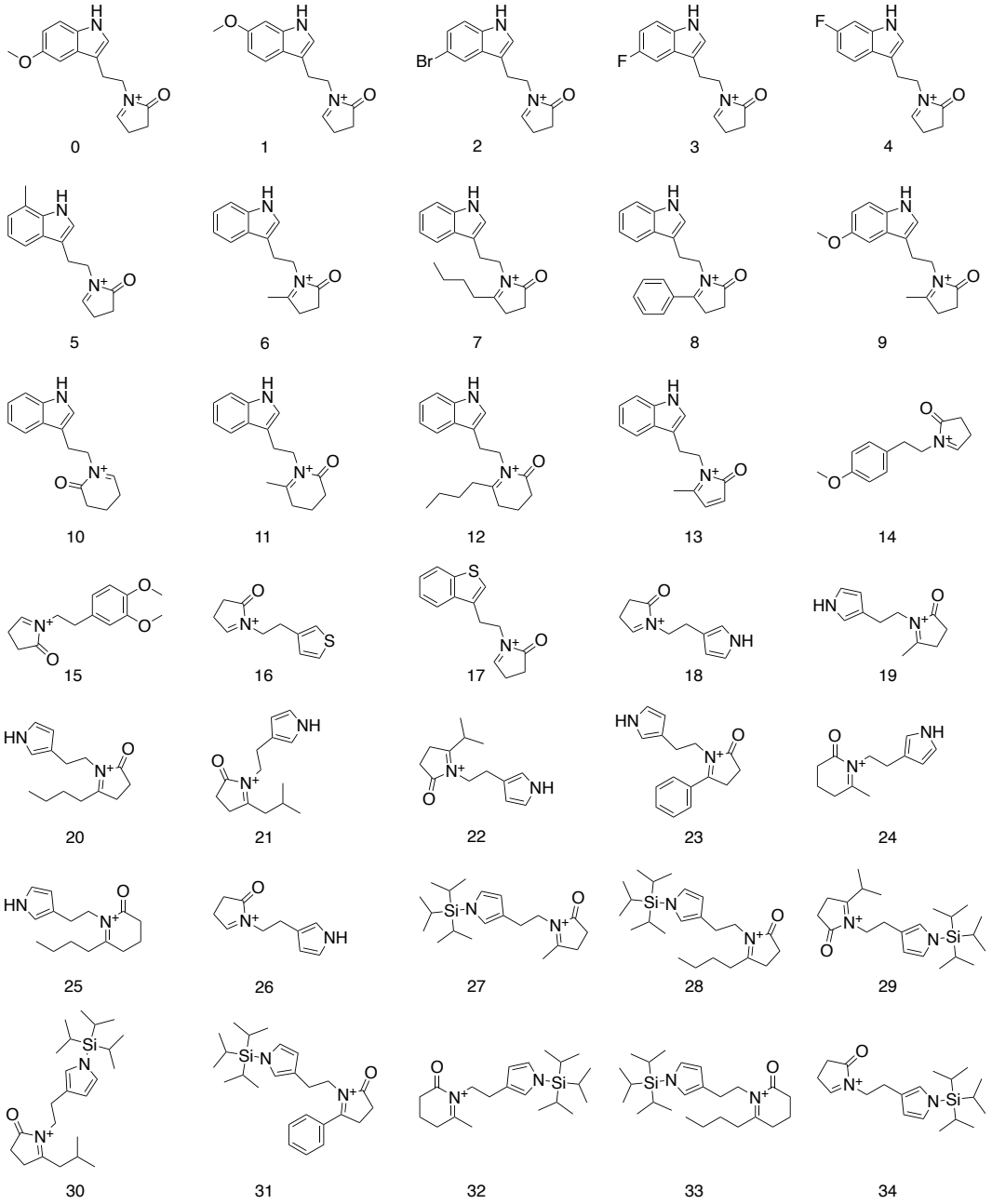
Scheme 10: Proposed chlorolactam formation and anion-binding mechanism.[497]

^{1}H NMR studies of a hydroxylactam substrate in the presence of TMSCl indicated that the formation of the corresponding chlorolactam is fast and irreversible. Furthermore, the increase in reactivity of the alkylated (R = Me) *vs* the reduced amine (R = H) suggested an S_N1-type mechanism in the cyclization step.[497] Although these experiments confirmed the presence of an *N*-acyliminium ion during the reaction pathway, the way the catalyst interacts with the substrate during the enantiodetermining step was still undefined. To explore this further, the authors performed DFT calculations. However, attempts to compute some of the *N*-acyliminium ions bound to the thiourea failed to converge. A notable interaction was identified between the thiourea and the chlorolactam, involving the α-chloro substituent. Therefore it was proposed that the reaction occurs *via* an ion pair constituted by the chiral thiourea-bound and *N*-acyliminium chloride.[497] This specie results from the dissociation of the chloride in α-position induced by the thiourea's proton catalyst. To support the idea of an anion-binding model, the authors highlighted halide counterion effects which increased as the dimension of the anion increased (i.e., Cl, 97% *ee*; Br, 68% *ee*; I, <5% *ee*) together with solvent effects (MTBE, 97% *ee*; CH_2Cl_2, <5% *ee*). Moreover, in agreement with the proposed S_N1-type mechanism, a rate acceleration was observed with the increase in substituents at the electrophilic center.[495,497]

4.4.1. Computational workflows

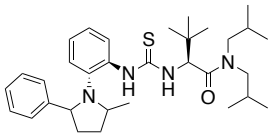
A total of 90 experimentally tested reactions (35 substrates – Scheme 11 and 45 catalysts – Scheme 12), extracted from two different papers,[497,498] compose our new data set. MTBE (Methyl tert-butyl ether) was used as solvent and the temperatures tested were at -78°C, -55°C, -30°C, and 0°C. *ee* was converted to $\Delta\Delta G^{\ddagger}$ as described in Eq. 4.1.

Experimentally tested substrates

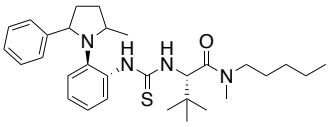


Scheme 11: Experimentally tested substrates for the 90 reactions under study.[497,498]

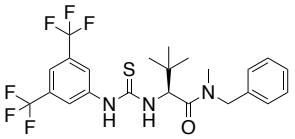
Experimentaly tested catalysts



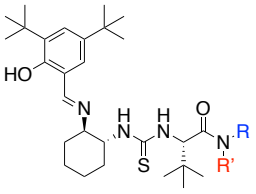
0



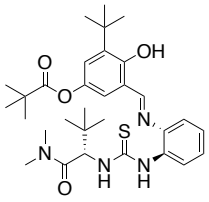
1



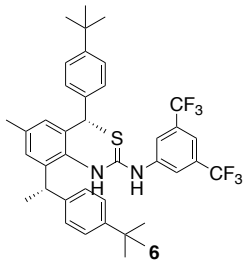
2



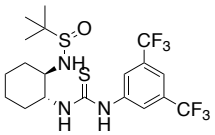
3 R = toluene, R' = Me
4 R = toluene, R' = toluene



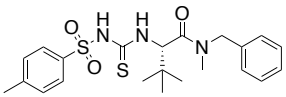
5



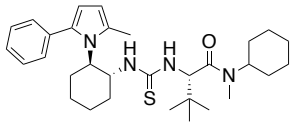
6



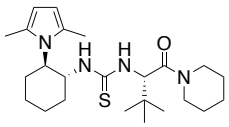
7



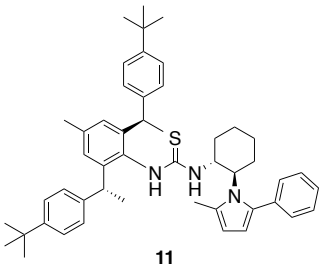
8



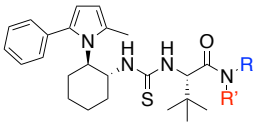
9



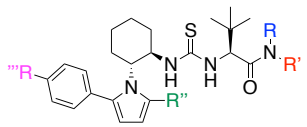
10



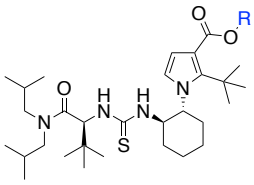
11



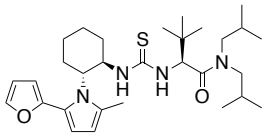
12 R = n-butyl, R' = H
13 R =n-pentyl, R' = H
14 R = n-hexyl, R' = H
15 R = n-heptyl, R' = H
16 R = iso-butyl, R' = H
17 R = tert-butyl, R' = H
18 R = n-butyl, R' = Me
19 R = n-hexyl, R' = Me
20 R = iso-butyl, R' = Me
21 R = tert-butyl, R' = Me
22 R = n-butyl, R' = iPr
24 R = toluene, R' = H
25 R = toluene, R' = Me
26 R = 1,2-dibenzyl, R' = H
27 R = 1,2-dibenzyl, R' = Me



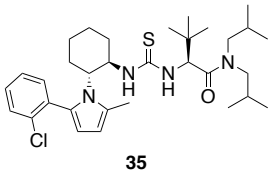
23 R = n-pentyl, R' = n-pentyl, R" = Me, R"' = H
30 R = iso-butyl, R' = iso-butyl,R" = H, R"' = H
31 R = iso-butyl, R' = iso-butyl, R" = Et, R"'= H
33 R = iso-butyl, R' = iso-butyl, R" = Me, R"' = OMe
34 R = iso-butyl, R' = iso-butyl, R" = Me, R"' = F



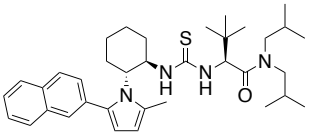
28 R = Me
29 R = tert-butyl



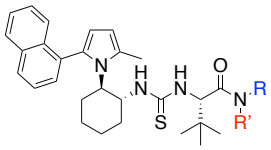
32



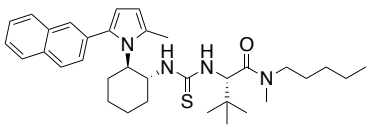
35



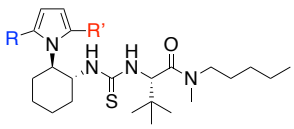
36



37 R = iso-butyl, R' = iso-butyl
39 R = n-pentyl, R' = Me



38



40 R = p-OMeBn, R" = Me
41 R = p-BnF, R" = Me
42 R = Me, R' = Me
43 R = Ph, R" = Ph
44 R = Ph, R" = H

Scheme 12: Experimentally tested catalysts for the 90 reactions under study.[497,498]

For this data set most of the reactions have an average $\Delta\Delta G^{\ddagger}$ ranging from 4 kJ/mol to 7 kJ/mol). A few reactions perform particularly well with $\Delta\Delta G^{\ddagger} > 8$ kJ/mol, while around 20 show low selectivity, with $\Delta\Delta G^{\ddagger} < 3$ kJ/mol (Figure 70). Among the tested substrates (Scheme 11), **6** is the most frequently used (55 reactions), while in terms of catalysts (Scheme 12), **1** is the most prevalent. The substrates show high *Tanimoto* similarity indicating structural similarities; the lowest average *Tanimoto* index (0.44) corresponds to substrate **30**, and the highest (0.66) to substrate **6**. On the other hand, catalyst **7** exhibits the lowest average *Tanimoto* index (0.23) and catalyst **20** the highest (0.75). Based on this information, we defined our external test set to comprise 10% of the data, leading to 9 reactions (see datasets in the supplementary material, subdirectory Chapter 4.4).

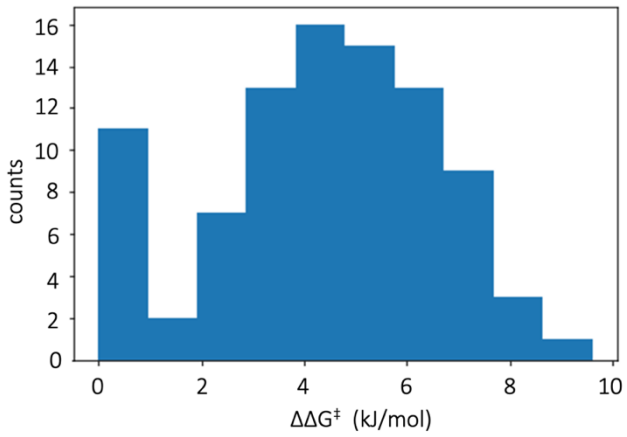


Figure 70: Histogram showing the frequency of occurrences of $\Delta\Delta G^{\ddagger}$ values for the experimentally tested reactions. Only a few reactions perform well ($\Delta\Delta G^{\ddagger} > 8$ kJ/mol).

To describe the catalysts and substrates, we investigated two types of descriptors, Morgan fingerprints and DFT descriptors generated based on Model 5b (§ 4.2.2). In the DFT model, catalysts and substrates are treated separately, although the reaction involves a transfer of chloride from the substrate to the catalyst. To accurately capture this transition, we introduced a chloride anion coordinated to the catalyst and a chlorine atom linked to the substrate. To identify the most important DFT descriptors we employed LASSOCV. Mordred descriptors were considered initially; however, they were disregarded due to poor performance (supplementary material - subdirectory Chapter 4.4).

The same ML algorithms and evaluation metrics used in §4.3 are considered here. Most regressors demonstrated limited predicting power when applied to the external test set with the DFT descriptors, therefore, only the results from LASSOCV are presented here. For classification, a binary approach was employed, with class 1 representing $\Delta\Delta G^{\ddagger} >= 4$ kJ/mol and class 0 representing $\Delta\Delta G^{\ddagger} < 4$ kJ/mol. For the 90 reactions in the data set, 34 fall into class

0, and 56 fall into class 1. To address the class imbalance, additional sampling points were generated for the minority class. After the synthetic data points are added, there are in total 112 data points, meaning 22 synthetic samples are used. As previously described in *Pythia* (§3.2), the entire dataset was cross-validated for training, and 9 reactions were reserved for testing. From the 81 remaining reactions, 30 are classified as 0 and 51 as 1. 21 synthetic data are generated, resulting in a training set of 102 data points.

As explained in §4.3 utilizing classification models for predicting $\Delta\Delta G^{\ddagger}$, is not conventional and the inclusion of classification models in this analysis is primarily intended to showcase *Pythia's* output. The focus of this study is on the regression task and more specifically on LASSOCV.

4.4.2. Results and discussion

This section presents the ML models generated to predict selectivity in the Pictet-Spengler cyclisation of hydroxylactams reactions. The models are based on two types of descriptors: Morgan fingerprints and DFT, which were trained with classifiers and LASSOCV. A summary of the best performing models can be found in Table 15.

Table 15: Summary of the best performing algorithms, for the unseen dataset, according to the different descriptor models.

Descriptors	*Model*	*Metrics*
Fingerprints	kNN	accuracy = 0.89
	LASSOCV	R^2 = 0.78, RMSE = 1.11 kJ/mol
DFT	ET	accuracy = 0.83
	LASSOCV	R^2 = 0.76, RMSE = 1.44 kJ/mol

Morgan fingerprints

After removing features represented by 0 across all reactions, each reaction is described by 241 bits. GP and LR present the highest accuracy (0.85) as well as similar MCC (0.70) and g-mean (0.85). Ada Boost follows closely with high accuracy (0.83), MCC (0.66) and g-mean (0.83) (Table 16). The confusion matrices show that while both GP and LR perform well, LR has higher sensitivity with more TPs but also more FPs, while GP has higher specificity with less FP but also less TP predictions. GP also has a higher AUC (0.92) indicating that it learns faster (Appendix D - Figure 100).

Table 16: Performance of the different classifiers when using Morgan fingerprints for the training set.

Classifier	*Accuracy*	*Sensitivity*	*Specificity*	*MCC*	*Precision*	*g-mean*
kNN	0.82	0.91	0.73	0.65	0.77	0.82
GP	0.85	0.89	0.80	0.70	0.82	0.85
DT	0.78	0.77	0.79	0.55	0.78	0.78
ET	0.79	0.91	0.68	0.61	0.74	0.79
Ada Boost	0.83	0.86	0.80	0.66	0.81	0.83
LR	0.85	0.91	0.79	0.70	0.81	0.85

For the test set, GP and Ada Boost exhibit consistent high accuracy (0.89) and excellent specificity and precision, making only one false prediction. They also show high MCC (0.80) and g-mean (0.98) (Table 17). Interestingly, kNN shows similar metrics but slightly higher AUC (0.97, compared to 0.95 for GP and Ada Boost, Appendix D - Figure 101). On the other hand, LR shows lower accuracy (0.78) and is no longer among the top three models (Table 17). Further analysis reveals that LR's lower accuracy is due to one reaction being classified as FP, while GP, Ada Boost and kNN have no FPs. Despite the lower accuracy, the AUC of LR is still high (0.9) indicating that LR is still able to separate well between the classes, although not as well as GP and Ada Boost (Appendix D - Figure 101).

Table 17: Performance of the different classifiers when using Morgan fingerprints for the test set.

Classifier	*Accuracy*	*Sensitivity*	*Specificity*	*MCC*	*Precision*	*g-mean*
kNN	0.89	0.80	1.00	0.80	1.00	0.89
GP	0.89	0.80	1.00	0.80	1.00	0.89
DT	0.67	0.60	0.75	0.35	0.75	0.67
ET	0.78	1.00	0.50	0.60	0.71	0.71
Ada Boost	0.89	0.80	1.00	0.80	1.00	0.89
LR	0.78	0.80	0.75	0.55	0.80	0.77

Finally, LASSOCV shows good correlation and low error both for the training (R^2 = 0.72, RMSE = 1.17 kJ/mol) and the test set (R^2 = 0.78, RMSE = 1.11 kJ/mol, Figure 71). These results indicate that the ML models considering Morgan fingerprints can effectively predict selectivity without over-fitting. However, relying solely on fingerprints presents challenges when attempting to extract chemical insights from them, which is a crucial aspect of this analysis.

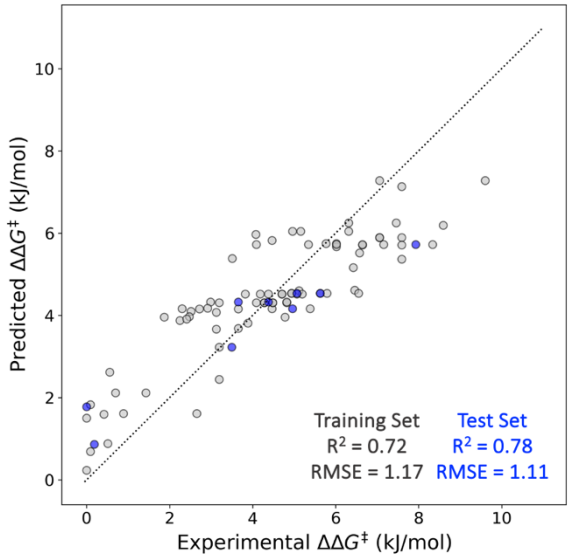


Figure 71: LASSOCV model with Morgan fingerprints. Both the training (in grey) and the test set (in blue) show good correlations and low errors.

DFT descriptors

To describe the reactions using DFT computed descriptors, 60 features are employed for the catalysts and the substrates. The substrates are represented by 18 features, including HOMO, LUMO, N (chlorolactam) and C_2 (indole numbering system) charges, dipole moment, and steric descriptors calculated along the maroon arrows in Figure 72a. The catalysts are represented by 42 features, including HOMO, LUMO, dipole moment, H and Cl^- charges, average NMR shifts for the H and C of the thiourea moiety and the Cl^-, BOs and steric descriptors calculated along the maroon arrows in Figure 72b.

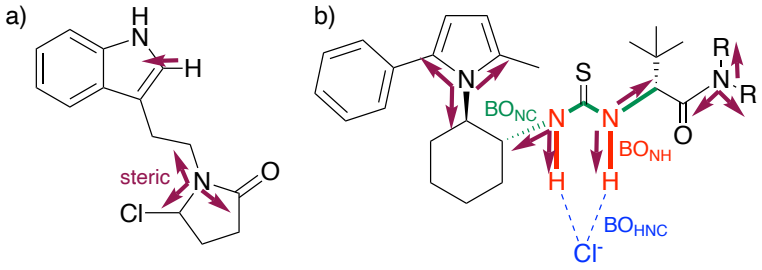


Figure 72: Graphical representation of the descriptors calculated for the reactions under study. a) For the substrates; b) For the catalysts. In green, red, and blue the different BO. In maroon the bond scanned to calculate steric descriptors.

When employing DFT descriptors with classifiers, GP shows the highest performance across all metrics, offering a well-balanced prediction capability for both positive and negative outcomes. It gives an accuracy, sensitivity, specificity, precision, and g-mean of 0.79, and an MCC of 0.57. Ada Boost and LR follow with small differences in their metrics (

Table **18**). The ROC curves show that the three classifiers gain skill fast and can separate well between classes (AUC > 0.80, Appendix D - Figure 102).

Table 18: Performance of the different classifiers when using DFT descriptors for the training set.

Classifier	Accuracy	Sensitivity	Specificity	MCC	Precision	g-mean
kNN	0.71	0.71	0.71	0.42	0.71	0.71
GP	0.79	0.79	0.79	0.57	0.79	0.79
DT	0.71	0.66	0.75	0.41	0.73	0.70
ET	0.71	0.68	0.73	0.41	0.72	0.70
Ada Boost	0.75	0.77	0.73	0.50	0.74	0.75
LR	0.73	0.70	0.77	0.47	0.75	0.73

For the test set, GP performs excellent, kNN, shows good accuracy (0.78) and excellent specificity and precision as it shows no FP predictions (Table 19, Appendix D - Figure 103). ET shows a moderate performance, with a g-mean of 0.67, indicating a fair ability to identify both positive and negative cases correctly. The underperformance of Ada Boost is unexpected; it exhibits the lowest performance among the models, with an accuracy of 0.44 and sensitivity of 0.20. The model seems to struggle in identifying positive cases and maintaining a balance between sensitivity and specificity, as evidenced by the g-mean of 0.39. The MCC score is negative (-0.06), further highlighting the model's difficulties in providing a balance between FP and FN errors (Table 19). Upon further analysis of the prediction scores, it is revealed that the model misclassifies four out of the nine reactions, assigning them with low probabilities.

Table 19: Performance of the different classifiers when using DFT descriptors for the test set.

Classifier	Accuracy	Sensitivity	Specificity	MCC	Precision	g-mean
kNN	0.78	0.60	1.00	0.63	1.00	0.77
GP	1.00	1.00	1.00	1.00	1.00	1.00
DT	0.56	0.40	0.75	0.16	0.67	0.55
ET	0.67	0.60	0.75	0.35	0.75	0.67
Ada Boost	0.44	0.20	0.75	-0.06	0.50	0.39
LR	0.56	0.40	0.75	0.16	0.67	0.55

Finally, the most informative model is generated when using DFT descriptors with LASSOCV. As already discussed, LASSOCV allows for the identification of features influencing the reactions and provides robust models for prediction tasks. Here, the model demonstrates good correlations and low errors both for the training (R^2 = 0.73, RMSE = 1.15 kJ/mol) and the test set (R^2 = 0.71, RMSE = 1.28 kJ/mol). The model exhibits good generalization capabilities, with only a slight decrease in performance from the training set to the test set (Figure 73). While the fingerprints model showed a slightly higher correlation and lower error for the test set (R^2 =

0.78 and RMSE = 1.11 kJ/mol, Figure 71), the model based on DFT descriptors provides a deeper understanding of the underlying patterns in the data. By analyzing the important coefficients derived from LASSOCV, we obtained Eq. 4.4, where descriptors for the substrates are shown in blue and the descriptors for the catalysts and temperature are shown in black (Figure 74).

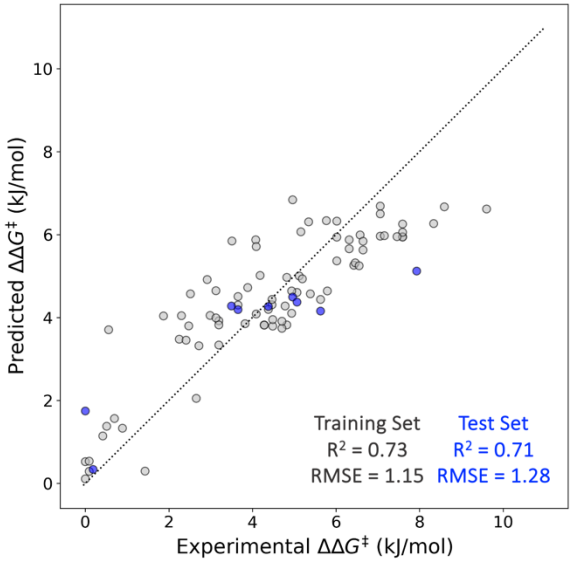


Figure 73: DFT model with physical chemical descriptors of the lowest energy conformation for the substrates and catalysts at the PBE-D3BJ/def2-SVP level of theory (Model 5b). In grey the training set (90% of the data) shows good correlation and low error. In blue the unseen test set (10% of the data) confirms the predictive power of the model.

$$\Delta\Delta G^{\ddagger} = 1.78 + 193N_2\text{charge} + 61C_{19}\text{charge} + 1C_{19}H_{27}\,(B_1) + 0.15N_1C_{42}(\text{L}) + 0.2N_{26}C_{50}(B_5) + 0.1N_{26}C_{51}(B_5) - 0.02\text{Temp} \quad (4.4)$$

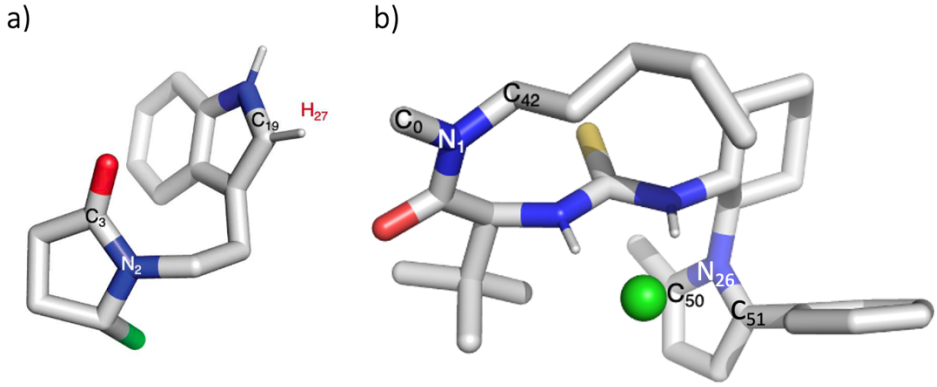


Figure 74: Representation of the important atoms that according to Eq.4.4 influence the $\Delta\Delta G^{\ddagger}$ a) For the substrates; b) For the catalysts.

From Eq. 4.4, the charges of the N_2 and the indolic/pyrrolic C_{19} substrate centers play an important role in determining the $\Delta\Delta G^{\ddagger}$. The large coefficients should not confuse the reader as the charges' values are very small (≈ -0.05) and therefore the actual effect is much smaller.

The N_2 is involved in stabilizing the iminium cation (Figure 74a), while the C_{19} charge descriptor corresponds to the nucleophilic center. Notably, even if the descriptors do not consider the species as ionic couples (catalyst-substrate adduct), the model identified the most important atoms involved in the cyclization step and the charge modifications associated to them. Finally, the B_1 descriptor for $C_{19}H_{27}$, representing the steric parameter orthogonal to the bond C−H, correlates with the substitutions on the indole/pyrrole highlighting that presence of protecting groups on the nearby N atom (i.e., TIPS groups) can increase selectivity.

For the catalysts, N_1C_{42} (L), $N_{26}C_{50}$ (B_5) and $N_{26}C_{51}$ (B_5) are important in determining selectivity (Eq. 4.4, Figure 74b). Feature N_1C_{42} (L) is related to the dimensions of the substituents on the N of the amide, defining the chiral space. Features $N_{26}C_{50}$ and $N_{26}C_{51}$, correspond to the substitutions on the pyrrole core, which have been found to increase enantioselectivity.[497,498] It is interesting to see that our model recover these two most relevant parts of the catalyst defining the chiral space, suggesting that fine-tunning these two regions dramatically influences the stereochemical outcome. Finally, the temperature is an important parameter, as increasing temperatures has a negative effect on *ee*.

Conclusions

In conclusion, we employed Morgan fingerprints and DFT descriptors to predict selectivity in the Pictet-Spengler cyclisation of hydroxylactams. While Morgan fingerprints slightly outperformed DFT descriptors, the difference in accuracy was marginal, both for classifiers and regressors; therefore, we further investigated the DFT models to gain chemical insights. This reaction is an intramolecular reaction where the anion is delivered from the substrate to the catalyst. Thus, the catalyst simply prepares a chiral environment around the substrate, where the intramolecular reaction takes place. The analysis of the important descriptors agrees with mechanistic features of this reaction, despite it having no knowledge of the reaction pathway. For instance, the substrate charges play a major role in determining selectivity, which agrees with the fact that both descriptors are associated to atoms involved in the enantiodetermining cyclization step. Additionally, the substituents on the amide moiety of the catalyst are fundamental in defining the chiral space and induce selectivity. Interestingly in the Strecker's reaction (§4.3) similar descriptors were found to define selectivity, suggesting the existence of similar interactions between the amide, the chloride, and the iminium-cation intermediate.

4.5. Summary and conclusions

In this chapter, we investigated three distinct HB organocatalytic reactions, including the enantioselective formation of β-fluoroamines, Strecker synthesis of α-amino acids, Pictet-Spengler cyclisation of hydroxylactams. The objective was to establish a general and efficient workflow for predicting selectivity.

For the enantioselective formation of β-fluoroamines, we evaluated the effect of using either fingerprints or DFT descriptors (e.g., charges, HOMO, LUMO and steric effects), on the accuracy and efficiency of our models. The utilization of DFT descriptors offers interpretability advantages that are absent when employing fingerprints. We investigated DFT descriptors at different levels of theory and found that more expensive computational approaches do not necessarily translate into superior predictive performance. Our final regression model, Model 5b, employing LASSOCV and DFT descriptors, yielded RMSE ≈ 1 kJ/mol and allowed for the identification of crucial electronic and steric effects governing selectivity (Figure 58, Figure 59, Figure 61). For example, electron withdrawing groups on the BINAM enhance selectivity, which aligns with experimental observations. Based on these results new catalysts were suggested which remain to be tested.

Our protocol, using Model 5b, was subsequently applied to predict enantioselectivity in the Strecker synthesis of α-amino acids. For this system we obtained slightly higher error than before (RMSE ≈ 1.44 kJ/mol), but still good correlation (R^2 = 0.76). We hypothesize this is likely due to the smaller number of experimentally tested reactions. The ability to stabilize the iminium ion, measured by the charge on the N_1 of the substrate (Figure 69a), was found to be the main factor affecting selectivity, consistent with both experimental observations and previous computational analyses.

Finally, we investigated the Pictet-Spengler cyclisation of hydroxylactams, a chemically distinct and less computationally explored reaction. Model 5b yielded RMSE ≈ 1.28 kJ/mol and captured the crucial chemistry underlying this reaction for example, substitutions on the N of the amide, of the catalyst, define the chiral space (Figure 74b). For the last two reactions, we explored several regression and classification methods, utilizing both fingerprints and Mordred descriptors, in addition to DFT-computed descriptors. Mordred descriptors exhibited lower accuracy and higher errors compared to fingerprints and DFT descriptors.

Overall, this chapter established a comprehensive framework for *ee* prediction in HB-organocatalysis providing valuable interpretations to improve our understanding of catalytic systems and aid their further design.

Our work builds on previous works in the field by experts such as Sigman,[97,98] Sunoj,[136] and Doyle[132,133] which have used ML models to predict *ee* for nucleophilic additions to imines,[97] regioselectivity for difluorination reactions of alkenes,[136] and reaction yields for alcohol deoxyfluorination reactions[132] (as discussed in detail in §1.3). This work distinguishes itself in several aspects. Firstly, it targets the enantioselectivity of HB reactions, which has been studied using DFT models by Paton and coworkers[488,489] but not studied using ML. Secondly, while we use similar descriptors to those employed by Sigman and Doyle, we demonstrate that low-level DFT computed descriptors at the minima are sufficient, and there is no need to employ TS descriptors as done by Sigman. Drawing inspiration from Sigman and Sunoj's efforts on interpretability, we developed models that prioritize interpretability, avoiding those that lack true feature learning and instead rely on capturing patterns in the data.[134] In contrast to other works, which focused on specific protocols for specific reactions, we successfully applied our methodology to three distinct reactions without modifications. We demonstrated our methodoly's robustness and generalizability, paving the way for broader applications in catalyst design. Because of the variation in datasets, descriptors, and ML algorithms across these studies, direct comparison is not feasible, however, it is evident that each model excels in predicting the reactions it was trained on.

5. Predicting viscosity and CO_2 solubility in ionic liquids with graph neural networks

This chapter showcases the implementation of graph neural networks (GNNs) for predicting the viscosity and CO_2 solubility of ionic liquids (ILs). ILs exhibit extraordinary properties and hold immense potential across diverse applications. However, accurately predicting their physicochemical characteristics, including viscosity and solubility, poses a formidable challenge due to the complex dependence of these properties on temperature and pressure. The ideal IL should have low viscosity and high CO_2 solubility and predicting these properties early on is essential for future developments.

While several ML models exist for IL property predictions, many of them lack sufficient testing or use small datasets for training, which limits their ability to extrapolate to structurally diverse ILs. Others require extensive manual work, such as laborious identification of group contributions. Here, we present a GNN-based model that accurately predicts viscosity and solubility. The model is built using published data sets on viscosity[230] and solubility,[241] and the architecture is inspired by the work of Mitsos *et al*.[391] We provide an automated, end-to-end approach for predicting properties of ILs, which enables robust predictions and facilitates advancements in IL research.

5.1. Data collection

The datasets presented in this chapter have been cleaned and curated to remove duplicates and inconsistencies introduced in previous publications. The datasets are available in the supplementary material (directory Chapter 5 - Data)

Viscosity dataset

For viscosity, 15,213 data points were gathered from literature[230] at different temperatures (253 K – 573 K) and atmospheric pressure (1 bar). The data set covers a wide range for viscosity values (0.66 mPa s – 77,441 mPa s). To make this property more amenable to ML, we focused on predicting the natural logarithm of viscosity (ln η), which follows a Gaussian distribution ranging from −0.22 to 11.26 (Figure 75a-b). The temperature data were normalized using min-max scaling (Figure 75c-d).

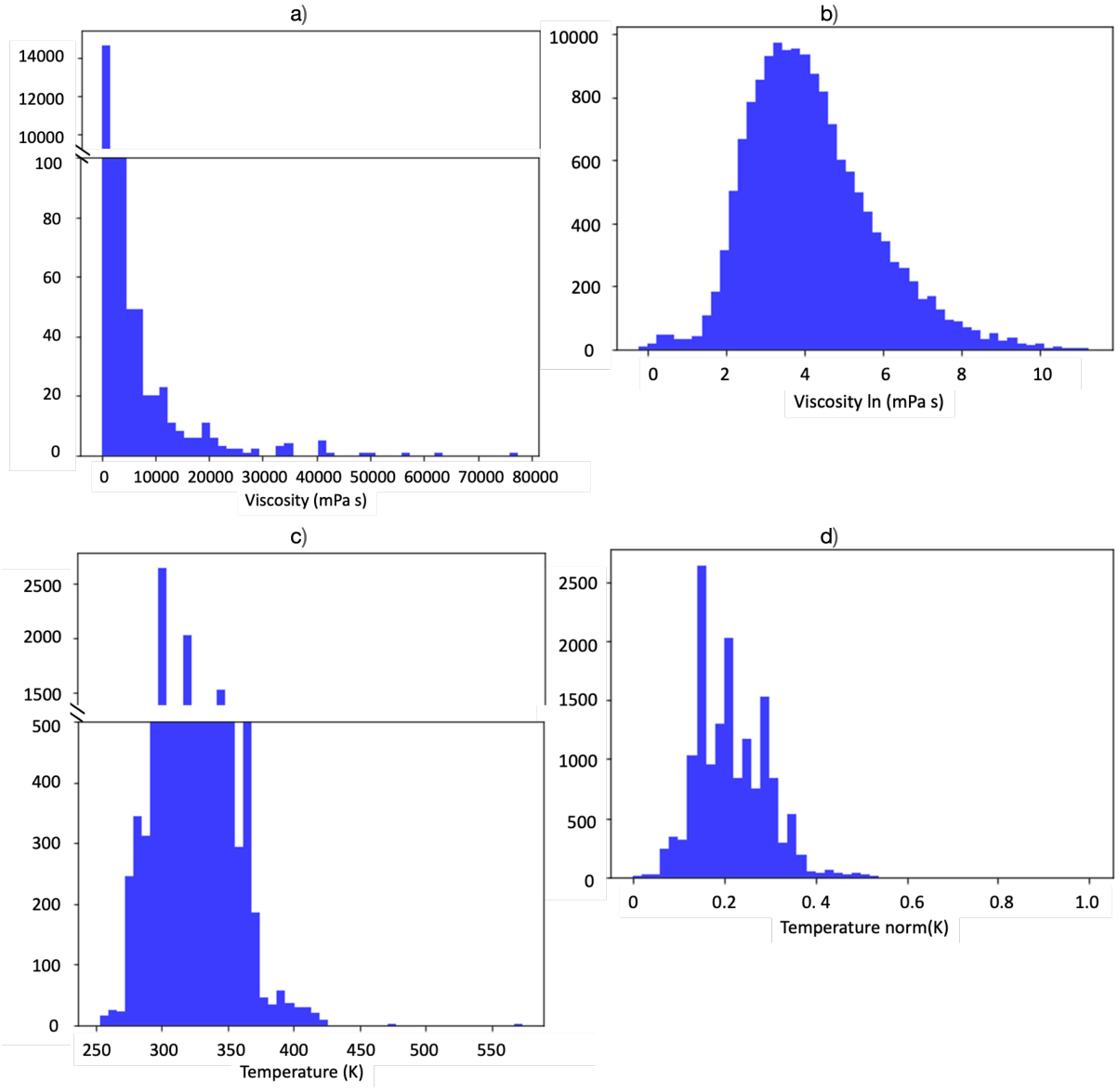


Figure 75: Scaling of the viscosity and temperature as they present high variations in their values. a) For the viscosity values the natural logarithm has been applied and once scaled the data follow the gaussian distribution. b) The temperature data have been normalized with min-max scaling.

The viscosity data include 24 different cationic families and 15 different anionic families (Table 20 and Scheme 13). The most well-represented cationic family is the imidazolium, while tetrazolium only occurs once. For the anions, the most well-represented family is the $[TF_2N]^-$, while borates are the most underrepresented.

We utilized *Tanimoto* similarity index to assess the structural similarities between ILs with high viscosity (>2,046 mPa s, 34 datapoints at an average temperature of 302 K) and low viscosity (<10 mPa s, 1,333 datapoints at an average temperature of 350 K). Analysis of the anions and cations of the 34 datapoints with high viscosity, show low *Tanimoto* similarity

indices, with none exceeding 0.37 and 0.45, respectively[4]. For the anions, sulfonates were the most frequently occurring family (53%), followed by the inorganics family (30%). The cations belonged to either imidazolium, ammonium, or phosphonium families. From the 1,333 datapoints with low viscosity the most frequently occurring anionic family was the carboxylates (30%), followed by the pyrrolidinium family (29%). Notably the imidazolium family remains the most popular cation (43%). From this analysis it can be speculated that sulfonate anions result in high viscosities, while carboxylate anions may contribute to lower viscosities in ILs.

Table 20: Viscosity data set. The cations and anions have been categorized in families. Each of the families is present in several ILs.

Cation Family	*Number of IL*	*Anion Family*	*Number of IL*
amidium	16	alkoxides	124
ammonium	2061	aminoacids	375
azepanium	167	BF_4 derivatives	1069
bicyclic	127	carboxylates	1500
cyclic amidium	14	dicyanamides	811
cyclic phosphonium	4	heterocyclic amines	510
cyclic sulfonium	60	inorganics	1106
cyclopropanium	263	metal complexes	340
guanidinium	209	methanides	172
imidazolium	6426	TF_2N derivatives	5540
morpholinium	336	organic borates	95
oxazolidinium	12	PF_6 derivatives	1043
phosphonium	1780	phosphates	431
piperazinium	5	sulfates	1008
piperidinium	554	sulfonates	1183
pyrazolium	108		
pyridinium	1444		
pyrrolidinium	1162		
quinolinium	58		
sulfonium	217		
tetrazolium	1		
thiazolium	37		
thiouronium	87		
triazolium	159		
Total Families	**24**	**Total Families**	**15**

Solubility dataset

For solubility, a total of 9,499 data points were collected at various temperatures (243 K – 453 K) and pressures (0.008 bar – 500 bar) from literature.[241] CO_2 solubility is measured in molar fraction, which is a unitless quantity ranging from 0 to 1. A total of 8 different cationic families

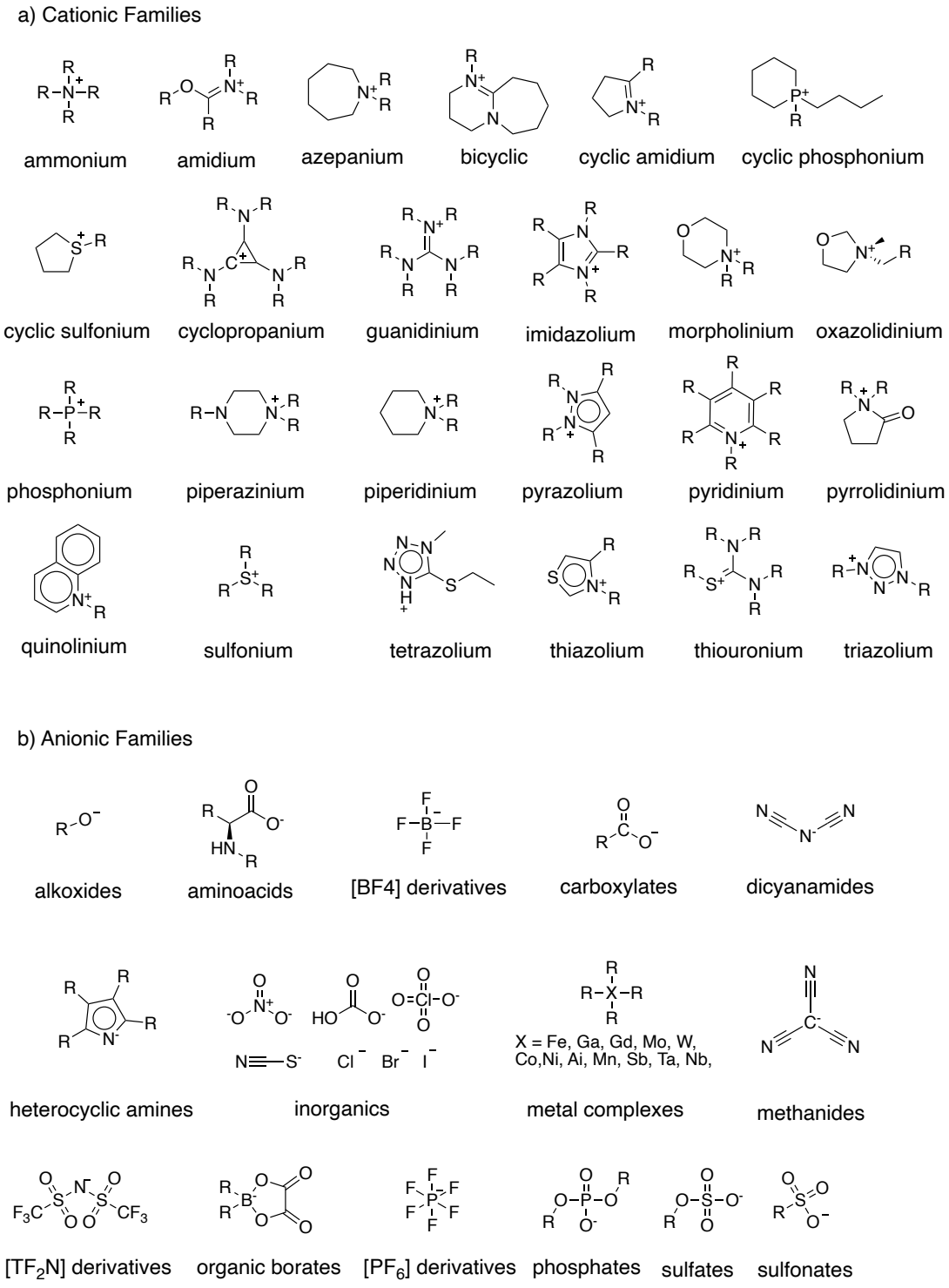
[4] We note here that the similarity index value ranges from 0 to 1, with 0 being no similarity at all and 1 being the same molecule.

and 10 different anionic families are present in the solubility dataset (Table 21 and Scheme 13). The most well-represented cationic family is the imidazolium, while the most underrepresented is the piperidinium. For the anions, the most well-represented family is the $[TF_2N]^-$, and the most underrepresented is the phosphates.

We utilized *Tanimoto* similarity index to assess the structural similarities between ILs with high solubility (>=0.9, 17 datapoint at average temperature of 312 K and average pressure of 110 bar), and low solubility (<0.01, 306 datapoints at an average temperature of 317 K and average pressure of 0.31 bar). For the high solubility category, we identified 17 datapoints. The anions belong to the $[BF_4]^-$ family (15/17) and the $[TF_2N]^-$ family (2/17). The cations belonged to the imidazolium family (15/17) and to the pyridinium family (2/17). For the low solubility category, we identified 306 ILs. The families for both anions and cations were diverse, and we hypothesize that the low solubilities might be related to low pressures.

Table 21: Solubility data. The cations and anions have been categorized in families. Each of the families is present in several ILs.

Cation Family	*Number of IL*	*Anion Family*	*Number of IL*
ammonium	411	BF_4 derivatives	1134
imidazolium	7472	carboxylates	278
phosphonium	533	dicyanamides	220
piperidinium	36	inorganics	523
pyrazolium	75	methanides	776
pyridinium	156	TF_2N derivatives	4072
pyrolidinium	777	PF_6 derivatives	929
sulfonium	39	phosphates	87
		sulfates	411
		sulfonates	1069
Total Families	**8**	**Total Families**	**10**



Scheme 13: Structural representation of the cationic and anionic families that exist in our datasets.

Training and test set creation

To test the prediction accuracy of our model, we use 20% of the total data set as an unseen test set. Additionally, we apply a 90% - 10% random split for the training/validation data set to the remaining data. We repeat 40 times taking the average metrics for each split, to ensure generalizability (Figure 76a). We will be referring to this model as *Generalization* model.

To evaluate the extrapolation capability to molecular structures, we performed another split, into a training/validation sets and test set. This time, the test set includes molecules that have not be seen by the training/validation set. Specifically, for the test set, we randomly select 5% of the unique SMILES of both anions and cations[5], while the remaining data points are used as the training/validation set. Analogously to the test set, we randomly select 5% of unique molecules from the training/validation set to create the validation set. Therefore, the three sets contain ILs not included in the other sets. Note that for each random training/validation split in the extrapolation analysis, the number of data points in the validation set typically varies because the number of anion/cation molecules involved may vary for different IL (Figure 76b). We refer to this model as *Extrapolation* model.

Figure 76: Train - validation -test set splits a) for the Generalization model where the same ILs might exist in all sets under different conditions, and b) for the Extrapolation model where only unique ILs exist in the test sets.

Finally, we employ ensemble learning, a technique that aggregates predictions from multiple models. Ensembles can increase the robustness of prediction models, by averaging out under- and over-predictions.[402,499,500] Specifically, we randomly split the data not used for testing into

[5] For viscosity, there are 1,107 unique SMILES, leading to 55 unique SMILES in the test set, leading to 2,499 data points containing these SMILES. For solubility there are 84 unique SMILES in total, leading to 4 unique SMILES in the test set, leading to 167 data points containing these SMILES.

a training and validation sets before the training of each model. After training, the outputs of all models are averaged to obtain the reported property prediction. We found that the validation MAE tends to stabilize after including 40 models.

5.2. GNN architecture

As explained in §2.2.2 the first step in developing a GNN is to transform the molecules into graphs. Nodes (atoms) are denoted with $v \in V$, and their feature vector as $f^V(v)$, while edges (bonds) connecting two nodes v, w are denoted with $e_{vw} \in E$ and their feature vector as $f^E(e_{vw})$. The set of nodes and edges with their corresponding feature vectors describes the attributed molecular graph $G(m) = \{V,E,f^V,f^E\}$ for a molecule m.[392,393] To generate the feature vectors, the atom and bond features shown in Table 22 are generated with RDKit.[391,392,501]

Table 22: Features used for the nodes and edges to create the feature vector for each molecule.

Feature	*Description*	*Dimension*
Atom type	C,O,N,F,S,Cl,P,B,Br,Al,Sb,I, H,Si,Fe,Ta,Nb,W,Mo	19
Atom in ring	whether the atom is part of a ring	1
Is aromatic	whether the atom is part of an aromatic system	1
Charge	formal charge of the atom (-3,-2,-1, 0, 1,2,3)	6
Hybridization	sp, sp2, sp3, or sp3d2	4
#Hs	number of bonded hydrogen atoms	4
Bond type	single, double, triple, or aromatic	4
Conjugated	whether the bond is conjugated	1
Bond in ring	whether the bond is part of a ring	1

The GNN model employed here is implemented in Python and utilizes the geometric deep learning package PyTorch Geometric (PyG) developed by Fey and Lenssen.[502] After converting the molecules into graphs and establishing their feature vectors, they are given as an input to the GNN. In the message passing phase of the GNN, we use two separate graph convolutional layer channels, one for the molecular graph of the cation and one for the anion. Thus, the same graph convolutional layers are applied independently to the molecular graph of the anion and the molecular graph of the cation (Figure 77). For the graph convolutions, we apply a gated recurrent unit (GRU) with the GINE-operator[401,503] that utilizes an MLP_{GINE} to map the ε-scaled hidden state of a node (ε being a learnable parameter) and the received information from the neighborhood (transformed by an activation function σ) to the updated hidden state (h_v^l, for a given node v and layer l), leading to the following update function (Eq. 5.1):

$$h_v^l = GRU\left(h_v^{l-1}, \sigma\left(MLP_{GINE}\left((1+\varepsilon)\cdot h_v^{l-1} + \sum_{w \in N(v)} \sigma(h_w^{l-1} + f_{e_{vw}})\right)\right)\right). \quad (5.1)$$

Note that here both the initial hidden node states and the edge features are linearly transformed by a learnable parameter matrix (θ) to match the dimension of the following hidden states, i.e., $h_v^0 = \theta_v \cdot f^V(v)$ and $f_{e_{vw}} = \theta_E \cdot f^E(e_{vw})$ respectively.

After the graph convolution layers sum pooling is applied, capturing the collective information of the atoms and bonds, yielding the molecular fingerprint of the cation (h_c) and the anion (h_a). Then, the interactions between the cation and anion molecules are modeled with MLP_{IL}, a MLP that transforms and concatenates the two molecular fingerprints. The output of this interaction MLP is then concatenated with the normalized temperature, and pressure of the IL and subsequently fed into MLP_{TP}, an MLP providing the prediction for viscosity or solubility (Figure 77).

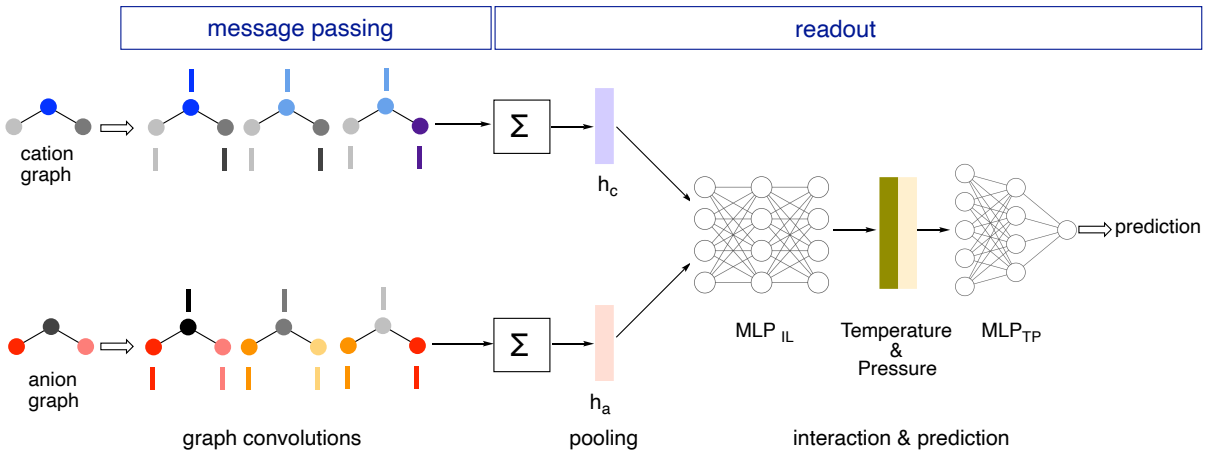


Figure 77: Graphical representation of the GNN architecture. Figure adapted from Mitsos *et al.*[391]

Hyperparameter optimizations

Following the work of Mitsos *et al.*, the hyperparameters were tuned in a two-step process.[391] In the first step, a grid search is performed to determine the optimal hyperparameters for the GNN architecture, including the graph convolutional type ∈ {NNConv, GINEConv}, number of graph convolutional layers ∈ {1, 2, 3}, usage of GRU in graph convolutions ∈ {True, False}, dimension of molecular fingerprint ∈ {64, 128}, number of layers in MLP-channels in interaction network ∈ {1, 2, 3}, activation function ∈ {Leaky ReLU, ReLU, ELU}.

The number of neurons for the interaction MLP_{IL} is fixed to 256 for all MLP-channel layers. The structure of the MLP_{TP} is not varied and set to three layers with 258 (two additional dimension for the temperature and pressure), 128, and 1 neurons. The following training hyperparameters are applied: initial learning rate 0.001, learning rate decay of 0.8 with a patience of 3 epochs, batch size 64, maximum number of epochs 100 for viscosity and 120 for solubility, optimizer adam, early stopping patience of 25 epochs, dropout rate in both MLPs of 0.05.

The first step of the hyperparameter search results in a final model architecture with the graph convolutional type GINEConv employed in two layers in combination with a GRU, a fingerprint dimension of 64, a number of layers in MLP-channels of 3, and Leaky ReLU as activation function. In the second step of the hyperparameter tuning, a grid search to fine-tune the GNN training parameters is conducted, i.e., varying the initial learning rate $\in$ {0.01, 0.001, 0.0001}, the batch size $\in$ {32, 64, 128}, and the dropout rate $\in$ {0.1, 0.05, 0}. We select the best model based on the validation error with a random split of the initial data set into training and validation sets. This leads to an optimal initial learning rate of 0.001, a batch size of 64, and a dropout rate of 0. In Figure 78 we present the loss curves that helped us identify the number of epochs needed to train the models.

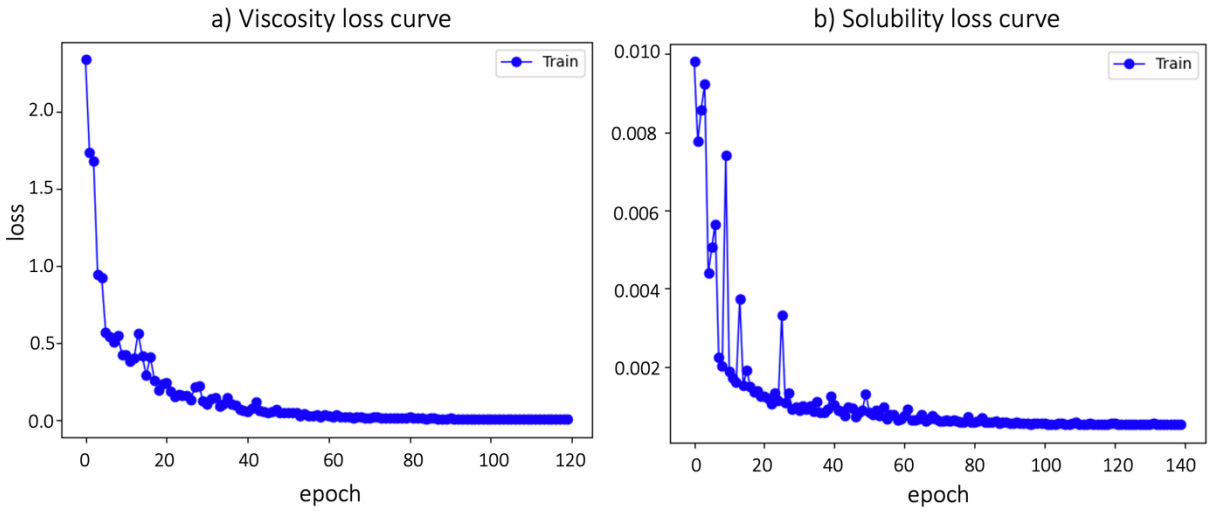


Figure 78: Loss curves depicting the training epochs required for datasets. a) The viscosity dataset shows minimal variations post the 80th epoch, with no further change after 100 epochs. b) The solubility dataset exhibits slight fluctuations after the 100th epoch, stabilizing thereafter with no significant change after 120 epochs.

We also tested a single graph GNN where the anion and cation are represented by a one graph (Appendix E1). However, this approach did not yield better accuracies than previous viscosity prediction models. This led us to construct the two graph GNN. It is also worth noting that we initially trained a LASSOCV model with Morgan fingerprints, for the viscosity dataset. This

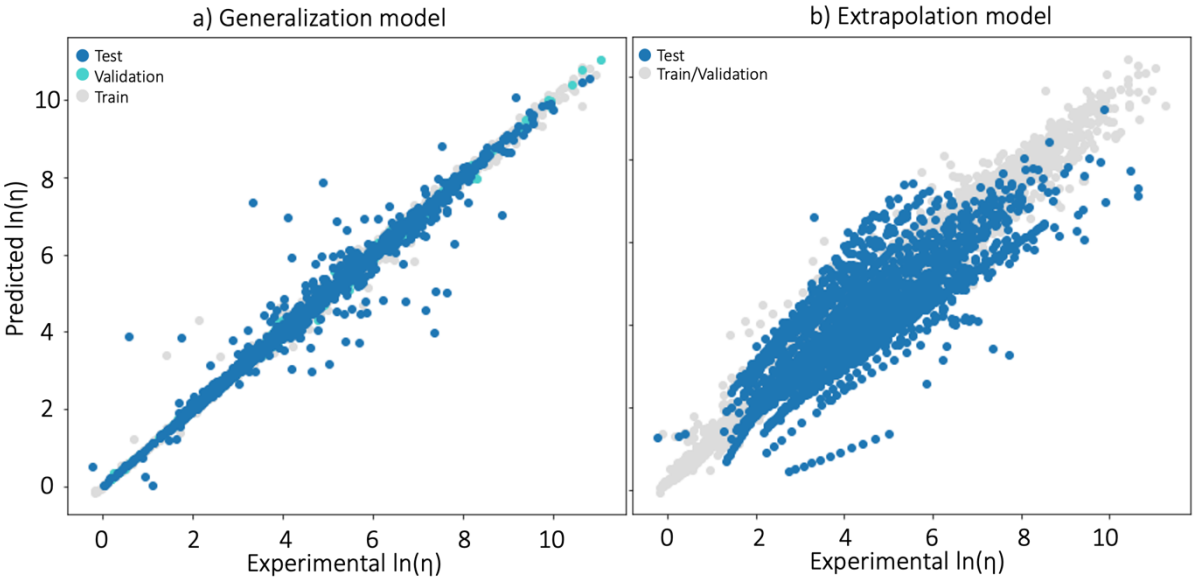
was an early attempt to establish a baseline model for our data set. However, it was discarded as it proved to be less accurate than previously published models (Appendix E2).

5.3. Results and discussion

5.3.1. Viscosity

For the *Generalization* model the dataset was split randomly into training (70% of the data), validation (10% of the data), and test (20% of the data). This was done 40 times, and the average metrics were obtained. The results demonstrate high predictive power as evident by the excellent correlation coefficient and low errors for the test set (R^2 = 0.98, MAE = 0.07 and RMSE = 0.21, Figure 79a).

The *Extrapolation* model, exhibits worse correlation and higher errors (R^2 = 0.62, MAE = 0.71 and RMSE = 0.92) for the test set compared to the train/validation set (R^2 = 0.97, MAE = 0.18 and RMSE = 0.29, Figure 79b). However, we attribute this difference to the fact that the test set contains structurally diverse and unseen molecules, compared to the training set. Nonetheless, the model still demonstrates its extrapolation capabilities. We do not believe that this correlation indicates over-fitting, instead it underscores the moderate yet significant extrapolation capabilities of our model.



Average metrics for 40 models

Data set	*MAE*	*RMSE*	R^2
Train	0.04	0.07	0.99
Validation	0.04	0.07	0.99
Test	0.07	0.21	0.98

Ensemble metrics for 40 models

Data set	*MAE*	*RMSE*	R^2
Train/Validation	0.18	0.29	0.97
Test	0.71	0.92	0.62

Figure 79: Viscosity results for a) The Generalization model that shows excellent correlation and low errors; b) the Extrapolation model that shows higher errors and worse correlation, displaying clusters of outliers.

To understand the poorer performance of the *Extrapolation* model we investigated further. We identified clusters of outliers in the test set, which we color coded according to anion and cation families (Figure 80). The bottom blue/pink clusters correspond to alkoxides and phosphonium respectively. While both families are well represented in our dataset, the combination of the two is less so (in total only 71 datapoints contain alkoxides and phosphonium). For the middle clusters no clear family trends are observed. Finally, the top cluster contains guanidinium and cyclopopanium families which are underrepresented in general. These findings suggest that these outliers arise because of the limited data available for them. We have no evidence that the temperature plays a significant role for these outliers.

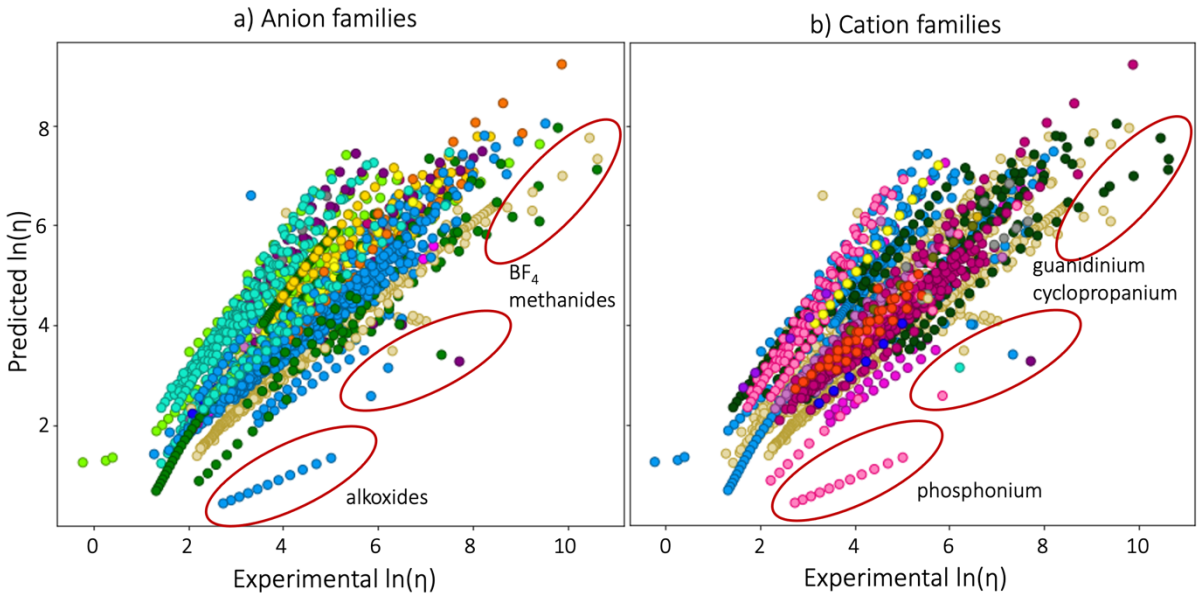


Figure 80: Color coded families help us identify in which families the outliers belong to; a) For the anions we observe that alkoxides, BF_4^-, and methanides form the main outliers. b) For the cations the phosphonium, the guanidinium and the cyclopropenium form the main outliers.

Upon careful examination of Figure 80, it seems that higher viscosity values deviate more. This observation is expected due to the scarcity of high viscosity values in our dataset, resulting in greater variability in the predictions for such cases. Based on this observation, we conducted a final test to evaluate the model's performance in predicting extreme values. We used the five ILs with the lowest and highest viscosity as an unseen test set. Although the correlation is high ($R^2 = 0.85$), we observe high errors (MAE = 1.7, RMSE = 2.16). These errors arise primarily from the highest viscosity data points. For example, the largest viscosity value 11.26 was predicted as 8.2. This indicates that our model faces challenges in accurately predicting high viscosity values, as already observed in Figure 80.

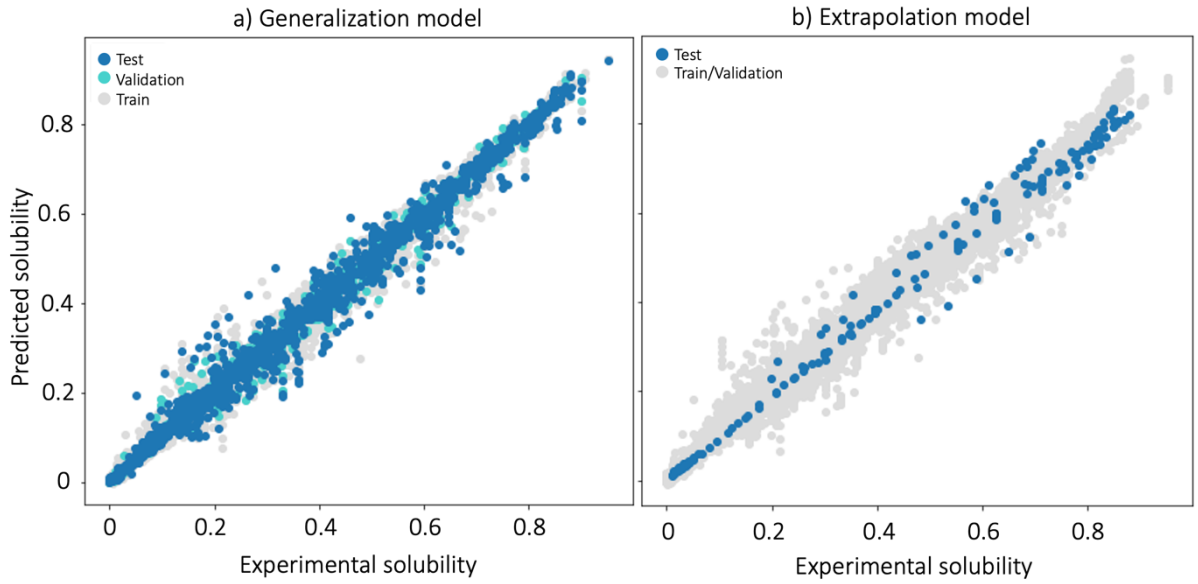
In conclusion, these results suggest that while the *Generalization* model outperforms those previously reported in the literature (as detailed in §1.4.2), the *Extrapolation* model could

benefit from further refinement. It's important to underscore, that the *Extrapolation* model has been tested on a diverse set of molecular structures. This represents a significant departure from prior studies in this field, which often did not challenge their models with such structurally varied test sets. As such, our work not only pushes the boundaries of model testing but also illuminates avenues for future improvements in model robustness and generalizability. It is important to acknowledge that our current model lacks consideration of pressure variations. Future efforts should include development of a model that accounts for variations in pressure.

5.3.2. CO_2 Solubility

Similarly to viscosity, the solubility dataset was randomly split 40 times to training validation and test sets and the average metrics were obtained to construct the *Generalization* model, which resulted in a highly predictive model as it displays excellent correlation and low errors for the test set (R^2 = 0.99, MAE = 0.02 and RMSE = 0.03, Figure 81a).

The *Extrapolation* model also exhibits perfect correlation and very low errors both for the train/validation set (R^2 = 0.98, MAE = 0.02 and RMSE = 0.03) and the test set (R^2 = 0.98, MAE = 0.03 and RMSE = 0.04, Figure 81b), despite the highly different structural molecules included in the test set compared to the train set.



Average metrics for 40 models

Data set	*MAE*	*RMSE*	R^2
Train	0.01	0.02	0.99
Validation	0.01	0.02	0.99
Test	0.02	0.03	0.99

Ensemble metrics for 40 models

Data set	*MAE*	*RMSE*	R^2
Train/Validation	0.02	0.03	0.98
Test	0.03	0.04	0.98

Figure 81: Solubility results. a) The Generalization model and b) The Extrapolation model show excellent correlation and very low errors.

To further challenge the solubility model, we used as a test set 436 data points that contain the two anions and the three cations with the lowest average *Tanimoto* similarity. Pleasingly the correlation stays high and the errors low for this test set (R^2 = 0.88, MAE = 0.04 and RMSE = 0.04, Table 23). Next, we used as a test set the six anions that show the lowest average *Tanimoto* similarity, which are present in 2,376 data points. In this case we observe a lower correlation (R^2 = 0.66) compared to the previous models; however, the errors continue to be small (MAE = 0.06 and RMSE = 0.11, Table 24), which implies that even when the model is pushed to predict highly diverse structures it still gives accurate predictions. Additionally, we investigated how the model predicts extreme values, meaning the five highest and lowest solubility values, yielding an excellent correlation and low errors (R^2 = 0.99, MAE = 0.02, RMSE = 0.03, Table 25). These results confirm the robustness of our model.

Table 23: Metrics for the solubility model when the test set contains ILs composed by the two anions and the three cations with the lowest *Tanimoto* similarity, resulting in 436 structurally diverse datapoints.

Data set	*MAE*	*RMSE*	R^2
Train	0.02	0.02	0.98
Validation	0.01	0.02	0.99
Test	0.04	0.06	0.88

Table 24: Metrics for the solubility model when the test set contains ILs composed by the six anions with the lowest *Tanimoto* similarity, resulting in 2,376 structurally diverse datapoints.

Data set	*MAE*	*RMSE*	R^2
Train	0.02	0.03	0.98
Validation	0.02	0.02	0.99
Test	0.06	0.11	0.66

Table 25: Metrics for the solubility model when the test set contains data points responsible for the five highest and five lowest solubility values.

Data set	*MAE*	*RMSE*	R^2
Train	0.02	0.03	0.98
Validation	0.02	0.02	0.99
Test	0.02	0.03	0.99

A final test was performed to evaluate our model's ability to predict the CO_2 solubility of 24 experimentally tested ILs that have been used for carbon capture.[171] These ILs were tested under a range of experimental conditions, from temperatures of 298K to 323K and pressures from 9 bar to 29.5 bar. This diverse test set not only demonstrates the real-world applicability of our model, but also confirms its effectiveness within the specific context of IL research and carbon capture applications. The results obtained are highly encouraging. With a good correlation and low errors (R^2 = 0.70, MAE = 0.04, and RMSE = 0.07, Table 26), our model shows considerable promise for future use in carbon capturing.

Table 26: Metrics for the solubility model when the test set contains 24 ILs that have been tested for carbon capture capacity.

Data set	*MAE*	*RMSE*	*R2*
Train	0.01	0.02	0.99
Validation	0.01	0.02	0.99
Test	0.04	0.07	0.70

Interpretability

We attempted to interpret our predictions, employing GNNExplainer[504] and XGNN.[505] The former provides input-dependent explanations at an instant level, monitoring the change of prediction with respect to different input and determines input importance scores. XGNN[505] identifies the general decision-making patterns and structures within the GNN model. However, we were unable to apply these explaining tools to our model. As it is built on two distinct graphs representing anions and cations separately, while these techniques are designed for homogenous graphs. Moreover, these techniques are not well-suited for regression tasks such as the ones we have used.

One potential alternative approach would involve constructing a second GNN, similar to the work conducted by Farimani *et al*., which employs a single graph and performs classification tasks.[242] However, it is important to note that this approach would be limited in terms of explaining the predictions of our regression-based model. As the two models serve different prediction purposes, utilizing a classification based GNN to explain the predictions of a regression model may not provide a comprehensive understanding of the factors influencing the regression model's predictions.

To interpret our GNN model, we tried training a RF, as they are known for their good performance and interpretability, though they have not been used for IL property predictions.[132,133,140] We used the feature vectors for the anions and cations from the pooling step (previously mentioned as h_c and h_a in Figure 77), along with temperature and pressure information. Although this method does not identify chemical groups, it can discern wether important features are related to the anion or cation. Notably, temperature and pressure stood out as key solubility descriptors. The other 10 top features belonged to the anion. We would like to note here that the RF model for solubility shows worse correlation than the GNN and slightly higher errors (R^2 = 0.88, MAE = 0.05, RMSE = 0.08). Unfortunately, the RF for viscosity showed bad correlation and high errors (R^2 = 0.32, MAE = 1.04, RMSE = 1.37) so it was not considered further.

This evidence the urgent need for tools explaining more complex GNN architectures, which could significantly increase the impact of the studies under this architecture.

5.4. Conclusions

Our study introduced a GNN model to train two independent models for predicting viscosity and CO_2 solubility of ILs. We report that our model for viscosity yielded $R^2 = 0.98$ and MAE = 0.07 and for solubility $R^2 = 0.99$ and MAE = 0.02, for the test sets. We evaluated the generalization and extrapolation capabilities of our GNN models by testing them on unseen and dissimilar molecules. The trained models discussed in this chapter are accessible in the supplementary material (Chapter 5). Once formally published, researchers will have the opportunity to utilize these models for their own predictive analyses.

While other models have been developed to predict viscosity, including QSPR and group contribution (GC) models paired with regression algorithms, many of these models were trained on small datasets and/or validated on limited unseen data. As an exception Paduszynski *et al.,*[230] employed the largest dataset available to date (41,250 data points). They developed a two-stage modeling protocol involving a NN-derived reference term and a SVM for temperature correction. Our aim was therefore to develop an end-to-end model for viscosity prediction, without the need to model reference and correction terms or define numerous group contributions.

Other models have also been trained to predict CO_2 solubility of ILs, using NN-GC and SVM-GC models, however, their testing has been confined to a single randomized train-test dataset split. This limitation prompts questions regarding the models' predictive capacity and ability to generalize to unseen data. Zhou *et al.*[241] compiled a comprehensive database of 10,116 CO_2 solubility data, however, during our study over 600 data points were found to be duplicated. Despite this, it stands as the largest dataset available (which we also employ), encompassing publications up to 2020. The authors acknowledged the inherent challenges associated with constructing a GC model, as it necessitates the prior decomposition of molecules into constituent building groups, a process that typically requires manual intervention.

In our approach, we represent anions and cations as graphs, a technique previously utilized only by Farimani.[242] They applied a similar method to a solubility dataset (originally from Zhou *et al*.[241]), achieving remarkable metrics (MAE = 0.01 and $R^2 = 0.99$), similar to ours. However, their random split of training and testing sets raises questions about the model generalizability.

Additionally, our attempts to adapt their model to our dataset were unsuccessful. They convert their data into a NumPy file format containing only numerical values, which are then used by the GNN. However, they don't provide a consistent method for this conversion, implying potential limitations in its application to diverse datasets. Furthermore, our models run in about 20 minutes on a 2 GHz Quad-Core Intel Core i5 MacBook Pro, whereas their model takes more than a day, most probably due to the architecture they have chosen to employ.

Future research could be directed towards improving the accuracy of the viscosity model, by expanding the dataset to include a wider range of IL structures and incorporating information about pressure. Additionally, achieving interpretability with GNNExplainer, could extend its applicability. Moreover, we envision the development of a general model capable of simultaneously predicting viscosity, solubility, density, and absorption capacity of ILs. Instead of optimizing individual properties researchers can aim for ILs that exhibit a combination of desirable properties. However, the challenges of such an approach might include balancing competing objectives and ensuring that improvements in one property do not negatively impact others, which demands understanding of the IL behavior and structure-property relationships.

6. Conclusions

Significant progress in computer science has propelled the field of computational chemistry, revolutionizing traditional modeling approaches and offering a compelling alternative to time- and resource-intensive experiments. In recent decades, ML models have been extensively developed and applied in various subfields of chemistry for property and reaction outcome predictions.

In this Thesis we have devised *Pythia*, a user-friendly and adaptable framework designed to streamline the application of ML in chemistry. It brings together several ML algorithms and representation methods. *Pythia* integrates feature elimination techniques and interpretability tools such as SHAP to enhance model interpretability. To our knowledge there is no other open-source tool designed to offer such capabilities while at the same time prioritizes accessibility to novice users. Initially used for selectivity predictions, *Pythia* has evolved to incorporate new functionalities such as protein-ligand binding affinity predictions and property estimations, demonstrating its adaptability and versatility across diverse chemistry domains. The unique combination of features and functionalities make *Pythia* a powerful and accessible ML toolkit, empowering researchers to leverage ML techniques in their chemistry-oriented endeavors.

In Chapter 4, we showcased the use of *Pythia* in predicting selectivity in three organocatalytic reactions, demonstrating accurate predictions in regression models using LASSOCV and DFT descriptors. Predicting enantioselectivity of HB reactions, using ML techniques, is an unexplored area. We investigated the enantioselective formation of β-fluoramines (RMSE ≈1 kJ/mol), the enantioselective Strecker synthesis of α-amino acids (RMSE ≈ 1.44 kJ/mol), and the Pictet-Spengler cyclisation of hydroxylactams (RMSE = 1.28 kJ/mol). We show that even low level DFT descriptors are sufficient for predicting $\Delta\Delta G^{\ddagger}$. While previous studies have utilized similar techniques to predict selectivity, our work stands out for its broader application across diverse systems. Moreover, for the enantioselective formation of β-fluoramines, we designed new catalysts taking into consideration our model's suggestion to add electron withdrawing groups on the BINAM. Indeed, these changes have been experimentally proven to improve selectivity, as they increase the acidity of the urea and consequently the strength of the HBs.

In Chapter 5 we explored the use of GNNs for predicting viscosity and CO_2 solubility in ILs. These systems have emerged as promising candidates for carbon capture technologies; however, the effective design and implementation of such technologies necessitates a thorough understanding of their viscosity and solubility. To address this gap, we developed a comprehensive end-to-end workflow for the accurate prediction of these properties. The models developed showcase exceptional correlation with experimental data, with minimal errors (for viscosity $R^2 = 0.98$ and MAE = 0.07, and for solubility $R^2 = 0.99$ and MAE = 0.02). Our models will be made accessible through pre-trained formats, facilitating their adoption by researchers seeking to forecast these properties for their specific IL compositions. Moreover, the same architecture could be adapted to predict additional properties such as density and absorption capacity in ILs, thereby broadening its utility and impact within the materials science domain. Future research could focus on developing more interpretable models to understand the molecular features that govern viscosity and solubility. The inclusion of uncertainty quantification methods is also a relevant aspect necessary to further enhance the accuracy and reliability of predictions. These advancements will not only enhance our understanding of IL properties but also bolster their practical application in tackling pressing environmental challenges.

Overall, this Thesis serves as a comprehensive exploration of ML workflows in computational chemistry, including organocatalysis and the design of carbon capture technologies based on ILs. The development of *Pythia* streamlines applications of ML in chemistry. As ML models continue to advance in terms of their robustness, reliability, and popularity, we hope experimental chemists will increasingly and rigorously employ these models, challenging their use in novel chemical systems, and identifying new avenues for further improvement.

7. References

(1) Sterling, A. J.; Zavitsanou, S.; Ford, J.; Duarte, F. Selectivity in Organocatalysis—From Qualitative to Quantitative Predictive Models. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2021**, *11*, e1518.

(2) Mitchell, T. *Machine Learning*; 1997.

(3) Hastie, T.; Tibshirani, R.; Friedman, J. *Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction*; 2009.

(4) Judith, H.; Daniel, K. *Machine Learning for Dummies*; 2018.

(5) Theodoridis, S.; Koutroumbas, K. *Pattern Recognition*; 2006.

(6) Theodoridis, S.; York, N.; Diego, S. *Machine Learning A Bayesian and Optimization Perspective*; 2015.

(7) Todhunter, I. *A History of the Mathematical Theory of Probability: From the Time of Pascal to That of Laplace*; 2022.

(8) Gillies, D. *Philosophical Theories of Probability*; Taylor and Francis, 2012.

(9) Dale, A. *A History of Inverse Probability: From Thomas Bayes to Karl Pearson*; 2012.

(10) Sheynin, O. B. Early History of the Theory of Probability. *Arch. Hist. Exact Sci.* **1977**, *17* (3), 201–259.

(11) Stigler, S. Gauss and the Invention of Least Squares. *Ann. Stat.* **1981**, *9* (3), 465–474.

(12) Galton, F. Typical Laws of Heredity. *Nature* **1877**, *15* (388), 492–495.

(13) Galton, F. One Vote, One Value. *Nature* **1907**, *75* (1948), 414–414.

(14) Varberg, D. E. The Development of Modern Statistics. *Source Math. Teach.* **1963**, *56* (4), 252–257.

(15) Stigler, S. M. Francis Galton's Account of the Invention of Correlation. *Stat. Sci.* **1989**, *4* (2), 73–79.

(16) Pearson, K. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *London, Edinburgh, Dublin Philos. Mag. J. Sci.* **1901**, *2* (11), 559–572.

(17) Pearson, K. X. On the Criterion That a given System of Deviations from the Probable in the Case of a Correlated System of Variables Is Such That It Can Be Reasonably Supposed to Have Arisen from Random Sampling. *London, Edinburgh, Dublin Philos. Mag. J. Sci.* **1900**, *50* (302), 157–175.

(18) Pearson, K. DAS FEHLERGESETZ UND SEINE VERALLGEMEINER-UNGEN DURCH FECHNER UND PEARSON. *Biometrika* **1905**, *4*, 169–212.

(19) Vidal, R.; Ma, Y.; Sastry, S. S. Principal Component Analysis. *Interdiscip. Appl. Math.* **2016**, *40*, 25–62.

(20) Aldrich, J. R.A. Fisher and the Making of Maximum Likelihood 1912-1922. *Stat. Sci.* **1997**, *12* (3), 162–176.

(21) Pirie, W. *Spearman Rank Correlation Coefficient*; John Wiley & Sons, Ltd, 2006.

(22) Weisstein, E. W. *Student's t-Distribution*. http://mathworld.wolfram.com/Studentst-Distribution.html.

(23) Neyman, J. On the Two Different Aspects of the Representative Method: The Method of Stratified Sampling and the Method of Purposive Selection. *J. R. Stat. Soc.* **1992**, *97*, 123–150.

(24) Turing, A. M. On Computable Numbers, with an Application to the Entscheidungs Problem. *Proc. London Math. Soc.* **1936**, 230–265.

(25) McCulloch, W. S.; Pitts, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bull. Math. Biophys.* **1943**, *5* (4), 115–133.

(26) Hebb, D. O. *The Organization of Behavior*; Psychology Press, 2002.

(27) Turing, A. M. Computing Machinery and Intelligence. *Mind* **1950**, *LIX*, 433–460.

(28) Harnad, S. On Turing (1950) on Computing, Machinery, and Intelligence. **2008**.

(29) Schaeffer, J. A Gamut of Games. *AI Mag.* **2001**, *22* (3), 29.

(30) Samuel, A. L. Eight-Move Opening Utilizing Generalization Learning. *IBM J.* **1959**, *3* (3), 210–229.

(31) Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychol. Rev.* **1958**, *65* (6), 386–408.

(32) Schmidhuber, J. *Deep Learning*; Springer, Boston, MA, 2017.

(33) Cover, T, P. H. Nearest Neighbor Pattern Classification. *IEEE Trans. Inf. Theory* **1967**, *13* (1), 21–27.

(34) Werbos, P. *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting*; 1994.

(35) Ho, T. Random Decision Forests. In *Proceedings of 3rd international conference*; 1995.

(36) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45* (1), 5–32.

(37) Hinton, G.; Osindero, S, Teh, Y. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* **2006**, *18*, 1527–1554.

(38) Koul, A.; Ganju, S.; Kasam, M. *Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras & TensorFlow*; 2019.

(39) Moravec, H. The Stanford Cart and the CMU Rover. *Proc. IEEE* **1983**, *71* (7), 872–884.

(40) Hsu, F. IBM's Deep Blue Chess Grandmaster Chips. *IEEE Micro* **1999**, *19* (2), 70–81.

(41) Watson, I. *IBM Watson: How it works*.

(42) Rob, H. *The Era of Cognitive Systems: An inside Look at IBM Watson and How It Works*; 2012.

(43) Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J.; Blum, M.; Hutter, F. Efficient and Robust Automated Machine Learning. *Adv. Neural Inf. Process. Syst.* **2015**, *18*, 1527–1554.

(44) Yao, Q.; Wang, M.; Chen, Y.; Dai, W.; Li, Y.; Tu, W.; Yang, Q.; Yu, Y. Taking Human out of Learning Applications: A Survey on Automated Machine Learning. *arXiv* **2018**, 1810.13306.

(45) Hutter, F.; Kotthoff, L.; Vanschoren, J. *Automated Machine Learning: Methods, Systems, Challenges*; 2019.

(46) Goh, G. B.; Hodas, N. O.; Vishnu, A. Deep Learning for Computational Chemistry. *J. Comput. Chem.* **2017**, *38* (16), 1291–1307.

(47) Mater, A. C.; Coote, M. L. Deep Learning in Chemistry. *J. Chem. Inf. Model.* **2019**, *59*, 2545–2559.

(48) Janet, J. P.; Kulik, H. J. *Machine Learning in Chemistry*; ACS In Focus; 2020.

(49) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine Learning for Molecular and Materials Science. *Nature* **2018**, *559* (7715), 547–555.

(50) Tkatchenko, A. Machine Learning for Chemical Discovery. *Nat. Commun.* **2020**, *11*, 4125.

(51) Cova, T.; Pais, A. Deep Learning for Deep Chemistry: Optimizing the Prediction of Chemical Patterns. *Front. Chem.* **2019**, *7*, 809.

(52) Schwaller, P.; Vaucher, A. C.; Laplaza, R.; Bunne, C.; Krause, A.; Corminboeuf, C.; Laino, T. Machine Intelligence for Chemical Reaction Space. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2022**, *12* (5).

(53) Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. Quantum-Chemical Insights from Deep Tensor Neural Networks. *Nat. Commun. 2017 81* **2017**, *8*, 13890.

(54) Ramakrishnan, R.; Lilienfeld, O. A. von. Machine Learning, Quantum Chemistry, and Chemical Space. *Rev. Comput. Chem.* **2017**, *30*, 225–256.

(55) Oishi, A.; Yagawa, G. Computational Mechanics Enhanced by Deep Learning. *Comput. Methods Appl. Mech. Eng.* **2017**, *327*, 327–351.

(56) Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. DeepTox: Toxicity Prediction Using Deep Learning. *Front. Environ. Sci.* **2016**, *3*, 80.

(57) Strieth-Kalthoff, F.; Sandfort, F.; Segler, M. H. S.; Glorius, F. Machine Learning the Ropes: Principles, Applications and Directions in Synthetic Chemistry. *Chem. Soc. Rev.* **2020**, *49* (17), 6154–6168.

(58) Pflüger, P. M.; Glorius, F. Molecular Machine Learning: The Future of Synthetic Chemistry? *Angew. Chemie Int. Ed.* **2020**, *59* (43), 18860–18865.

(59) Coley, C. W.; Green, W. H.; Jensen, K. F. Machine Learning in Computer-Aided Synthesis Planning. *Acc. Chem. Res.* **2018**, *51* (5), 1281–1289.

(60) Granda, J. M.; Donina, L.; Dragone, V.; Long, D. L.; Cronin, L. Controlling an Organic Synthesis Robot with Machine Learning to Search for New Reactivity. *Nature* **2018**, *559* (7714), 377–381.

(61) de Almeida, A. F.; Moreira, R.; Rodrigues, T. Synthetic Organic Chemistry Driven by Artificial Intelligence. *Nat. Rev. Chem.* **2019**, *3* (10), 589–604.

(62) Debus, B.; Parastar, H.; Harrington, P.; Kirsanov, D. Deep Learning in Analytical Chemistry. *Trends Anal. Chem.* **2021**, *145*, 116459.

(63) Schweidtmann, A. M.; Esche, E.; Fischer, A.; Kloft, M.; Repke, J. U.; Sager, S.; Mitsos, A. Machine Learning in Chemical Engineering: A Perspective. *Chemie Ing. Tech.* **2021**, *93* (12), 2029–2039.

(64) Jing, Y.; Bian, Y.; Hu, Z.; Wang, L.; Xie, X.-Q. S. Deep Learning for Drug Design: An Artificial Intelligence Paradigm for Drug Discovery in the Big Data Era. *AAPS J.* **2018**, *20* (3), 58.

(65) Vamathevan, J.; Clark, D.; Czodrowski, P.; Dunham, I.; Ferran, E.; Lee, G.; Li, B.; Madabhushi, A.; Shah, P.; Spitzer, M.; Zhao, S. Applications of Machine Learning in Drug Discovery and Development. *Nat. Rev. Drug Discov.* **2019**, *18*, 463–477.

(66) Lavecchia, A. Machine-Learning Approaches in Drug Discovery: Methods and Applications. *Drug Discov. Today* **2015**, *20*, 318–331.

(67) Zhang, L.; Tan, J.; Han, D.; Zhu, H. From Machine Learning to Deep Learning: Progress in Machine Intelligence for Rational Drug Discovery. *Drug Discov. Today* **2017**, *22*, 1680–1685.

(68) Lo, Y.; Rensi, S.; Torng, W.; Altman, R. Machine Learning in Chemoinformatics and Drug Discovery. *Drug Discov. Today* **2018**, *23*, 1538–1546.

(69) Kulik, H. J.; Tiwary, P. Artificial Intelligence in Computational Materials Science. *MRS Bull.* **2022**, *47* (9), 927–929.

(70) Schmidt, J.; Marques; Botti, S. Recent Advances and Applications of Machine Learning in Solid-State Materials Science. *npj Comput. Mater.* **2019**, *5*, 83.

(71) Wei, J.; Chu, X.; Sun, X.; Xu, K.; Deng, H.; Chen, J.; Wei, Z.; Lei, M. Machine Learning in Materials Science. *Wiley Online Libr.* **2019**, *1* (3), 338–358.

(72) Baum, Z. J.; Yu, X.; Ayala, P. Y.; Zhao, Y.; Watkins, S. P.; Zhou, Q. Artificial Intelligence in Chemistry: Current Trends and Future Directions. *J. Chem. Inf. Model.* **2021**, *61* (7), 3197–3212.

(73) *CAS Content | CAS*. https://www.cas.org/about/cas-content (accessed 2023-04-12).

(74) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Nature, M. F.-; 2021, U. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature* **2021**, *596*, 583–589.

(75) Ramsundar, B.; Eastman, P.; Walters, P.; Pande, V. *Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More*; O'Reilly Media, 2019.

(76) Schwaller, P.; Laino, T.; Gaudin, T.; Bolgar, P.; Hunter, C. A.; Bekas, C.; Lee, A. Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Publ.* **2019**, *5* (9), 1572–1583.

(77) Schwaller, P.; Gaudin, T.; Lanyi, D.; Bekas, C.; Science, T. L.-C.; 2018, U. "Found in Translation": Predicting Outcomes of Complex Organic Chemistry Reactions Using Neural Sequence-to-Sequence Models. *Chem. Sci.* **2018**, *9*, 6091–6098.

(78) Schwaller, P.; Petraglia, R.; Zullo, V.; Nair, V. H.; Haeuselmann, R. A.; Pisoni, R.; Bekas, C.; Iuliano, A.; Laino, T. Predicting Retrosynthetic Pathways Using Transformer-Based Models and a Hyper-Graph Exploration Strategy. *Chem. Sci.* **2020**, *12*, 3316–3325.

(79) Toniato, A.; Schwaller, P, Cardinale, A. Unassisted Noise Reduction of Chemical Reaction Datasets. *Nat. Mach. Intell.* **2021**, *3*, 485–494.

(80) Alain C. Vaucher, Philippe Schwaller, Joppe Geluykens, Vishnu H. Nair, A. I. & T. L. Inferring Experimental Procedures from Text-Based Representations of Chemical Reactions. *Nat. Commun.* **2021**, *12*, 2573.

(81) Alain C. Vaucher, Federico Zipoli, Joppe Geluykens, Vishnu H. Nair, P. S. & T. L. Automated Extraction of Chemical Synthesis Actions from Experimental Procedures. *Nature* **2020**, *11*, 3601.

(82) Savage, N. *Tapping into the drug discovery potential of AI*. Nature. https://www.nature.com/articles/d43747-021-00045-7.pdf (accessed 2023-01-24).

(83) Hammett, L. P. Some Relations between Reaction Rates and Equilibrium Constants. *Chem. Rev.* **1935**, *17* (1), 125–136.

(84) Taft, R. W. Linear Free Energy Relationships from Rates of Esterification and Hydrolysis of Aliphatic and Ortho-Substituted Benzoate Esters. *J. Am. Chem. Soc.* **1952**, *74* (11), 2729–2732.

(85) Taft, R. W.; Lewis, I. C. Evaluation of Resonance Effects on Reactivity by Application of the Linear Inductive Energy Relationship. V. Concerning a ΣR Scale of Resonance Effects1,2. *J. Am. Chem. Soc.* **1959**, *81* (20), 5343–5352.

(86) Charton, M. Steric Effects. I. Esterification and Acid-Catalyzed Hydrolysis of Esters. *J. Am. Chem. Soc.* **1975**, *97* (6), 1552–1556.

(87) Tropsha, A. Best Practices for QSAR Model Development, Validation, and Exploitation. *Mol. Inform.* **2010**, *29*, 476–488.

(88) E, N, Muratov, J, Bajorath, R, P., Sheridan, I, V, Tetko, D, Filimonov, V, Poroikov, T, I. Oprea, I, I. Baskin, A, Varnek, A, Roitberg, O, Isayev, S, Curtalolo D, Fourches, Y, Cohen, A, Aspuru-Guzi, A, Cherkasov, A, T. QSAR without Borders. *Chem. Soc. Rev.* **2020**, *49* (11), 3525–3564.

(89) Santiago, C. B.; Guo, J. Y.; Sigman, M. S. Predictive and Mechanistic Multivariate Linear Regression Models for Reaction Development. *Chem. Sci.* **2018**, *9* (9), 2398–2412.

(90) Dearden, J. C.; Cronin, M. T. D.; Kaiser, K. L. E. How Not to Develop a Quantitative Structure–Activity or Structure–Property Relationship (QSAR/QSPR). *SAR QSAR Environ. Res.* **2009**, *20*, 241–266.

(91) Sigman, M. S.; Harper, K. C.; Bess, E. N.; Milo, A. The Development of Multidimensional Analysis Tools for Asymmetric Catalysis and Beyond. *Acc. Chem. Res.* **2016**, *49* (6), 1292–1301.

(92) Harper, K. C.; Bess, E. N.; Sigman, M. S. Multidimensional Steric Parameters in the Analysis of Asymmetric Catalytic Reactions. *Nat. Chem.* **2012**, *4* (5), 366–374.

(93) Bess, E. N.; Bischoff, A. J.; Sigman, M. S.; Jacobsen, E. N. Designer Substrate Library for Quantitative, Predictive Modeling of Reaction Performance. *Proc. Natl. Acad. Sci. U. S. A.* **2014**, *111* (41), 14698–14703.

(94) Milo, A.; Neel, A. J.; Dean Toste, F.; Sigman, M. S. A Data-Intensive Approach to Mechanistic Elucidation Applied to Chiral Anion Catalysis. *Science (80-.).* **2015**, *347*, 737–743.

(95) Milo, A.; Bess, E.; Sigman, M. Interrogating Selectivity in Catalysis Using Molecular Vibrations. *Nature* **2014**, *507*, 210–214.

(96) Miller, J. J.; Sigman, M. S. Quantitatively Correlating the Effect of Ligand-Substituent Size in Asymmetric Catalysis Using Linear Free Energy Relationships. *Angew. Chemie Int. Ed.* **2008**, *47* (4), 771–774.

(97) Reid, J.; Sigman, M. Holistic Prediction of Enantioselectivity in Asymmetric Catalysis. *Nature* **2019**, *571*, 343–348.

(98) Reid, J. P.; Proctor, R. S. J.; Sigman, M. S.; Phipps, R. J. Predictive Multivariate Linear Regression Analysis Guides Successful Catalytic Enantioselective Minisci Reactions of Diazines. *J. Am. Chem. Soc.* **2019**, *141* (48), 19178–19185.

(99) Toyao, T.; Maeno, Z.; Takakusagi, S.; Kamachi, T.; Takigawa, I.; Shimizu, K. I. Machine Learning for Catalysis Informatics: Recent Applications and Prospects. *ACS Catal.* **2020**, *10* (3), 2260–2297.

(100) Williams, W. L.; Zeng, L.; Gensch, T.; Sigman, M. S.; Doyle, A. G.; Anslyn, E. V. The Evolution of Data-Driven Modeling in Organic Chemistry. *ACS Cent. Sci.* **2021**, *7*, 1622–1637.

(101) Keith, J. A.; Vassilev-Galindo, V.; Cheng, B.; Chmiela, S.; Gastegger, M.; Müller, K. R.; Tkatchenko, A. Combining Machine Learning and Computational Chemistry for Predictive Insights into Chemical Systems. *Chem. Rev.* **2021**, *121* (16), 9816–9872.

(102) Artrith, N.; Butler, K. T.; Coudert, F. X.; Han, S.; Isayev, O.; Jain, A.; Walsh, A. Best Practices in Machine Learning for Chemistry. *Nat. Chem. 2021 136* **2021**, *13* (6), 505–508.

(103) Bender, A.; Schneider, N.; Segler, M.; Patrick Walters, W.; Engkvist, O.; Rodrigues, T. Evaluation Guidelines for Machine Learning Tools in the Chemical Sciences. *Nat. Rev. Chem.* **2022**, *6*, 428–442.

(104) Mansouri, K.; Grulke, C. M.; Judson, R. S.; Williams, A. J. OPERA Models for Predicting Physicochemical Properties and Environmental Fate Endpoints. *J. Cheminform.* **2018**, *10*.

(105) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing Learned Molecular Representations for Property Prediction. *ACS Publ.* **2019**, *59*, 3370−3388.

(106) *GitHub - chemprop/chemprop: Message Passing Neural Networks for Molecule Property*

Prediction. https://github.com/chemprop/chemprop (accessed 2023-04-12).

(107) Wei, J. N.; Duvenaud, D.; Aspuru-Guzik, A. Neural Networks for the Prediction of Organic Chemistry Reactions. *ACS Cent. Sci.* **2016**, *2* (10), 725–732.

(108) Coley, C. W.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; Jensen, K. F. A Graph-Convolutional Neural Network Model for the Prediction of Chemical Reactivity. *Chem. Sci.* **2019**, *10* (2), 370–377.

(109) Coley, C. W.; Barzilay, R.; Jaakkola, T. S.; Green, W. H.; Jensen, K. F. Prediction of Organic Reaction Outcomes Using Machine Learning. *ACS Cent. Sci.* **2017**, *3* (5), 434–443.

(110) Kovács, D.; McCorkindale, W.; Lee, A. Quantitative Interpretation Explains Machine Learning Models for Chemical Reaction Prediction and Uncovers Bias. *Nature* **2021**, *12*, 1695.

(111) Mann, V.; Journal, V. V. Predicting Chemical Reaction Outcomes: A Grammar Ontology-based Transformer Framework. *AIChE J.* **2021**, *67* (3).

(112) Jin, W.; Coley, C, Barzilay, R, Jaakkola, T. Predicting Organic Reaction Outcomes with Weisfeiler-Lehman Network. In *Advances in Neural Information Processing Systems*; 2017; Vol. 30.

(113) Gao, H.; Struble, T. J.; Coley, C. W.; Wang, Y.; Green, W. H.; Jensen, K. F. Using Machine Learning to Predict Suitable Conditions for Organic Reactions. *ACS Publ.* **2018**, *4*, 1465–1476.

(114) Zhou, Z.; Li, X.; Zare, R. N. Optimizing Chemical Reactions with Deep Reinforcement Learning. *ACS Cent. Sci.* **2017**, *3* (12), 1337–1344.

(115) Shields, B. J.; Stevens, J.; Li, J.; Parasram, M.; Damani, F.; Alvarado, J. I. M.; Janey, J. M.; Adams, R. P.; Doyle, A. G. Bayesian Reaction Optimization as a Tool for Chemical Synthesis. *Nature* **2021**, *590*, 89–96.

(116) Shim, E.; Kammeraad, J.; Xu, Z.; Tewari, A.; Cernak, T. Predicting Reaction Conditions from Limited Data through Active Transfer Learning. *Chem. Sci.* **2022**, *13*, 6655–6668.

(117) Elton, D.; Boukouvalas, Z.; Fugea, M.; Chunga, P. Deep Learning for Molecular Design-a Review of the State of the Art. *Mol. Syst. Des. Eng.* **2019**, *3*, 828–849.

(118) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse Molecular Design Using Machine Learning:Generative Models for Matter Engineering. *Science (80-.).* **2018**, *361*, 360–365.

(119) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52* (7), 1757–1768.

(120) Irwin, J. J.; Shoichet, B. K. ZINC-A Free Database of Commercially Available Compounds for Virtual Screening. *J. Chem. Inf. Model.* **2005**, *45*, 177–182.

(121) Ruddigkeit, L.; Van Deursen, R.; Blum, L. C.; Reymond, J. L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *J. Chem. Inf. Model.* **2012**, *52* (11), 2864–2875.

(122) Ramakrishnan, R.; Dral, P.; Rupp, M.; Lilienfeld, O. Von. Quantum Chemistry Structures and Properties of 134 Kilo Molecules. *Sci. Data* **2014**, *1*.

(123) Bento, A.; Gaulton, A.; Hersey, A.; Bellis, L. The ChEMBL Bioactivity Database: An Update. *Nucleic Acids Res.* **2014**, *42* (D1), D1083–D1090.

(124) Gaulton, A.; Bellis, L.; Bento, A.; Chambers, J. ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40* (D1), D1100–D1107.

(125) Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A. The ChEMBL Database in 2017. *Nucleic Acids Res.* **2017**, *45* (D1), D945–D954.

(126) Tomberg, A.; Johansson, M. J.; Norrby, P. O. A Predictive Tool for Electrophilic Aromatic

Substitutions Using Machine Learning. *J. Org. Chem.* **2019**, *84* (8), 4695–4703.

(127) Neel, A.; Hilton, M.; Sigman, M.; Toste, F. Exploiting Non-Covalent π Interactions for Catalyst Design. *Nature* **2017**, *543*, 637–646.

(128) Foscato, M.; Jensen, V. R. Automated in Silico Design of Homogeneous Catalysts. *ACS Catal.* **2020**, *10* (3), 2354–2377.

(129) Zahrt, A. F.; Athavale, S. V.; Denmark, S. E. Quantitative Structure-Selectivity Relationships in Enantioselective Catalysis: Past, Present, and Future. *Chem. Rev.* **2020**, *120* (3), 1620–1689.

(130) Jorner, K.; Tomberg, A.; Bauer, C.; Sköld, C. Organic Reactivity from Mechanism to Machine Learning. *Nat. Rev. Chem.* **2021**, *5*, 240–255.

(131) Yang, W.; Fidelis, T.; WH, S. Machine Learning in Catalysis, from Proposal to Practicing. *ACS Omega* **2019**, *5*, 83–88.

(132) Nielsen, M. K.; Ahneman, D. T.; Riera, O.; Doyle, A. G. Deoxyfluorination with Sulfonyl Fluorides: Navigating Reaction Space with Machine Learning. *J. Am. Chem. Soc.* **2018**, *140*, 5004–5008.

(133) Ahneman, D. T.; Estrada, J. G.; Lin, S.; Dreher, S. D.; Doyle, A. G. Predicting Reaction Performance in C–N Cross-Coupling Using Machine Learning. *Science (80-.).* **2018**, *360*, 186–190.

(134) Żurański, A. M.; Martinez Alvarado, J. I.; Shields, B. J.; Doyle, A. G. Predicting Reaction Yields via Supervised Learning. *Acc. Chem. Res.* **2021**, *54* (8), 1856–1865.

(135) Zhao, Y.; Liu, X.; Lu, H.; Zhu, X.; Wang, T.; Luo, G.; Zheng, R. An Optimized Deep Convolutional Neural Network for Yield Prediction of Buchwald-Hartwig Amination. *Chem. Phys.* **2021**, *550*, 111296.

(136) Banerjee, S.; Sreenithya, A.; Sunoj, R. Machine Learning for Predicting Product Distributions in Catalytic Regioselective Reactions. *Phys. Chem. Chem. Phys.* **2018**, *20* (27), 18311–18318.

(137) Hoque, A.; Sunoj, R. Deep Learning for Enantioselectivity Predictions in Catalytic Asymmetric β-C–H Bond Activation Reactions. *Digit. Discov.* **2022**, *1* (6), 926–940.

(138) Singh, S, Sunoj, R. A Transfer Learning Protocol for Chemical Catalysis Using a Recurrent Neural Network Adapted from Natural Language Processing. *Digit. Discov.* **2022**, No. 3, 303–312.

(139) Zahrt, A. F.; Henle, J. J.; Rose, B. T.; Wang, Y.; Darrow, W. T.; Denmark, S. E. Prediction of Higher-Selectivity Catalysts by Computer-Driven Workflow and Machine Learning. *Science (80-.).* **2019**, *363* (6424).

(140) Li, X.; Zhang, S.; Xu, L.; Hong, X. Predicting Regioselectivity in Radical C− H Functionalization of Heterocycles through Machine Learning. *AngeChemie Int.* **2020**, *59* (32), 13253–13259.

(141) Gallarati, S.; Fabregat, R.; Laplaza, R.; Bhattacharjee, S. Reaction-Based Machine Learning Representations for Predicting the Enantioselectivity of Organocatalysts. *Chem. Sci.* **2021**, *12* (20), 6879–6889.

(142) Coley, C. W.; Barzilay, R.; Green, W. H.; Jaakkola, T. S.; Jensen, K. F. Convolutional Embedding of Attributed Molecular Graphs for Physical Property Prediction. *J. Chem. Inf. Model.* **2017**, *57* (8), 1757–1772.

(143) Lei, Z.; Chen, B.; Koo, Y. M.; Macfarlane, D. R. Introduction: Ionic Liquids. *Chem. Rev.* **2017**, *117* (10), 6633–6635.

(144) Walden, P. Ueber Die Molekulargrosse Und Elektrische Leitfahigkeit Einiger Geschmolzenen Salze. *mathnet.ru* **1914**, *8*, 405–422.

(145) Welton, T. Ionic Liquids: A Brief History. *Biophys. Rev.* **2018**, *10* (3), 691–706.

(146) Gutowski, K. E. Industrial Uses and Applications of Ionic Liquids. *Phys. Sci. Rev.* **2018**, *3* (5).

(147) Siriwardana, A. I. *Industrial Applications of Ionic Liquids - Electrochemistry in Ionic Liquids*; Springer International Publishing, 2015.

(148) Greer, A.; Jacquemin, J.; Hardacre, C. Industrial Applications of Ionic Liquids. *molecules* **2020**, *25*, 5207.

(149) Welton, T. Ionic Liquids in Catalysis. *Coord. Chem. Rev.* **2004**, *248* (21–24), 2459–2477.

(150) Plechkova N.V., S. K. R. *In Methods and Reagents for Green Chemistry: An Introduction: Ionic Liquids:"Designer" Solvents for Green Chemistry*; 2007.

(151) Earle, M. J.; Seddon, K. R. Ionic Liquids. Green Solvents for the Future. *Pure Appl. Chem.* **2000**, *72* (7), 1391–1398.

(152) Ruokonen, S. K.; Sanwald, C.; Sundvik, M.; Polnick, S.; Vyavaharkar, K.; Duša, F.; Holding, A. J.; King, A. W. T.; Kilpeläinen, I.; Lämmerhofer, M.; Panula, P.; Wiedmer, S. K. Effect of Ionic Liquids on Zebrafish (Danio Rerio) Viability, Behavior, and Histology; Correlation between Toxicity and Ionic Liquid Aggregation. *Environ. Sci. Technol.* **2016**, *50* (13), 7116–7125.

(153) Paduszynski, K.; Domanska, U. Viscosity of Ionic Liquids: An Extensive Database and a New Group Contribution Model Based on a Feed-Forward Artificial Neural Network. *J. Chem. Inf. Model.* **2014**, *54* (5), 1311–1324.

(154) Araque, J. C.; Yadav, S. K.; Shadeck, M.; Maroncelli, M.; Margulis, C. J. How Is Diffusion of Neutral and Charged Tracers Related to the Structure and Dynamics of a Room-Temperature Ionic Liquid? Large Deviations from Stokes–Einstein. *J. Phys. Chem. B* **2015**, *119* (23), 7015–7029.

(155) Bedrov, D.; Piquemal, J. P.; Borodin, O.; MacKerell, A. D.; Roux, B.; Schröder, C. Molecular Dynamics Simulations of Ionic Liquids and Electrolytes Using Polarizable Force Fields. *Chem. Rev.* **2019**, *119* (13), 7940.

(156) Gebbie, M. A.; Smith, A. M.; Dobbs, H. A.; Lee, A. A.; Warr, G. G.; Banquy, X.; Valtiner, M.; Rutland, M. W.; Israelachvili, J. N.; Perkin, S.; Atkin, R. Long Range Electrostatic Forces in Ionic Liquids. *Chem. Commun.* **2017**, *53* (7), 1214–1224.

(157) Zheng, W.; Liu, X.; Zhang, J.; Cheng, Y.; Wang, W. Molecular Dynamics Simulation of Ionic Liquid Electrospray: Microscopic Presentation of the Effects of Mixed Ionic Liquids. *Int. J. Heat Mass Transf.* **2022**, *182*, 121983.

(158) Ghatee, M. H.; Zolghadr, A. R. Local Depolarization in Hydrophobic and Hydrophilic Ionic Liquids/Water Mixtures: Car-Parrinello and Classical Molecular Dynamics Simulation. *J. Phys. Chem. C* **2013**, *117* (5), 2066–2077.

(159) Wei Jiang; Wang, Y.; Voth, G. A. Molecular Dynamics Simulation of Nanostructural Organization in Ionic Liquid/Water Mixtures. *J. Phys. Chem. B* **2007**, *111* (18), 4812–4818.

(160) Chang, T. M.; Billeck, S. E. Structure, Molecular Interactions, and Dynamics of Aqueous [BMIM][BF4] Mixtures: A Molecular Dynamics Study. *J. Phys. Chem. B* **2021**, *125* (4), 1227–1240.

(161) Krossing, I.; Slattery, J. M.; Daguenet, C.; Dyson, P. J.; Oleinikova, A.; Weingärtner, H. Why Are Ionic Liquids Liquid? A Simple Explanation Based on Lattice and Solvation Energies. *J. Am. Chem. Soc.* **2006**, *128* (41), 13427–13434.

(162) Philippi, F.; Rauber, D.; Springborg, M.; Hempelmann, R. Density Functional Theory Descriptors for Ionic Liquids and the Charge-Transfer Interpretation of the Haven Ratio. *J. Phys. Chem. A* **2019**, *123* (4), 851–861.

(163) Zhang, Y.; He, H.; Dong, K.; Fan, M.; Zhang, S. A DFT Study on Lignin Dissolution in Imidazolium-Based Ionic Liquids. *RSC Adv.* **2017**, *7* (21), 12670–12681.

(164) Kiratidis, A. L.; Miklavcic, S. J. Density Functional Theory of Confined Ionic Liquids: A Survey of the Effects of Ion Type, Molecular Charge Distribution, and Surface Adsorption. *J. Chem. Phys.* **2019**, *150* (18), 184502.

(165) Qin, M.; Zhong, F.; Sun, Y.; Tan, X.; Hu, K.; Zhang, H.; Kong, M.; Wang, G.; Zhuang, L. Experimental and DFT Studies on Surface Properties of Sulfonate-Based Surface Active Ionic Liquids. *J. Mol. Struct.* **2020**, *1215*, 128258.

(166) Shah, J. K. Ab Initio Molecular Dynamics Simulations of Ionic Liquids. *Annu. Rep. Comput. Chem.* **2018**, *14*, 95–122.

(167) Seeger, Z. L.; Izgorodina, E. I. A Systematic Study of DFT Performance for Geometry Optimizations of Ionic Liquid Clusters. *J. Chem. Theory Comput.* **2020**, *16* (10), 6735–6753.

(168) Thitakamol, B.; Veawab, A.; Aroonwilas, A. Environmental Impacts of Absorption-Based CO2 Capture Unit for Post-Combustion Treatment of Flue Gas from Coal-Fired Power Plant. *Int. J. Greenh. Gas Control* **2007**, *1*, 318–342.

(169) Shao, R.; Technical, S. A. *Amines Used in CO2 Capture - Health and Environmental Impacts*; Oslo, 2009. https://bellona.org/content/uploads/sites/3/fil_Bellona_report_September_2009_-_Amines_used_in_CO2_capture.pdf (accessed 2022-12-30).

(170) Blanchard, L.; Hancu, D.; Beckman, E.; JF, B. Green Processing Using Ionic Liquids and CO2. *Nature* **1999**, *399*, 28–29.

(171) Zhang, X.; Zhang, X.; Dong, H.; Zhao, Z.; S, Z. Carbon Capture with Ionic Liquids: Overview and Progress. *Energy Environ. Sci.* **2012**, *5*, 6668–6681.

(172) Shukla, S. K.; Khokarale, S. G.; Bui, T. Q.; Mikkola, J. P. T. Ionic Liquids: Potential Materials for Carbon Dioxide Capture and Utilization. *Front. Mater.* **2019**, *6*, 42.

(173) Blanchard, L. A.; Gu, Z.; Brennecke, J. F. High-Pressure Phase Behavior of Ionic Liquid/CO2 Systems. *J. Phys. Chem. B* **2001**, *105* (12), 2437–2444.

(174) Shariati, A.; Fluids, C. P. High-Pressure Phase Behavior of Systems with Ionic Liquids: Part III. The Binary System Carbon Dioxide+ 1-Hexyl-3-Methylimidazolium Hexafluorophosphate. *J. Supercrit. Fluids* **2004**, *30* (2), 139–144.

(175) Kim, J.; Lim, J.; JW Kang. Measurement and Correlation of Solubility of Carbon Dioxide in 1-Alkyl-3-Methylimidazolium Hexafluorophosphate Ionic Liquids. *Fluid Phase Equilib.* **2011**, *306* (2), 251–255.

(176) Kim, Y. S.; Choi, W. Y.; Jang, J. H.; Yoo, K. P.; Lee, C. S. Solubility Measurement and Prediction of Carbon Dioxide in Ionic Liquids. *Fluid Phase Equilib.* **2005**, *228–229*, 439–445.

(177) Aki, S. N. V. K.; Mellein, B. R.; Saurer, E. M.; Brennecke, J. F. High-Pressure Phase Behavior of Carbon Dioxide with Imidazolium-Based Ionic Liquids. *J. Phys. Chem. B* **2004**, *108* (52), 20355–20365.

(178) Cadena, C.; Anthony, J. L.; Shah, J. K.; Morrow, T. I.; Brennecke, J. F.; Maginn, E. J. Why Is CO2 so Soluble in Imidazolium-Based Ionic Liquids? *J. Am. Chem. Soc.* **2004**, *126* (16), 5300–5308.

(179) Anthony, J. L.; Anderson, J. L.; Maginn, E. J.; Brennecke, J. F. Anion Effects on Gas Solubility in Ionic Liquids. *J. Phys. Chem. B* **2005**, *109* (13), 6366–6374.

(180) Bates, E. D.; Mayton, R. D.; Ntai, I.; Davis, J. H. CO2 Capture by a Task-Specific Ionic Liquid. *J. Am. Chem. Soc.* **2002**, *124* (6), 926–927.

(181) Cao, B.; Du, J.; Liu, S.; Zhu, X.; Sun, X.; Sun, H.; Fu, H. Carbon Dioxide Capture by Amino-Functionalized Ionic Liquids: DFT Based Theoretical Analysis Substantiated by FT-IR Investigation. *RSC Adv.* **2016**, *6* (13), 10462–10470.

(182) Gurkan, B. E.; De La Fuente, J. C.; Mindrup, E. M.; Ficke, L. E.; Goodrich, B. F.; Price, E. A.;

Schneider, W. F.; Brennecke, J. F. Equimolar CO2 Absorption by Anion-Functionalized Ionic Liquids. *J. Am. Chem. Soc.* **2010**, *132* (7), 2116–2117.

(183) Kasahara, S.; Kamio, E.; Shaikh, A.; Matsuki, T. Effect of the Amino-Group Densities of Functionalized Ionic Liquids on the Facilitated Transport Properties for CO2 Separation. *J. Memb. Sci.* **2016**, *503*, 148–157.

(184) Luo, X.; Guo, Y.; Ding, F.; Zhao, H.; Cui, G.; Li, H.; Wang, C.; Luo, X.; Guo, Y.; Ding, F.; Zhao, H.; Cui, G.; Li, H.; Wang, C. Significant Improvements in CO2 Capture by Pyridine-Containing Anion-Functionalized Ionic Liquids through Multiple-Site Cooperative Interactions. *Angew. Chemie* **2014**, *126* (27), 7173–7177.

(185) Zhang, S.; Sun, N.; He, X.; Lu, X.; Zhang, X. Physical Properties of Ionic Liquids: Database and Evaluation. *J. Phys. Chem. Ref. Data* **2006**, *35* (4), 1475–1517.

(186) Sun, N.; He, X.; Dong, K.; Zhang, X.; Lu, X.; He, H.; Zhang, S. Prediction of the Melting Points for Two Kinds of Room Temperature Ionic Liquids. *Fluid Phase Equilib.* **2006**, *246* (1–2), 137–142.

(187) Varnek, A.; Kireeva, N.; Tetko, I. V.; Baskin, I. I.; Solov'ev, V. P. Exhaustive QSPR Studies of a Large Diverse Set of Ionic Liquids: How Accurately Can We Predict Melting Points? *J. Chem. Inf. Model.* **2007**, *47* (3), 1111–1122.

(188) Lazzús, J. A. A Group Contribution Method to Predict the Melting Point of Ionic Liquids. *Fluid Phase Equilib.* **2012**, *313*, 1–6.

(189) Wang, J.; Li, Z.; Li, C.; Wang, Z. Density Prediction of Ionic Liquids at Different Temperatures and Pressures Using a Group Contribution Equation of State Based on Electrolyte Perturbation Theory. *Ind. Eng. Chem. Res.* **2010**, *49* (9), 4420–4425.

(190) Paduszyński, K.; Domańska, U. A New Group Contribution Method for Prediction of Density of Pure Ionic Liquids over a Wide Range of Temperature and Pressure. *Ind. Eng. Chem. Res.* **2012**, *51*, 591–604.

(191) Paduszyński, K. Extensive Databases and Group Contribution QSPRs of Ionic Liquids Properties. 1. Density. *Ind. Eng. Chem. Res.* **2019**, *58* (13), 5322–5338.

(192) Rostami, A.; Baghban, A.; Shirazian, S. On the Evaluation of Density of Ionic Liquids: Towards a Comparative Study. *Chem. Eng. Res. Des.* **2019**, *147*, 648–663.

(193) Cho, C. W.; Pham, T. P. T.; Zhao, Y.; Stolte, S.; Yun, Y. S. Review of the Toxic Effects of Ionic Liquids. *Sci. Total Environ.* **2021**, *786*, 147309.

(194) Sivapragasam, M.; Moniruzzaman, M.; Goto, M. An Overview on the Toxicological Properties of Ionic Liquids toward Microorganisms. *Biotechnol. J.* **2020**, *15* (4), 1900073.

(195) Abramenko, N.; Kustov, L.; Metelytsia, L.; Kovalishyn, V.; Tetko, I.; Peijnenburg, W. A Review of Recent Advances towards the Development of QSAR Models for Toxicity Assessment of Ionic Liquids. *J. Hazard. Mater.* **2020**, *384*, 121429.

(196) Salam, M. A.; Abdullah, B.; Ramli, A.; Mujtaba, I. M. Structural Feature Based Computational Approach of Toxicity Prediction of Ionic Liquids: Cationic and Anionic Effects on Ionic Liquids Toxicity. *J. Mol. Liq.* **2016**, *224*, 393–400.

(197) Cao, L.; Zhu, P.; Zhao, Y.; Zhao, J. Using Machine Learning and Quantum Chemistry Descriptors to Predict the Toxicity of Ionic Liquids. *J. Hazard. Mater.* **2018**, *352*, 17–26.

(198) *RDKit*. RDKit: Open-source cheminformatics; http://www.rdkit.org. https://www.rdkit.org/ (accessed 2021-04-01).

(199) Chipofya, M.; Tayara, H.; Chong, K. T. Deep Probabilistic Learning Model for Prediction of Ionic Liquids Toxicity. *Int. J. Mol. Sci.* **2022**, *23* (9), 5258.

(200) Yan, J.; Liu, G.; Chen, H.; Hu, S.; Wang, X.; Yan, B.; Yan, X. ILTox: A Curated Toxicity

Database for Machine Learning and Design of Environmentally Friendly Ionic Liquids. *Environ. Sci. Technol. Lett.* **2023**.

(201) Sharma, A. K.; Srivastava, G. N.; Roy, A.; Sharma, V. K. ToxiM: A Toxicity Prediction Tool for Small Molecules Developed Using Machine Learning and Chemoinformatics Approaches. *Front. Pharmacol.* **2017**, *8*, 880.

(202) Martin; Todd. User's Guide for T. E. S. T. (Toxicity Estimation Software Tool) Version 5.1 A Java Application to Estimate Toxicities and Physical Properties from Molecular Structure. 2020. https://www.epa.gov/chemical-research/toxicity-estimation-software-tool-test (accessed 2023-04-18).

(203) Weingärtner, H. Understanding Ionic Liquids at the Molecular Level: Facts, Problems, and Controversies. *Angew. Chemie Int.* **2008**, *47* (4), 654–670.

(204) Mahinder Ramdin, Theo W. de Loos, T. J. H. V. State-of-the-Art of CO2 Capture with Ionic Liquids. *Ind. Eng. Chem. Res.* **2012**, *51* (24), 8149–8177.

(205) Gurkan, B.; Goodrich, B. F.; Mindrup, E. M.; Ficke, L. E.; Massel, M.; Seo, S.; Senftle, T. P.; Wu, H.; Glaser, M. F.; Shah, J. K.; Maginn, E. J.; Brennecke, J. F.; Schneider, W. F. Molecular Design of High Capacity, Low Viscosity, Chemically Tunable Ionic Liquids for CO2 Capture. *J. Phys. Chem. Lett.* **2010**, *1* (24), 3494–3499.

(206) Liu, X.; Zhou, G.; Zhang, S.; Yao, X. Molecular Dynamics Simulation of Dual Amino-Functionalized Imidazolium-Based Ionic Liquids. *Fluid Phase Equilib.* **2009**, *284*, 44–49.

(207) Jacquemin, J.; Husson, P.; Padua, A. A. H.; Majer, V. Density and Viscosity of Several Pure and Water-Saturated Ionic Liquids. *Green Chem.* **2006**, *8* (2), 172–180.

(208) Jacquemin, J.; Husson, P.; Padua, A.; Majer, V. Density and Viscosity of Several Pure and Water-Saturated Ionic Liquids. *Green Chem.* **2006**, *8* (2), 172–180.

(209) Abbott, A. P. Application of Hole Theory to the Viscosity of Ionic and Molecular Liquids. *ChemPhysChem* **2004**, *5* (8), 1242–1246.

(210) Slattery, J. M.; Daguenet, C.; Dyson, P. J.; Schubert, T. J.; Krossing, I. How to Predict the Physical Properties of Ionic Liquids: A Volume-based Approach. *Angew. Chemie* **2007**, *119* (28), 5480–5484.

(211) Bandrés, I.; Alcalde, R.; Lafuente, C.; Atilhan, M.; Aparicio, S. On the Viscosity of Pyridinium Based Ionic Liquids: An Experimental and Computational Study. *J. Phys. Chem. B* **2011**, *115* (43), 12499–12513.

(212) Dutt, N.; Ravikumar, Y.; Engineering, K. R. Representation of Ionic Liquid Viscosity-Temperature Data by Generalized Correlations and an Artificial Neural Network (Ann) Model. *Chem. Eng. Commun.* **2013**, *200* (12), 1600–1622.

(213) Tochigi, K.; Yamamoto, H. Estimation of Ionic Conductivity and Viscosity of Ionic Liquids Using a QSPR Model. *J. Phys. Chem. C* **2007**, *111* (43), 15989–15994.

(214) Bini, R.; Malvaldi, M.; Pitner, W. R. QSPR Correlation for Conductivities and Viscosities of Low-temperature Melting Ionic Liquids. *J. Phys. Org. Chem.* **2008**, *21* (7–8), 622–629.

(215) Han, C.; Yu, G.; Wen, L.; Zhao, D.; Asumana, C.; Chen, X. Data and QSPR Study for Viscosity of Imidazolium-Based Ionic Liquids. *Fluid Phase Equilib.* **2011**, *300* (1–2), 95–104.

(216) Valderrama, J. O.; Muñoz, J. M.; Rojas, R. E. Viscosity of Ionic Liquids Using the Concept of Mass Connectivity and Artificial Neural Networks. *Korean J. Chem. Eng.* **2011**, *28* (6), 1451–1457.

(217) Yu, G.; Zhao, D.; Wen, L.; Yang, S.; X Chen, X. Viscosity of Ionic Liquids: Database, Observation, and Quantitative Structure-property Relationship Analysis. *AIChE J.* **2011**, *58* (9), 2885–2899.

(218) Chen, B.; Liang, M.; Wu, T.; Wang, H. A High Correlate and Simplified QSPR for Viscosity of Imidazolium-Based Ionic Liquids. *Fluid Phase Equilib.* **2013**, *350*, 37–42.

(219) Matsuda, H.; Yamamoto, H.; Kurihara, K.; Tochigi, K. Prediction of the Ionic Conductivity and Viscosity of Ionic Liquids by QSPR Using Descriptors of Group Contribution Type. *J. Comput. Aided Chem.* **2007**, *8*, 114–127.

(220) Gardas, R.; Coutinho, J. A Group Contribution Method for Viscosity Estimation of Ionic Liquids. *Fluid Phase Equilib.* **2008**, *266* (1–2), 195–201.

(221) Gardas, R.; Coutinho, J. Group Contribution Methods for the Prediction of Thermophysical and Transport Properties of Ionic Liquids. *AIChE J.* **2009**, *55* (5), 1274–1290.

(222) Gharagheizi, F.; Ilani-Kashkouli, P. Development of a Group Contribution Method for Determination of Viscosity of Ionic Liquids at Atmospheric Pressure. *Chem. Eng. Sci.* **2012**, *80*, 326–333.

(223) Chen, B. K.; Liang, M. J.; Wu, T. Y.; Wang, H. P. A High Correlate and Simplified QSPR for Viscosity of Imidazolium-Based Ionic Liquids. *Fluid Phase Equilib.* **2013**, *350*, 37–42.

(224) Yu, G.; Wen, L.; Zhao, D.; Asumana, C.; Chen, X. QSPR Study on the Viscosity of Bis(Trifluoromethylsulfonyl)Imide-Based Ionic Liquids. *J. Mol. Liq.* **2013**, *184*, 51–59.

(225) Mirkhani, S.; Gharagheizi, F. Predictive Quantitative Structure–Property Relationship Model for the Estimation of Ionic Liquid Viscosity. *Ind. Eng. Chem. Res.* **2012**, *51* (5), 2470–2477.

(226) Zhao, Y.; Huang, Y.; Zhang, X.; Zhang, S. A Quantitative Prediction of the Viscosity of Ionic Liquids Using Sσ-Profile Molecular Descriptors. *Phys. Chem. Chem. Phys.* **2015**, *17* (5), 3761–3767.

(227) Gharagheizi, F.; Ilani-Kashkouli, P.; Mohammadi, A. H.; Ramjugernath, D.; Richon, D. Development of a Group Contribution Method for Determination of Viscosity of Ionic Liquids at Atmospheric Pressure. *Chem. Eng. Sci.* **2012**, *80*, 326–333.

(228) Lazzús, J. A.; Pulgar-Villarroel, G. A Group Contribution Method to Estimate the Viscosity of Ionic Liquids at Different Temperatures. *J. Mol. Liq.* **2015**, *209*, 161–168.

(229) Yan, F.; He, W.; Jia, Q.; Wang, Q.; Xia, S.; Ma, P. Prediction of Ionic Liquids Viscosity at Variable Temperatures and Pressures. *Chem. Eng. Sci.* **2018**, *184*, 134–140.

(230) Paduszyński, K. Extensive Databases and Group Contribution QSPRs of Ionic Liquids Properties. 2. Viscosity. *Ind. Eng. Chem. Res.* **2019**, *58* (36), 41.

(231) William, H. III. Experiments on the Quantity of Gases Absorbed by Water, at Different Temperatures, and under Different Pressures. *Philos. Trans. R. Soc. London* **1803**, *93*, 29–274.

(232) Lei, Z.; Dai, C.; Chen, B. Gas Solubility in Ionic Liquids. *Chem. Rev.* **2014**, *114* (2), 1289–1326.

(233) Muldoon, M. J.; Aki, S. N. V. K.; Anderson, J. L.; Dixon, J. K.; Brennecke, J. F. Improving Carbon Dioxide Solubility in Ionic Liquids. *J. Phys. Chem. B* **2007**, *111* (30), 9001–9009.

(234) Palomar, J.; Gonzalez-Miquel, M.; Polo, A. Understanding the Physical Absorption of CO2 in Ionic Liquids Using the COSMO-RS Method. *Ind. Eng. Chem. Res.* **2011**, *50* (6), 3452–3463.

(235) Almantariotis, D.; Gefflaut, T.; Pádua, A. A. H.; Coxam, J. Y.; Costa Gomes, M. F. Effect of Fluorination and Size of the Alkyl Side-Chain on the Solubility of Carbon Dioxide in 1-Alkyl-3-Methylimidazolium Bis(Trifluoromethylsulfonyl) Amide Ionic Liquids. *J. Phys. Chem. B* **2010**, *114* (10), 3608–3617.

(236) Fredenslund, A.; Jones, R. L.; Prausnitz, J. M. Group-contribution Estimation of Activity Coefficients in Nonideal Liquid Mixtures. *AIChE J.* **1975**, *21* (6), 1086–1099.

(237) Sedghamiz, M. A.; Rasoolzadeh, A.; Rahimpour, M. R. The Ability of Artificial Neural Network in Prediction of the Acid Gases Solubility in Different Ionic Liquids. *J. CO2 Util.*

2015, *9*, 39–47.

(238) Eslamimanesh, A.; Gharagheizi, F.; Mohammadi, A. H.; Richon, D. Artificial Neural Network Modeling of Solubility of Supercritical Carbon Dioxide in 24 Commonly Used Ionic Liquids. *Chem. Eng. Sci.* **2011**, *66* (13), 3039–3044.

(239) Tatar, A.; Naseri, S.; Bahadori, M.; Hezave, A. Z.; Kashiwao, T.; Bahadori, A.; Darvish, H. Prediction of Carbon Dioxide Solubility in Ionic Liquids Using MLP and Radial Basis Function (RBF) Neural Networks. *J. Taiwan Inst. Chem. Eng.* **2016**, *60*, 151–164.

(240) Deng, T.; Liu, F. H.; Jia, G. Z. Prediction Carbon Dioxide Solubility in Ionic Liquids Based on Deep Learning. *Mol. Phys.* **2020**, *118* (6), e1652367.

(241) Song, Z.; Shi, H.; Zhang, X.; Zhou, T. Prediction of CO2 Solubility in Ionic Liquids Using Machine Learning Methods. *Chem. Eng. Sci.* **2020**, *223*, 115752.

(242) Jian, Y.; Wang, Y.; Barati Farimani, A. Predicting CO2Absorption in Ionic Liquids with Molecular Descriptors and Explainable Graph Neural Networks. *ACS Sustain. Chem. Eng.* **2022**, *10* (50), 16681–16691.

(243) Schrodinger, E. Quantisierung Als Eigenwertproblem. *Ann. Phys.* **1926**, *385* (13), 437–490.

(244) Born, M. Born-Oppenheimer Approximation. *Ann. Phys.* **1927**, *84*, 457.

(245) Cramer, C. *Essentials of Computational Chemistry: Theories and Models*; John Wiley & Sons, 2013.

(246) Harvey, J. *Computational Chemistry*; 2018.

(247) David Sholl and Janice A Steckel. Density Functional Theory: A Practical Introduction. *John Wiley Sons* **2011**.

(248) Parr, R. G.; Yang, W. Density Functional Approach to the Frontier-Electron Theory of Chemical Reactivity. *J. Am. Chem. Soc.* **1984**, *106* (14), 4049–4050.

(249) Hohenberg, P.; Kohn, W. Inhomogeneous Electron Gas. *Phys. Rev.* **1964**, *136* (3B), B864–B871.

(250) Kohn, W.; Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **1965**, *140* (4A), A1133–A1138.

(251) Kanungo, B.; Zimmerman, P. M.; Gavini, V. Exact Exchange-Correlation Potentials from Ground-State Electron Densities. *Nat. Commun.* **2019**, *10*, 4497.

(252) Mardirossian, N.; Head-Gordon, M. Thirty Years of Density Functional Theory in Computational Chemistry: An Overview and Extensive Assessment of 200 Density Functionals. *Mol. Phys.* **2017**, *115* (19), 2315–2372.

(253) Goerigk, L.; Hansen, A.; Bauer, C.; S Ehrlich, S. A Look at the Density Functional Theory Zoo with the Advanced GMTKN55 Database for General Main Group Thermochemistry, Kinetics and Noncovalent. *Phys. Chem. Chem. Phys.* **2017**, *19*, 32184–32215.

(254) Perdew, J. P.; Schmidt, K. Jacob's Ladder of Density Functional Approximations for the Exchange-Correlation Energy. *AIP Conf. Proc.* **2001**, *577*, 1.

(255) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77* (18), 3865.

(256) Becke, A. D. Density-Functional Exchange-Energy Approximation with Correct Asymptotic Behavior. *Phys. Rev. A* **1988**, *38* (6), 3098.

(257) Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti Correlation-Energy Formula into a Functional of the Electron Density. *Phys. Rev. B* **1988**, *37* (2), 785–789.

(258) Tao, J.; Perdew, J. P.; Staroverov, V. N.; Scuseria, G. E. Climbing the Density Functional Ladder: Nonempirical Meta–Generalized Gradient Approximation Designed for Molecules

and Solids. *Phys. Rev. Lett.* **2003**, *91* (14).

(259) Adamo, C.; Barone, V. Toward Reliable Density Functional Methods without Adjustable Parameters: The PBE0 Model. *J. Chem. Phys.* **1999**, *110* (13), 6158–6170.

(260) Stephens, P. J.; Devlin, F. J.; Chabalowski, C. F.; Frisch, M. J. Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields. *J. Phys. Chem.* **1994**, *98* (45), 11623–11627.

(261) Becke, A. D. Density-functional Thermochemistry. III. The Role of Exact Exchange. *J. Chem. Phys.* **1998**, *98* (7), 5648.

(262) Zhao, Y.; Truhlar, D. G. The M06 Suite of Density Functionals for Main Group Thermochemistry, Thermochemical Kinetics, Noncovalent Interactions, Excited States, and Transition Elements: Two New Functionals and Systematic Testing of Four M06-Class Functionals and 12 Other Function. *Theor. Chem. Acc.* **2008**, *120*, 215–241.

(263) Peng, Q.; Duarte, F.; Reviews, R. P.-C. S.; 2016, U. Computing Organic Stereoselectivity–from Concepts to Quantitative Calculations and Predictions. *Chem. Soc. Rev.* **2016**, *45*, 6093–6107.

(264) Becke, A. D. Perspective: Fifty Years of Density-Functional Theory in Chemical Physics. *J. Chem. Phys.* **2014**, *140* (18), 18A301.

(265) Cohen, A. J.; Mori-Sánchez, P.; Yang, W. Challenges for Density Functional Theory. *Chem. Rev.* **2012**, *112* (1), 289–320.

(266) Puzder, A.; Dion, M.; Langreth, D. C. Binding Energies in Benzene Dimers: Nonlocal Density Functional Calculations. *J. Chem. Phys.* **2006**, *124*, 164105.

(267) Grimme, S. Density Functional Theory with London Dispersion Corrections. *WIREs Comput Mol Sci* **2011**, *1* (2), 211–228.

(268) Grimme, S.; Antony, J.; Ehrlich, S. A Consistent and Accurate Ab Initio Parametrization of Density Functional Dispersion Correction (DFT-D) for the 94 Elements H-Pu. *J. Chem. Phys.* **2010**, *132* (15), 224101.

(269) Becke, A. D.; Johnson, E. R. A Density-Functional Model of the Dispersion Interaction. *J. Chem. Phys.* **2005**, *123*, 154101.

(270) Grimme, S.; Ehrlich, S.; Goerigk, L. Effect of the Damping Function in Dispersion Corrected Density Functional Theory. *J. Comput. Chem.* **2011**, *32* (7), 1456–1465.

(271) Goerigk, L.; Grimme, S. A Thorough Benchmark of Density Functional Methods for General Main Group Thermochemistry, Kinetics, and Noncovalent Interactions. *Phys. Chem. Chem. Phys.* **2011**, *13*, 6670–6688.

(272) Wheeler, S. E.; Houk, K. N.; Schleyer, P. V. R.; Allen, W. D. A Hierarchy of Homodesmotic Reactions for Thermochemistry. *J. Am. Chem. Soc.* **2009**, *131* (7), 2547–2560.

(273) Nagy, P. R.; Kállay, M. Approaching the Basis Set Limit of CCSD(T) Energies for Large Molecules with Local Natural Orbital Coupled-Cluster Methods. *J. Chem. Theory Comput.* **2019**, *15* (10), 5275–5298.

(274) Gordon, M. S.; Barca, G.; Leang, S. S.; Poole, D.; Rendell, A. P.; Galvez Vallejo, J. L.; Westheimer, B. Novel Computer Architectures and Quantum Chemistry. *J. Phys. Chem. A* **2020**, *124* (23), 4557–4582.

(275) Dral, P. O. Quantum Chemistry in the Age of Machine Learning. *J. Phys. Chem. Lett.* **2020**, *11* (6), 2336–2347.

(276) Marenich, A. V.; Cramer, C. J.; Truhlar, D. G. Universal Solvation Model Based on Solute Electron Density and on a Continuum Model of the Solvent Defined by the Bulk Dielectric Constant and Atomic Surface Tensions. *J. Phys. Chem. B* **2009**, *113* (18), 6378–6396.

(277) Martin, J.; El-Yazal, J.; François, J. Basis Set Convergence and Performance of Density Functional Theory Including Exact Exchange Contributions for Geometries and Harmonic Frequencies. *Mol. Phys.* **1995**, *86* (6), 1450.

(278) Sure, R.; Brandenburg, J.; Grimme, S. Small Atomic Orbital Basis Set First-principles Quantum Chemical Methods for Large Molecular and Periodic Systems: A Critical Analysis of Error Sources. *Chem. Open Rev.* **2016**, *5*, 94–109.

(279) Kirschner, K. N.; Reith, D.; Heiden, W. The Performance of Dunning, Jensen, and Karlsruhe Basis Sets on Computing Relative Energies and Geometries. *Soft Mater.* **2020**, *18*, 200–214.

(280) Dyczmons, V. No N4-Dependence in the Calculation of Large Molecules. *Theor. Chim. Acta* **1973**, *28* (3), 307–310.

(281) Häser, M.; Ahlrichs, R. Improvements on the Direct SCF Method. *J. Comput. Chem.* **1989**, *10* (1), 104–111.

(282) Neese, F.; Wennmohs, F.; Hansen, A.; Becker, U. Efficient, Approximate and Parallel Hartree–Fock and Hybrid DFT Calculations. A "chain-of-Spheres" Algorithm for the Hartree–Fock Exchange. *Chem. Phys.* **2009**, *356*, 98–109.

(283) Neese, F. Software Update: The ORCA Program System, Version 4.0. *WIREs Comput Mol Sci* **2018**, *8*, e1327.

(284) Gaus, M.; Cui, Q.; Elstner, M. DFTB3: Extension of the Self-Consistent-Charge Density-Functional Tight-Binding Method (SCC-DFTB). *J. Chem. Theory Comput.* **2011**, *7* (4), 931–948.

(285) Grimme, S.; Bannwarth, C.; Shushkov, P. A Robust and Accurate Tight-Binding Quantum Chemical Method for Structures, Vibrational Frequencies, and Noncovalent Interactions of Large Molecular Systems. *J. Chem. Theory Comput.* **2017**, *13*, 1989–2009.

(286) Bannwarth, C.; Eike Caldeweyher, |; Ehlert, S.; Hansen, A.; Pracht, P.; Seibert, J.; Spicher, S.; Grimme, S. Extended Tight-binding Quantum Chemistry Methods. *WIREs Comput Mol Sci* **2020**, *11*, e1493.

(287) Christensen, A.; Kubar, T.; Cui, Q.; Elstner, M. Semiempirical Quantum Mechanical Methods for Noncovalent Interactions for Chemical and Biochemical Applications. *Chem. Rev.* **2016**, *116* (9), 5301–5337.

(288) Addicoat, M. A.; Stefanovic, R.; Webber, G. B.; Atkin, R.; Page, A. J. Assessment of the Density Functional Tight Binding Method for Protic Ionic Liquids. *J. Chem. Theory Comput.* **2014**, *10*, 4633–4643.

(289) Ochterski, J. W. Thermochemistry in Gaussian. **2000**.

(290) McQuarrie, D. *Statistical Mechanics*; University Science Books: Sausalito: California, USA, 2000.

(291) Jovic, A.; Brkić, K.; Jović, A.; Brkić, K.; Bogunović, N. A Review of Feature Selection Methods with Applications. In *38th International Convention on Information and Communication Technology, Electronics and Microelectronics*; 2015; pp 1200–1205.

(292) El Aboudi, N.; Benhlima, L. Review on Wrapper Feature Selection Approaches. In *2016 International Conference on Engineering & MIS*; Institute of Electrical and Electronics Engineers Inc., 2016; pp 1–5.

(293) Grisoni, F.; Ballabio, D.; Todeschini, R.; Consonni, V. Molecular Descriptors for Structure–Activity Applications: A Hands-on Approach. *Methods Mol. Biol.* **2018**, *1800*, 3–53.

(294) Raghunathan, S.; Priyakumar, U. D. Molecular Representations for Machine Learning Applications in Chemistry. *Int. J. Quantum Chem.* **2022**, *122* (7), 26870.

(295) Guha, R.; Willighagen, E. A Survey of Quantitative Descriptions of Molecular Structure. *Curr.*

Top. Med. Chem. **2012**, *12*, 1946–1956.

(296) Bartók, A. P.; Kondor, R.; Csányi, G. On Representing Chemical Environments. *Phys. Rev. B* **2013**, *87*, 184115.

(297) Devinyak, O.; Havrylyuk, D.; Lesyk, R. 3D-MoRSE Descriptors Explained. *J. Mol. Graph. Model.* **2014**, *54*, 194–203.

(298) Kubinyi, H. *Comparative Molecular Field Analysis (CoMFA)*; John Wiley & Sons, Ltd, 2002.

(299) Hong, H.; Xie, Q.; Ge, W.; Qian, F.; Fang, H.; Shi, L.; Su, Z.; Perkins, R.; Tong, W. Mold2, Molecular Descriptors from 2D Structures for Chemoinformatics and Toxicoinformatics. *J. Chem. Inf. Model.* **2008**, *48* (7), 1337–1344.

(300) Raschka, S. MLxtend: Providing Machine Learning and Data Science Utilities and Extensions to Python's Scientific Computing Stack. *JOSS* **2018**, *3* (24), 638.

(301) Yap, Chun, W. PaDEL-Descriptor: An Open Source Software to Calculate Molecular Descriptors and Fingerprints. *J. Comput. Chem.* **2011**, *32* (7), 1466–1474.

(302) Gálvez, J.; Garcia, R.; Salabert, T.; Soler, R. Charge Indexes. New Topological Descriptors. *J. Chem. Inf. Comput. Sci* **1994**, *34*, 520–525.

(303) Stanton, D. T.; Dimitrov, S.; Grancharov, V.; Mekenyan, O. G. Charged Partial Surface Area (CPSA) Descriptors QSAR Applications. *SAR QSAR Environ. Res.* **2002**, *13* (2), 341–351.

(304) Morgan, H. The Generation of a Unique Machine Description for Chemical Structures—a Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107–113.

(305) Capecchi, A.; Probst, D.; Reymond, J. One Molecular Fingerprint to Rule Them All: Drugs, Biomolecules, and the Metabolome. *J. Cheminform.* **2020**, *12*, 43.

(306) Riniker, S.; Landrum, G. A. Similarity Maps - a Visualization Strategy for Molecular Fingerprints and Machine-Learning Methods. *J. Cheminform.* **2013**, *5* (1), 43.

(307) Wang, J.; Li, H.; Zhao, W.; Pang, T.; Sun, Z.; Zhang, B.; Xu, H. MIFNN: Molecular Information Feature Extraction and Fusion Deep Neural Network for Screening Potential Drugs. *Curr. Issues Mol. Biol.* **2022**, *44* (11), 5638–5654.

(308) Durant, J.; Leland, B.; Henry, D.; Nourse, J. Reoptimization of MDL Keys for Use in Drug Discovery. *J. Chem. Inf. Comput. Sci.* **2002**, *42* (6), 1273–1280.

(309) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50* (5), 742–754.

(310) Smith, D. H.; Carhart, R. E.; Venkataraghavan, R. Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Applications. *J. Chem. Inf. Comput. Sci.* **1985**, *25* (2), 64–73.

(311) Hert, J.; Willett, P.; Wilton, D.; Acklin, P. Comparison of Topological Descriptors for Similarity-Based Virtual Screening Using Multiple Bioactive Reference Structures. *Org. Biomol. Chem.* **2004**, *2*, 3256–3266.

(312) Butina, D. Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way to Cluster Small and Large Data Sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39* (4), 747–750.

(313) Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and T. T. *Descriptor list*. https://mordred-descriptor.github.io/documentation/master/descriptors.html.

(314) Gao, P.; Liu, Z.; Tan, Y.; Zhang, J.; Xu, L.; Wang, Y.; Jeong, S. Y. Accurate Predictions of Drugs Aqueous Solubility via Deep Learning Tools. *J. Mol. Struct.* **2022**, *1249*, 131562.

(315) Yang, L.; Jin, C.; Yang, G.; Bing, Z.; Huang, L.; Niu, Y.; Yang, L. Transformer-Based Deep Learning Method for Optimizing ADMET Properties of Lead Compounds. *Phys. Chem. Chem.*

Phys. **2023**, *25* (3), 2377–2385.

(316) Wang, Z.; Chen, J.; Hong, H. Developing QSAR Models with Defined Applicability Domains on PPARγBinding Affinity Using Large Data Sets and Machine Learning Algorithms. *Environ. Sci. Technol.* **2021**, *55* (10), 6857–6866.

(317) Kobayashi, Y.; Yoshida, K. Development of QSAR Models for Prediction of Fish Bioconcentration Factors Using Physicochemical Properties and Molecular Descriptors with Machine Learning Algorithms. *Ecol. Inform.* **2021**, *63*, 101285.

(318) Mcdonagh, J. L.; Zavitsanou, S.; Harrison, A.; Zubarev, D. Y.; Wunsch, B. H.; Van Kessel, T.; Cipcigan, F. Chemical Space Analysis and Property Prediction for Carbon Capture Amine Molecules. *ChemRxiv (preprint)* **2023**. https://doi.org/10.26434/chemrxiv-2022-6gx8k-v2.

(319) Cusachs, L. C.; Politzer, P. On the Problem of Defining the Charge on an Atom in a Molecule. *Chem. Phys. Lett.* **1968**, *1* (11), 529–531.

(320) Saha, S.; Roy, R. K.; Ayers, P. W. Are the Hirshfeld and Mulliken Population Analysis Schemes Consistent with Chemical Intuition? *Int. J. Quantum Chem.* **2009**, *109* (9), 1790–1806.

(321) Jensen, F. *Introduction to Computational Chemistry*; John Wiley & Sons, 2017.

(322) Itatani, J.; Levesque, J.; Zeidler, D.; Niikura, H.; Pépin, H.; Kieffer, J.; Corkum, P.; Villeneuve, D. Tomographic Imaging of Molecular Orbitals. *Nature* **2004**, *432*, 867–871.

(323) Mineva, T.; Parvanov, V.; Petrov, I.; Neshev, N.; Russo, N. Fukui Indices from Perturbed Kohn−Sham Orbitals and Regional Softness from Mayer Atomic Valences. *J. Phys. Chem. A* **2001**, *105* (10), 1959–1967.

(324) DeVleeschouwer, F.; VanSpeybroeck, V.; Waroquier, M.; Geerlings, P.; DeProft, F. Electrophilicity and Nucleophilicity Index for Radicals. *Org. Lett.* **2007**, *9* (14), 2721–2724.

(325) Melin, J.; Aparicio, F.; Subramanian, V.; Galván, M.; Chattaraj, P. K. Is the Fukui Function a Right Descriptor of Hard-Hard Interactions? *J. Phys. Chem. A* **2004**, *108* (13), 2487–2491.

(326) Koopmans, T. Über Die Zuordnung von Wellenfunktionen Und Eigenwerten Zu Den Einzelnen Elektronen Eines Atoms. *Physica* **1934**, *1*, 104–113.

(327) Bredas, J. L. Mind the Gap! *Mater. Horizons* **2013**, *1*, 17–19.

(328) Lewis, G. N. The Atom and the Molecule. *J. Am. Chem. Soc.* **1916**, *38* (4), 762–785.

(329) Scerri, E. R. Have Orbitals Really Been Observed? *J. Chem. Educ.* **2000**, *77* (11), 1492–1494.

(330) Pham, B. Q.; Gordon, M. S. Can Orbitals Really Be Observed in Scanning Tunneling Microscopy Experiments? *J. Phys. Chem. A* **2017**, *121* (26), 4851–4852.

(331) Winstein, S.; Holness, N, J. Neighboring Carbon and Hydrogen. XIX. t-Butylcyclohexyl Derivatives. Quantitative Conformational Analysis. *J. Am. Chem. Soc.* **1955**, *77* (21), 5562–5578.

(332) Bott, G.; Field, L, D.; Sternhell, S. Steric Effects. A Study of a Rationally Designed System. *J. Am. Chem. Soc.* **1980**, *102* (17), 5618–5626.

(333) Adams, R.; Yuan, H. C. The Stereochemistry of Diphenyls and Analogous Compounds. *Chem. Rev.* **1933**, *12* (2), 261–338.

(334) Niksch, T.; Görls, H.; Weigand, W. The Extension of the Solid-Angle Concept to Bidentate Ligands. *Eur. J. Inorg. Chem.* **2010**, *2010* (1), 95–105.

(335) Tolman, C. A. Steric Effects of Phosphorus Ligands in Organometallic Chemistry and Homogeneous Catalysis. *Chem. Rev.* **1977**, *77* (3), 313–348.

(336) Taft, R. W. Polar and Steric Substituent Constants for Aliphatic and o-Benzoate Groups from Rates of Esterification and Hydrolysis of Esters. *J. Am. Chem. Soc.* **1952**, *74* (12), 3120–3128.

(337) Taft, R. W. Linear Steric Energy Relationships. *J. Am. Chem. Soc.* **1953**, *75* (18), 4538–4539.

(338) Charton, M. Steric Effects. II. Base-Catalyzed Ester Hydrolysis. *J. Am. Chem. Soc.* **1975**, *97* (13), 3691–3693.

(339) Charton, M. Steric Effects. 7. Additional V Constants. *J. Org. Chem.* **1976**, *41* (12), 2217–2220.

(340) Kutter, E.; Hansch, C. Steric Parameters in Drug Design. Monoamine Oxidase Inhibitors and Antihistamines. *J. Med. Chem.* **1969**, *12* (4), 647–652.

(341) Verloop, A. & Tipker, J. *Biological Activity and Chemical Structure*; 1977.

(342) Tipker, A. & Verloop, J. *QSAR in Drug Design and Toxicology*; 1987.

(343) Pauling, L.; Corey, R. B. Atomic Coordinates and Structure Factors for Two Helical Configurations of Polypeptide Chains. In *Proceedings of the National Academy of Sciences of the United States of America*; 1951; Vol. 37, pp 235–240.

(344) Ardkhean, R.; Roth, P. M. C.; Maksymowicz, R. M.; Curran, A.; Peng, Q.; Paton, R. S.; Fletcher, S. P. Enantioselective Conjugate Addition Catalyzed by a Copper Phosphoramidite Complex: Computational and Experimental Exploration of Asymmetric Induction. *ACS Catal.* **2017**, *7* (10), 6729–6737.

(345) Ardkhean, R.; Mortimore, M.; Paton, R. S.; Fletcher, S. P. Formation of Quaternary Centres by Copper Catalysed Asymmetric Conjugate Addition to β-Substituted Cyclopentenones with the Aid of a Quantitative Structure-Selectivity Relationship. *Chem. Sci.* **2018**, *9*, 2628–2632.

(346) Brethomé, A. V.; Fletcher, S. P.; Paton, R. S. Conformational Effects on Physical-Organic Descriptors: The Case of Sterimol Steric Parameters. *ACS Catal.* **2019**, *9* (3), 2313–2323.

(347) London, F. Théorie Quantique Des Courants Interatomiques Dans Les Combinaisons Aromatiques. *J. Phys. Radium* **1937**, *8* (10), 397–409.

(348) Ditchfield, R. Molecular Orbital Theory of Magnetic Shielding and Magnetic Susceptibility. *J. Chem. Phys.* **1972**, *56*, 5688.

(349) Helgaker, T.; Jaszunski, M.; Ruud, K. Ab Initio Methods for the Calculation of NMR Shielding and Indirect Spin-Spin Coupling Constants. *Chem. Rev.* **1999**, *99* (1), 293–352.

(350) Pulay, P.; Hinton, J. F.; Wolinski, K. *Efficient Implementation of the GIAO Method for Magnetic Properties: Theory and Application*; Springer Netherlands, 1993.

(351) Auer, A. A.; Gauss, J.; Stanton, J. F. Quantitative Prediction of Gas-Phase Nuclear Magnetic Shielding Constants. *J. Chem. Phys.* **2003**, *118*, 10407.

(352) Flaig, D.; Maurer, M.; Hanni, M.; Braunger, K.; Kick, L.; Thubauville, M.; Ochsenfeld, C. Benchmarking Hydrogen and Carbon NMR Chemical Shifts at HF, DFT, and MP2 Levels. *J. Chem. Theory Comput.* **2014**, *10* (2), 572–578.

(353) Neese, F. The ORCA Program System. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2012**, *2* (1), 73–78.

(354) Freedman, D.; Pisani, R.; Purves, R. *Statistics*; W.W. Norton, 1998.

(355) Lin, T. S.; Coley, C. W.; Mochigase, H.; Beech, H. K.; Wang, W.; Wang, Z.; Woods, E.; Craig, S. L.; Johnson, J. A.; Kalow, J. A.; Jensen, K. F.; Olsen, B. D. BigSMILES: A Structurally-Based Line Notation for Describing Macromolecules. *ACS Cent. Sci.* **2019**, *5* (9), 1523–1531.

(356) Golbraikh, A.; Tropsha, A. Beware of Q2! *J. Mol. Graph. Model.* **2002**, *20* (4), 269–276.

(357) Kotsiantis, S. B. Supervised Machine Learning: A Review of Classification Techniques. *Informatica* **2007**, *31*, 249–268.

(358) Criminisi, A.; Shotton, J.; Konukoglu, E. *Decision Forests: A Unified Framework for*

Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning; 2011; Vol. 7.

(359) Dietterich, T. Overfitting and Undercomputing in Machine Learning. *ACM Comput. Survevs* **1995**, *27* (3), 326–327.

(360) Hawkins, D. M. The Problem of Overfitting. *J. Chem. Inf. Comput. Sci.* **2004**, *44* (1), 1–12.

(361) Emmert-Streib, F.; Dehmer, M. Machine Learning & Knowledge Extraction High-Dimensional LASSO-Based Computational Regression Models: Regularization, Shrinkage, and Selection. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 359–383.

(362) Yuan, G. X.; Ho, C. H.; Lin, C. J. Recent Advances of Large-Scale Linear Classification. In *Proceedings of the IEEE*; 2012; Vol. 100, pp 2584–2603.

(363) Cramer, J. S. The Origins of Logistic Regression. *Tinbergen Inst. Discuss. Pap.* **2002**, *119*, 4.

(364) Boser, B. E.; Guyon, I. M.; Vapnik, V. N. Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*; 1992; pp 144–152.

(365) Schmidt, M.; Le Roux, N.; Bach, F. Minimizing Finite Sums with the Stochastic Average Gradient. *Math. Program.* **2017**, *162* (1–2), 83–112.

(366) Kiefer, J.; Wolfowitz, J. Stochastic Estimation of the Maximum of a Regression Function. *Ann. Math. Stat.* **1952**, *23* (3), 462–466.

(367) Bordes, A.; Ertekin, S.; Weston, J.; Botton, L. Fast Kernel Classifiers with Online and Active Learning. *J. Mach. Learn. Res.* **2005**, *6*, 1579–1619.

(368) Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S. Online Passive Aggressive Algorithms. *J. Mach. Learn. Res.* **2006**, *7*, 551–585.

(369) Wang, X.; Benning, M. *Generalised Perceptron Learning*; 2020.

(370) Fisher, R. The Use of Multiple Measurements in Taxonomic Problems. *Ann. Eugen.* **1936**, *7* (2), 179–188.

(371) Geisser, S. Posterior Odds for Multivariate Normal Classifications. *J. R. Stat. Soc. Ser. B* **1964**, *26* (1), 69–76.

(372) Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. *Classification and Regression Trees*; CRC Press, 1984.

(373) Fix, E.; Hodges, J. L. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. *Int. Stat. Rev. / Rev. Int. Stat.* **1989**, *57* (3), 238–247.

(374) Chawla, N.; Bowyer, K.; LO Hall, L. SMOTE: Synthetic Minority over-Sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357.

(375) He, H.; Bai, Y.; Garcia, E. A.; Li, S. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In *Proceedings of the International Joint Conference on Neural Networks*; 2008; pp 1322–1328.

(376) Hart, P. The Condensed Nearest Neighbor Rule. *IEEE Trans. Inf. Theory* **1968**, *14*, 515–516.

(377) Smith, M. R.; Martinez, T.; Giraud-Carrier, C. An Instance Level Analysis of Data Complexity. *Mach. Learn.* **2014**, *95*, 225–256.

(378) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Vincent, M.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. *Scikit-Learn: Machine Learning in Python*; 2011; Vol. 12. http://scikit-learn.sourceforge.net. (accessed 2021-01-07).

(379) Plehiers, P.; Symoens, S.; Amghizar, I.; Marin, G.; Stevens, C.; VanGeem, K. Artificial Intelligence in Steam Cracking Modeling: A Deep Learning Algorithm for Detailed Effluent

Prediction. *Engineering* **2019**, *5* (6), 1027–1040.

(380) Fletcher, R. *Practical Methods of Optimization*; John Wiley & Sons, Ltd: West Sussex, 2000.

(381) Kingma, D. P.; Ba, J. L. Adam: A Method for Stochastic Optimization. *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* **2014**.

(382) Fukushima, K. Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biol. Cybern.* **1980**, *36* (4), 193–202.

(383) Santos, G. L.; Endo, P. T.; Monteiro, K. H. de C.; Rocha, E. da S.; Silva, I.; Lynn, T. Accelerometer-Based Human Fall Detection Using Convolutional Neural Networks. *Sensors* **2019**, *19* (7), 1644.

(384) Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; Monfardini, G. The Graph Neural Network Model. *IEEE Trans. Neural Networks* **2009**, *20* (1), 61–80.

(385) Gori, M.; Monfardini, G.; Scarselli, F. A New Model for Learning in Graph Domains. In *EEE International Joint Conference on Neural Networks*; 2005; pp 729–734.

(386) Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Advances in Neural Information Processing Systems*; 2015; p 28.

(387) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular Graph Convolutions: Moving beyond Fingerprints. *J. Comput. Aided. Mol. Des.* **2016**, *30* (8), 595–608.

(388) Do, K.; Tran, T.; Venkatesh, S. Graph Transformation Policy Network for Chemical Reaction Prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; Association for Computing Machinery, 2019; pp 750–760.

(389) Jin, W.; Yang, K.; Barzilay, R.; Jaakkola, T. Learning Multimodal Graph-to-Graph Translation for Molecular Optimization. In *7th International Conference on Learning Representations*; 2019.

(390) Fout, A.; Byrd, J.; B, S. Protein Interface Prediction Using Graph Convolutional Networks. In *31st Conference on Neural Information Processing Systems*; 2017.

(391) Rittig, J. G.; Ben Hicham, K.; Schweidtmann, A. M.; Dahmen, M.; Mitsos, A. Graph Neural Networks for Temperature-Dependent Activity Coefficient Prediction of Solutes in Ionic Liquids. *Comput. Chem. Eng.* **2023**, *171*, 108153.

(392) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*; 2017; pp 1263–1272.

(393) Schweidtmann, A. M.; Rittig, J. G.; Kö, A.; Grohe, M.; Mitsos, A.; Dahmen, M. Graph Neural Networks for Prediction of Fuel Ignition Quality. *Energy and Fuels* **2020**, *34* (9), 11395–11407.

(394) Cho, K.; VanMerriënboer, B.; Bahdanau, D. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*; 2014; pp 103–111.

(395) Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations*; 2017.

(396) Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Stat, P. L. Graph Attention Networks. In *6th International Conference on Learning Represe*; 2018.

(397) Hamilton, W.; Ying, Z.; Neural, J. L.-A. in; 2017, U. Inductive Representation Learning on Large Graphs. In *31st Conference on Neural Information Processing Systems*; 2017; pp 1025–1035.

(398) Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L. Graph Neural Networks: A

Review of Methods and Applications. *AI Open* **2020**, *1*, 57–81.

(399) Sun, L.; Dou, Y.; Yang, C.; Zhang, K.; Wang, J.; Liu, Y.; Yu, P. S.; He, L.; Li, B. Adversarial Attack and Defense on Graph Data: A Survey. *IEEE Trans. Knowl. Data Eng.* **2022**, 1–20.

(400) Yuan, H.; Yu, H.; Gui, S.; Pattern, S. J. Explainability in Graph Neural Networks: A Taxonomic Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 5782–5799.

(401) Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; Leskovec, J. Strategies for Pre-Training Graph Neural Networks. In *International Conference on Learning Representations*; 2020.

(402) Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *24* (2), 123–140.

(403) Geurts, P.; Ernst, D.; Wehenkel, L. Extremely Randomized Trees. *Mach. Learn.* **2006**, *63* (1), 3–42.

(404) JH, F. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29* (5), 1189–1232.

(405) Freund, Y.; Schapire, R. E. A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting. *J. Comput. Syst.* **1997**, *55* (1), 119–139.

(406) Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 2016; pp 785–794.

(407) Kiyak, E. Data Mining and Machine Learning for Software Engineering. In *Data Mining*; IntechOpen, 2020.

(408) Bergstra, J.; Bengio, Y. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.

(409) Yu, T.; Zhu, H. Hyper-Parameter Optimization: A Review of Algorithms and Applications. *arXiv* **2020**, *2003.05689*.

(410) Jones, D. R.; Schonlau, M.; Welch, W. J. Efficient Global Optimization of Expensive Black-Box Functions. *J. Glob. Optim.* **1998**, *13* (4), 455–492.

(411) Tropsha, A.; Gramatica, P.; Gombar, V. K. The Importance of Being Earnest: Validation Is the Absolute Essential for Successful Application and Interpretation of QSPR Models. *Mol. Inform.* **2003**, *22* (1), 69–77.

(412) Gordon, A. D. *Classification*, 2nd ed.; Chapman & Hall/CRC, 1999.

(413) Hoffmann, R.; Malrieu, J. P. Simulation vs. Understanding: A Tension, in Quantum Chemistry and Beyond. Part A. Stage Setting. *Angew. Chemie Int. Ed.* **2020**, *59* (31), 12590–12610.

(414) Fey, N.; Koumi, A.; Malkov, A. V.; Moseley, J. D.; Nguyen, B. N.; Tyler, S. N. G.; Willans, C. E. Mapping the Properties of Bidentate Ligands with Calculated Descriptors (LKB-Bid). *Dalt. Trans.* **2020**, *49* (24), 8169–8178.

(415) VanderMaaten, L.; G Hinton, G. Visualizing Data Using T-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

(416) Liao, T.; Wei, Y.; Luo, M.; Zhao, G. P.; Zhou, H. Tmap: An Integrative Framework Based on Topological Data Analysis for Population-Scale Microbiome Stratification and Association Studies. *Genome Biol.* **2019**, *20*, 293.

(417) Lundberg, S. M.; Allen, P. G.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In *31st Conference on Neural Information Processing Systems*; 2017; pp 4768–4777.

(418) Ancona, M.; Ceolini, E.; Öztireli, C.; Gross, M. Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science. In *Gradient-Based*

Attribution Methods; Springer Verlag, 2019; Vol. 11700, pp 169–191.

(419) Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K. R.; Samek, W. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLoS One* **2015**, *10* (7), 1–46.

(420) Kluyver, T.; Ragan-Kelley, B.; Fernando, P.; Granger, B.; Bussonnier, M.; Frederic, J.; Willing, C. Jupyter Notebooks – a Publishing Format for Reproducible Computational Workflows. Players, Agents and Agendas 2016, pp 87–90.

(421) Haghighatlari, M.; Vishwakarma, G.; Altarawy, D.; Subramanian, R.; Kota, B. U.; Sonpal, A.; Setlur, S.; Hachmann, J. ChemML: A Machine Learning and Informatics Program Package for the Analysis, Mining, and Modeling of Chemical and Materials Data. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2020**, *10* (4), e1458.

(422) Khatib, M. El; de Jong, W. A. ML4Chem: A Machine Learning Package for Chemistry and Materials Science. *arXiv* **2020**, *2003.13388*.

(423) Korshunova, M.; Ginsburg, B.; Tropsha, A.; Isayev, O. OpenChem: A Deep Learning Toolkit for Computational Chemistry and Drug Design. *J. Chem. Inf. Model.* **2021**, *61* (1), 7–13.

(424) Stokes, J. M.; Yang, K.; Swanson, K.; Jin, W.; Cubillos-Ruiz, A.; Donghia, N. M.; MacNair, C. R.; French, S.; Carfrae, L. A.; Bloom-Ackerman, Z.; Tran, V. M.; Chiappino-Pepe, A.; Badran, A. H.; Andrews, I. W.; Chory, E. J.; Church, G. M.; Brown, E. D.; Jaakkola, T. S.; Barzilay, R.; Collins, J. J. A Deep Learning Approach to Antibiotic Discovery. *Cell* **2020**, *180* (4), 688–702.

(425) Bilodeau, C.; Kazakov, A.; Mukhopadhyay, S.; Emerson, J.; Kalantar, T.; Muzny, C.; Jensen, K. Machine Learning for Predicting the Viscosity of Binary Liquid Mixtures. *Chem. Eng. J.* **2023**, *464*, 142454.

(426) *GitHub - Mariewelt/OpenChem: OpenChem: Deep Learning toolkit for Computational Chemistry and Drug Design Research*. https://github.com/Mariewelt/OpenChem (accessed 2023-05-02).

(427) *Syracuse Research Corporation. Physical/Chemical Property Database (PHYSPROP)*; SRC Environmental Science Center: Syracuse, NY, 1994.

(428) *GitHub - aspuru-guzik-group/kraken: Code to compute electronic and steric features to create a database of ligands and their properties*. https://github.com/aspuru-guzik-group/kraken (accessed 2023-05-02).

(429) McKinney, W. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*; 2010; pp 51–56.

(430) Harris, C.R., Millman, K.J., van der Walt, S. J. et al. Array Programming with NumPy. *Nature* **2020**, *585*, 357–362.

(431) Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9* (3), 90–95.

(432) Seabold; Skipper; Perktold, J. Statsmodels: Econometric and Statistical Modeling with Python. In *Proceedings of the 9th Python in Science Conference*; 2010.

(433) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; Walt, S. J. van der; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C.; Polat, İ.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; Mulbreg, P. van. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17* (3), 261–272.

(434) Vachal, P.; Jacobsen, E. N. Enantioselective Catalytic Addition of HCN to Ketoimines. Catalytic Synthesis of Quaternary Amino Acids. *Org. Lett.* **2000**, *2* (6), 867–870.

(435) Zuend, S.; EN, J. Mechanism of Amido-Thiourea Catalyzed Enantioselective Imine Hydrocyanation: Transition State Stabilization via Multiple Non-Covalent Interactions. *J. Am. Chem. Soc.* **2009**, *131* (42), 15358–15374.

(436) Sigman, M. S.; Jacobsen, E. N. Schiff Base Catalysts for the Asymmetric Strecker Reaction Identified and Optimized from Parallel Synthetic Libraries. *J. Am. Chem. Soc.* **1998**, *120* (19), 4901–4902.

(437) Sigman, M.; Vachal, P.; Jacobsen, E. A General Catalyst for the Asymmetric Strecker Reaction. *Angew. Chemie* **2000**, *112* (7), 1336–1338.

(438) Pan, S. C.; List, B. The Catalytic Acylcyanation of Imines. *Chem. - An Asian J.* **2008**, *3* (2), 430–437.

(439) *GitHub - kjelljorner/morfeus: A Python package for calculating molecular features*. https://github.com/kjelljorner/morfeus (accessed 2023-05-12).

(440) Seayad, I.; List, B. Asymmetric Organocatalysis. *Org. Biomol. Chem.* **2005**, *3* (5), 719–724.

(441) Taylor, M. S.; Jacobsen, E. N.; Jacobsen, E. N.; Taylor, M. S. Reaction Mechanisms Asymmetric Catalysis by Chiral Hydrogen-Bond Donors. *Angew. Chemie Int. Ed.* **2006**, *45* (10), 1520–1543.

(442) Gaunt, M.; Johansson, C.; McNally, A.; Vo, N. Enantioselective Organocatalysis. *Drug Discov. Today* **2007**, *12* (1–2), 8–27.

(443) Dalko, P. I.; Moisan, L. In the Golden Age of Organocatalysis. *Angew. Chemie Int. Ed.* **2004**, *43* (39), 5138–5175.

(444) List, B. Introduction: Organocatalysis. *Chem. Rev.* **2007**, *107* (12), 5413–5415.

(445) J. K. Nørskov, F. Studt, F. Abild-Pedersen, and T. B. *Fundamental Concepts in Heterogeneous Catalysis*; John Wiley & Sons, Inc., 2014.

(446) Deutschmann, O.; Knözinger, H.; Kochloefl, K.; Turek, T. Heterogeneous Catalysis and Solid Catalysts. In *Ullmann's Encyclopedia of Industrial Chemistry*; Wiley-VCH Verlag GmbH, 2009.

(447) Blaser, H.; Pugin, B. The Industrial Application of Heterogeneous Enantioselective Catalysts. In *Handbook of Asymmetric Heterogeneous Catalysis*; John Wiley & Sons, Inc., 2008; pp 413–437.

(448) Sheldon, R.; Bekkum, H. Van. *Fine Chemicals through Heterogeneous Catalysis*; WILEY-VCH Verlag GmbH, 2008.

(449) Romano, U.; Ricci, M. Industrial Applications. In *Liquid Phase Oxidation via Heterogeneous Catalysis: Organic Synthesis and Industrial Applications*; John Wiley & Sons, Inc, 2013; pp 451–506.

(450) Erisman, J. W.; Sutton, M. A.; Galloway, J.; Klimont, Z.; Winiwarter, W. How a Century of Ammonia Synthesis Changed the World. *Nat. Geosci.* **2008**, *1*, 636–639.

(451) Honkala, K.; Hellman, A.; Remediakis, I. N.; Logadottir, A.; Carlsson, A.; Dahl, S.; Christensen, C. H.; Nørskov, J. K. Ammonia Synthesis from First-Principles Calculations. *Science (80-.).* **2005**, *307* (5709), 555–558.

(452) Schlögl, R. Catalytic Synthesis of Ammonia - A "Never-Ending Story"? *Angew. Chemie Int. Ed.* **2003**, *42* (18), 2004–2008.

(453) Xia, Q.; Ge, H.; Ye, C.; Liu, Z.; Su, K. Advances in Homogeneous and Heterogeneous Catalytic Asymmetric Epoxidation. *Chem. Rev.* **2005**, *105* (5), 1603–1662.

(454) Christopher, P.; Linic, S. Engineering Selectivity in Heterogeneous Catalysis: Ag Nanowires as Selective Ethylene Epoxidation Catalysts. *J. Am. Chem. Soc.* **2008**, *130* (34), 11264–11265.

(455) Oyama, S. Rates, Kinetics, and Mechanisms of Epoxidation: Homogeneous, Heterogeneous, and Biological Routes. In *Mechanisms in Homogeneous and Heterogeneous Epoxidation Catalysis*; Elsevier B.V., 2008; pp 3–99.

(456) Angelici, R. J. Heterogeneous Catalysis of the Hydrodesulfurization of Thiophenes in Petroleum: An Organometallic Perspective of the Mechanism. *Acc. Chem. Res.* **1988**, *21* (11), 387–394.

(457) Yadav, N.; Yadav, A.; Kumar, M. History of Neural Networks. In *An Introduction to Neural Network Methods for Differential Equations*; Springer, 2015; pp 13–15.

(458) Zhang, L.; Zhao, Z. J.; Gong, J. Nanostructured Materials for Heterogeneous Electrocatalytic CO2 Reduction and Their Related Reaction Mechanisms. *Angew. Chemie Int. Ed.* **2017**, *56* (38), 11326–11353.

(459) Mckone, J. R.; Marinescu, S. C.; Brunschwig, B. S.; Winkler, J. R.; Gray, H. B. Earth-Abundant Hydrogen Evolution Electrocatalysts. *Chem. Sci.* **2014**, *5*, 865.

(460) Zeng, M.; Li, Y. Recent Advances in Heterogeneous Electrocatalysts for Hydrogen Evolution Reaction. *J. Mater. Chem. A* **2015**, *3* (29), 14942–14962.

(461) HALPERN, Jackh. C. by C. C. Homogeneous Catalysis. In *Homogeneous Catalysis by Coordination Compounds*; Advances in Chemistry, 1974; pp 1–24.

(462) Chen, J. Homogeneous and Heterogeneous Catalysis: Teachings of the Thermal Energy and Power Engineering Course. *Int. J. Educ. Pedagog. Sci.* **2014**, *8* (12), 3923–3926.

(463) Haibach, M. C.; Kundu, S.; Brookhart, M.; Goldman, A. S. Alkane Metathesis by Tandem Alkane-Dehydrogenation-Olefin-Metathesis Catalysis and Related Chemistry. *Acc. Chem. Res.* **2012**, *45* (6), 947–958.

(464) Starks, C. M. Phase-Transfer Catalysis. I. Heterogeneous Reactions Involving Anion Transfer by Quaternary Ammonium and Phosphonium Salts. *J. Am. Chem. Soc.* **1971**, *93* (1), 195–199.

(465) Herriott, A. W.; Picker, D. Phase Transfer Catalysis. An Evaluation of Catalysis. *J. Am. Chem. Soc.* **1975**, *97* (9), 2345–2349.

(466) Halpern, M. Phase-Transfer Catalysis. In *Ullmann's Encyclopedia of Industrial Chemistry*; Wiley-VCH Verlag GmbH & Co.KGaA: Weinheim, Germany, 2000.

(467) Makosza, M. Phase-Transfer Catalysis. A General Green Methodology in Organic Synthesis. *Pure Appl. Chem.* **2000**, *72* (7), 1399–1403.

(468) Makosza, M. Reactions of Organic Anions. XI. Catalytic Alkylation of Indene. *Tetrahedron Lett.* **1966**, *7* (38), 4621–4624.

(469) Ooi, T.; Ohara, D.; Tamura, M.; Maruoka, K. Design of New Chiral Phase-Transfer Catalysts with Dual Functions for Highly Enantioselective Epoxidation of α,β-Unsaturated Ketones. *J. Am. Chem. Soc.* **2004**, *126* (22), 6844–6845.

(470) Wang, X.; Lan, Q.; Shirakawa, S.; Maruoka, K. Chiral Bifunctional Phase Transfer Catalysts for Asymmetric Fluorination of β-Keto Esters. *Chem. Commun.* **2009**, *46* (2), 321–323.

(471) Wang, Y. M.; Wu, J.; Hoong, C.; Rauniyar, V.; Toste, F. D. Enantioselective Halocyclization Using Reagents Tailored for Chiral Anion Phase-Transfer Catalysis. *J. Am. Chem. Soc.* **2012**, *134* (31), 12928–12931.

(472) Fleischmann, M.; Drettwan, D.; Sugiono, E.; Rueping, M.; Gschwind, R. M.; Fleischmann, M.; Drettwan, D.; Gschwind, R. M.; Sugiono, E.; Rueping, M. Brønsted Acid Catalysis: Hydrogen Bonding versus Ion Pairing in Imine Activation. *Angew. Chemie Int. Ed.* **2011**, *123* (28), 6488–6493.

(473) Doyle, A. G.; Jacobsen, E. N. Small-Molecule H-Bond Donors in Asymmetric Catalysis. *Chem. Rev.* **2007**, *107* (12), 5713–5743.

(474) Taylor, M. S.; Jacobsen, E. N. Asymmetric Catalysis by Chiral Hydrogen-Bond Donors. *Angew. Chemie Int. Ed.* **2006**, *45* (10), 1520–1543.

(475) Scheiner, S. *Hydrogen Bonding: A Theoretical*; Oxford University Press, 1997.

(476) Maruoka, K.; Ooi, T. Enantioselective Amino Acid Synthesis by Chiral Phase-Transfer Catalysis. *Chem. Rev.* **2003**, *103* (8), 3013–3028.

(477) Giese, B. E. V. Dehmlow, S. S. Dehmlow: Phase Transfer Catalysis, Vol. 11 Der Reihe: Monographs in Modern Chemistry. Verlag Chemie, Weinheim, Deerfield Beach, Basel 1980. 316 Seiten, Preis: DM 138. *Berichte der Bunsengesellschaft für Phys. Chemie* **1981**, *85* (1), 95–95.

(478) Pupo, G.; Gouverneur, V. Hydrogen Bonding Phase-Transfer Catalysis with Alkali Metal Fluorides and Beyond. *J. Am. Chem. Soc.* **2022**, *144* (12), 5200–5213.

(479) Kee, C. W. Molecular Understanding and Practical In Silico Catalyst Design in Computational Organocatalysis and Phase Transfer Catalysis—Challenges and Opportunities. *Molecules* **2023**, *28* (4), 1715.

(480) Ogawa, Y.; Tokunaga, E.; Kobayashi, O.; Hirai, K.; Shibata, N. Current Contributions of Organofluorine Compounds to the Agrochemical Industry. *iScience* **2020**, *23* (9), 101467.

(481) Ghuge, N.; Katkar, H.; Chokshi, R.; Mane, A. Fluorination. *Sect. Themat.* **2022**.

(482) Miller, P. W.; Long, N. J.; Vilar, R.; Gee, A. D. Imaging Methods Synthesis of 11 C, 18 F, 15 O, and 13 N Radiolabels for Positron Emission Tomography. *Angew. Chemie Int. Ed.* **2008**, *47* (47), 8998–9033.

(483) Deng, X.; Rong, J.; Wang, L.; Vasdev, N.; Zhang, L.; Josephson, L.; Liang, S. H. Chemistry for Positron Emission Tomography: Recent Advances in 11 C-, 18 F-, 13 N-, and 15 O-Labeling Reactions. *Angew. Chemie Int. Ed.* **2019**, *58* (9), 2580–2605.

(484) Campbell, M. G.; Ritter, T. Late-Stage Fluorination: From Fundamentals to Application. *Org. Process Res. Dev.* **2014**, *18* (4), 474–480.

(485) Clark, J. H. Fluoride Ion as a Base in Organic Synthesis. *Chem. Rev.* **1980**, *80* (5), 429–452.

(486) Harsanyi, A.; Sandford, G. Organofluorine Chemistry: Applications, Sources and Sustainability. *Green Chem.* **2015**, *17* (4), 2081–2086.

(487) Szpera, R.; Moseley, D. F. J.; Smith, L. B.; Sterling, A. J.; Gouverneur, V. The Fluorination of C−H Bonds: Developments and Perspectives. *Angew. Chemie - Int. Ed.* **2019**, *58* (42), 14824–14848.

(488) Pupo, G.; Ibba, F.; Ascough, D. M. H.; Vicini, A. C.; Ricci, P.; Christensen, K. E.; Pfeifer, L.; Morphy, J. R.; Brown, J. M.; Paton, R. S.; Gouverneur, V. Asymmetric Nucleophilic Fluorination under Hydrogen Bonding Phase-Transfer Catalysis. *Science (80-.).* **2018**, *360* (6389), 638–642.

(489) Pupo, G.; Vicini, A. C.; Ascough, D. M. H.; Ibba, F.; Christensen, K. E.; Thompson, A. L.; Brown, J. M.; Paton, R. S.; Gouverneur, V. Hydrogen Bonding Phase-Transfer Catalysis with Potassium Fluoride: Enantioselective Synthesis of β-Fluoroamines. *J. Am. Chem. Soc.* **2019**, *141* (7), 2878–2883.

(490) Gawley, R. E. Do the Terms "% Ee" and "% de" Make Sense as Expressions of Stereoisomer Composition or Stereoselectivity? *J. Org. Chem.* **2006**, *71* (6), 2411–2416.

(491) Alexander, D. L. J.; Tropsha, A.; Winkler, D. A. Beware of R2: Simple, Unambiguous Assessment of the Prediction Accuracy of QSAR and QSPR Models. *J. Chem. Inf. Model.* **2015**, *55* (7), 1316–1322.

(492) Zhu, X.; Robinson, D. A.; McEwan, A. R.; O'Hagan, D.; Naismith, J. H. Mechanism of Enzymatic Fluorination in Streptomyces Cattleya. *J. Am. Chem. Soc.* **2007**, *129* (47), 14597–

14604.

(493) Young, T. A.; Silcock, J. J.; Sterling, A. J.; Duarte, F. AutodE: Automated Calculation of Reaction Energy Profiles— Application to Organic and Organometallic Reactions. *Angew. Chemie - Int. Ed.* **2021**, *60* (8), 4266–4274.

(494) Zuend, S.; Coughlin, M.; Lalonde, M.; Jacobsen, E. Scaleable Catalytic Asymmetric Strecker Syntheses of Unnatural α-Amino Acids. *Nature* **2009**, *461* (7266), 968–970.

(495) Brak, K.; Jacobsen, E. N. Asymmetric Ion-Pairing Catalysis. *Angew. Chemie Int. Ed.* **2013**, *52* (2), 534–561.

(496) Zuend, S. J.; Jacobsen, E. N. Mechanism of Amido-Thiourea Catalyzed Enantioselective Imine Hydrocyanation: Transition State Stabilization via Multiple Non-Covalent Interactions. *J. Am. Chem. Soc.* **2009**, *131* (42), 15358–15374.

(497) Raheem, I. T.; Thiara, P. S.; Peterson, E. A.; Jacobsen, E. N. Enantioselective Pictet-Spengler-Type Cyclizations of Hydroxylactams: H-Bond Donor Catalysis by Anion Binding. *J. Am. Chem. Soc.* **2007**, *129* (44), 13404–13405.

(498) Raheem, I. T.; Thiara, P. S.; Jacobsen, E. N. Regio-and Enantioselective Catalytic Cyclization of Pyrroles onto N-Acyliminium Lons. *Org. Lett.* **2008**, *10* (8), 1577–1580.

(499) Breiman, L. Stacked Regressions. *Mach. Learn.* **1996**, *24* (1), 49–64.

(500) Dietterich, T. G. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems. Lecture Notes in Computer Science*; Springer, 2000; Vol. 1857, pp 1–15.

(501) Schweidtmann, A. M.; Rittig, J. G.; König, A.; Grohe, M.; Mitsos, A.; Dahmen, M. Graph Neural Networks for Prediction of Fuel Ignition Quality. *Energy and Fuels* **2020**, *34* (9), 11395–11407.

(502) Fey, M.; Lenssen, J. E. Fast Graph Representation Learning with PyTorch Geometric. *arXiv* **2019**, 1903.02428v3.

(503) Xu, K.; Jegelka, S.; Hu, W.; Leskovec, J. How Powerful Are Graph Neural Networks? In *7th International Conference on Learning Representations*; **2019**.

(504) Ying, Z.; Bourgeois, D.; You, J.; Zitnik, M.; Leskovec, J. GNNExplainer: Generating Explanations for Graph Neural Networks. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.

(505) Yuan, H.; Tang, J.; Hu, X.; Ji, S. XGNN: Towards Model-Level Explanations of Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; ACM: New York, NY, USA, **2020**; pp 430–438.

(506) Pfeifer, L.; Engle, K. M.; Pidgeon, G. W.; Sparkes, H. A.; Thompson, A. L.; Brown, J. M.; Gouverneur, V. Hydrogen-Bonded Homoleptic Fluoride-Diarylurea Complexes: Structure, Reactivity, and Coordinating Power. *J. Am. Chem. Soc.* **2016**, *138* (40), 13314–13325.

(507) Engle, K. M.; Pfeifer, L.; Pidgeon, G. W.; Giuffredi, G. T.; Thompson, A. L.; Paton, R. S.; Brown, J. M.; Gouverneur, V. Coordination Diversity in Hydrogen-Bonded Homoleptic Fluoride-Alcohol Complexes Modulates Reactivity. *Chem. Sci.* **2015**, *6* (9), 5293–5302.

(508) Harper, K. C.; Bess, E. N.; Sigman, M. S. Multidimensional Steric Parameters in the Analysis of Asymmetric Catalytic Reactions. *Nat. Chem.* **2012**, *4* (5), 366–374.

(509) PITZER, K. S. Origin of the Acentric Factor. In *Phase Equilibria and fluid properties in chemical industry*; ACS, **1977**; pp 1–10.

Appendix A

The supplementary data for this Thesis can be accessed at the following location:
File Name: 'Zavitsanou_thesis_data.zip'

Detailed data references for each chapter are as follows:

1. Data for Methods Chapter:

- Directory: 'Methods-nucleophilicity.zip'
- Contents: Optimized structure coordinates and raw/processed data tables used for the correlations ('nucleophilicities.xlsx').

2. Data for Chapter 3:

- Directory: 'Chapter3.zip'
- Contents: Pythia code (Modules and Jupyter Notebooks), 'DDG.csv,' 'DFTdata.csv,' and 'First_set_full.csv' (input data for Jupyter Notebooks).

3. Data for Chapter 4:

- Directory: 'Chapter4.zip'
- Contents:
  - 'Chapter4.2.zip': Contains optimized structure coordinates ('Coordinates'), raw/processed data for MLModels and results ('Raw-Preprocessed_data.xlsx'), and code for data processing/regression ('Scripts'). Please refer to the README file for instructions on running the code.
  - 'Chapter4.3.zip': Contains optimized structure coordinates ('Coordinates'), raw/processed data for MLModels ('Raw-Preprocessed_data.xlsx'), saved models, Jupyter Notebooks, and metrics/results ('MLModels').
  - 'Chapter4.4'.zip: Contains optimized structure coordinates ('Coordinates'), raw/processed data for MLModels ('Raw-Preprocessed_data.xlsx'), saved models, Jupyter Notebooks, and metrics/results ('MLModels').

4. Data for Chapter 5:

- Directory: 'Chapter5.zip'
- Contents:

- 'Data': Contains 'solubilitydata.xlsx' and 'viscositydata.xlsx,' which include original and processed data for the two datasets.
- 'gnn-final': Includes code for training the GNN model (for solubility and viscosity) and the final results presented in this thesis under the 'final_results' subdirectory. Please refer to the README file for instructions on running the code.
- 'gnn-onegraph': Contains code for training the GNN model when considering one graph to represent both the anion and the cation. Note that these results are only presented in Appendix C1 and are not part of the main Thesis.
- 'Regression-Fingerprints': Includes the Jupyter Notebook (part of *Pythia*) used to train the LASSOCV model and its results. Note that these results are only presented in Appendix C2 and are not part of the main Thesis.

Appendix B

B.1. Fukui - nucleophilicity descriptor

We initially focused on the calculation of nucleophilic Fukui function on the fluoride ion bound to the catalysts considered in Chapter 4.2. The nucleophilicity was not found to correlate with reaction yields (Figure 82, $R^2 = 0.0$). The main reason for the lack of correlation is the fact that the HOMO of the catalyst-F^- complex is located at the urea moiety which in turn is responsible for the nucleophilicity index value. Secondly, given the "hard" nature of the nucleophile, it is not surprising that descriptors based on orbital interactions behave poorly. In these cases, other electronical properties, like charges, are more suitable.

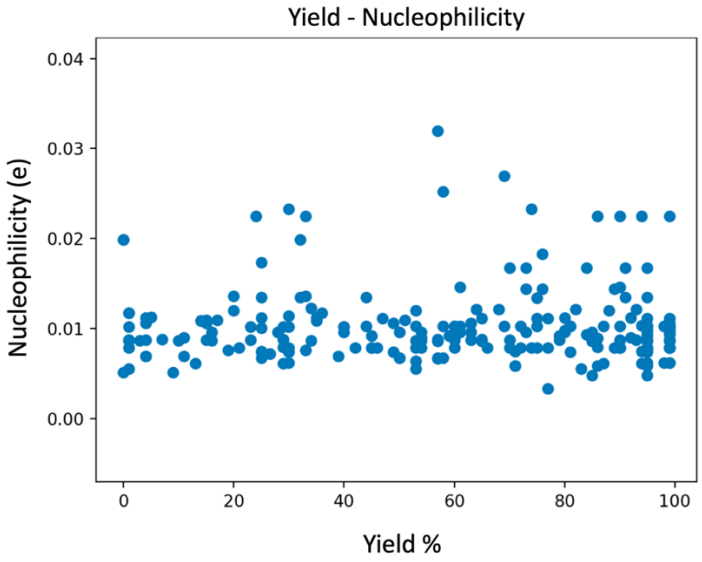


Figure 82: The yield of the 257 reactions against the nucleophilicity.

A poor correlation was observed between the fluoride charge and nucleophilicity (Figure 83, $R^2 = 0.54$). These results suggest that the nucleophilic Fukui function is not a good descriptor for evaluating reactivity on the fluoride centre and, therefore, it was not used as a descriptor for the ML models.

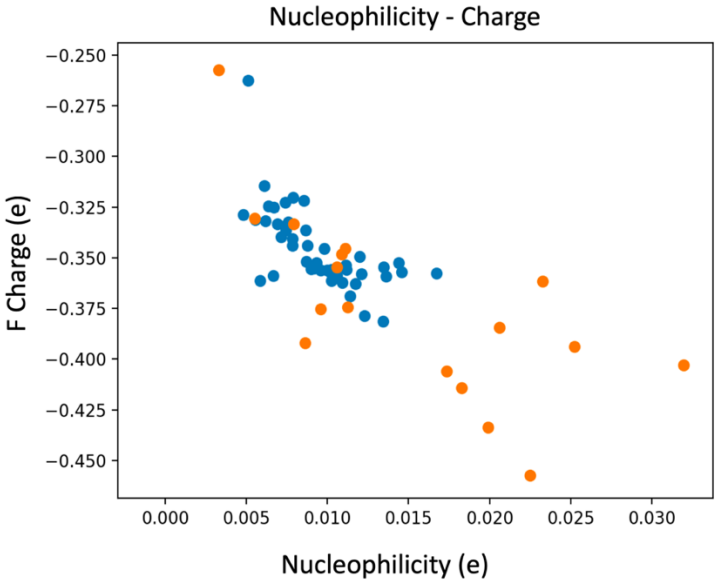
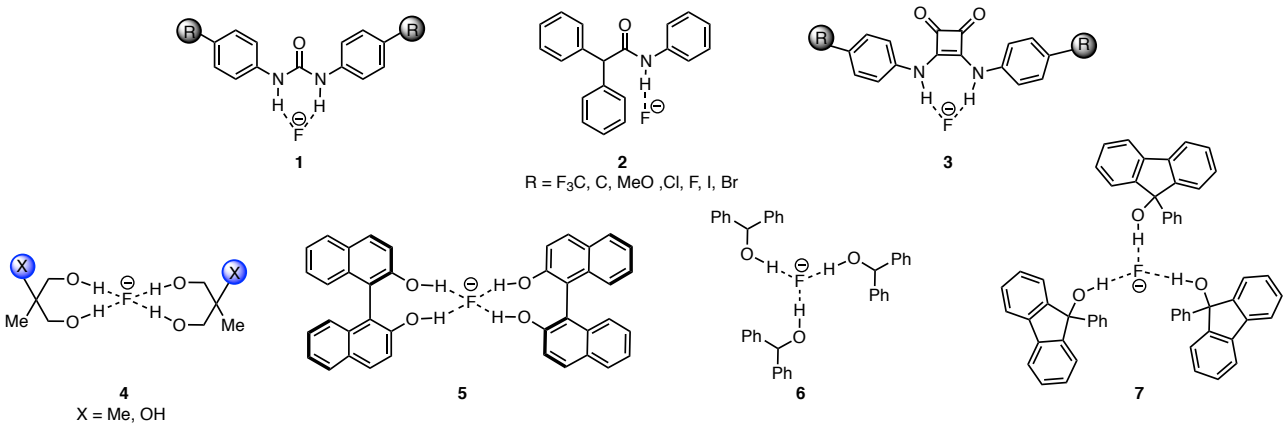


Figure 83: Nucleophilicity against fluoride ion charge for the experimentally tested catalysts. Monourea catalysts in orange, two urea catalysts in blue.

Fluoride-diarylurea & fluoride-alcohol complexes

To select relevant descriptors we also investigated homoleptic fluoride-diarylurea complexes that follow the $S_N2/E2$ mechanism but that do not undergo PTC (Scheme 14 – catalysts 1-3).[1] Correlation between the degree of complexation and the charge on the fluoride was examined (Figure 84a, $R^2 = 0.90$); the six complexes that have an amide functional group or a squaramide structure (Scheme 14 – catalysts 2-3) were disregarded as no BO was found in those cases. The reaction rate and the fluoride charge (Figure 84b, $R^2 = 0.85$) also show high correlation, leading to the conclusion that the fluoride charge highly correlates with the BO and the reaction rate.



Scheme 14: Representatives of the urea-fluoride (1-3) and alcohol-fluoride complexes (4-7).

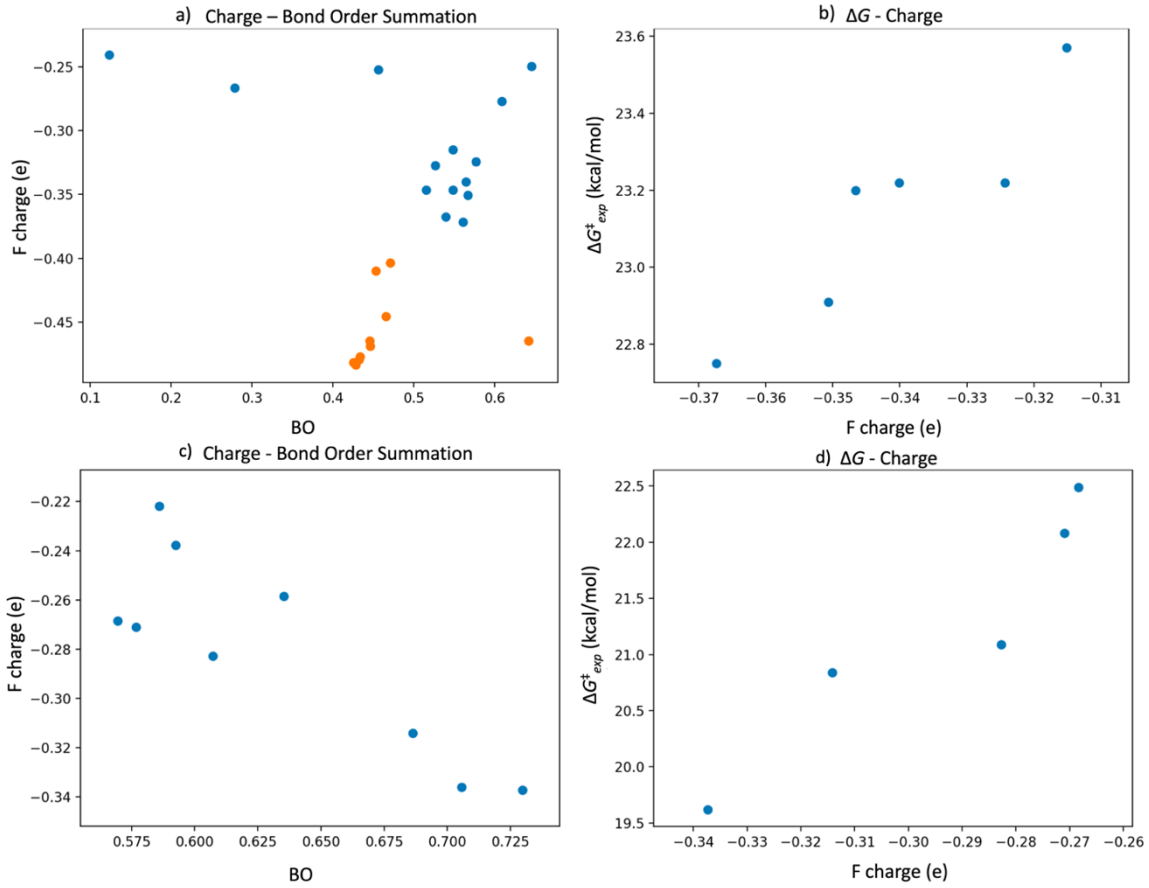


Figure 84: Top: Analysis of the fluoride-diarylurea complexes. a) Summation of H-F⁻ BO against the fluoride charge. In blue the di-coordinate catalysts, in orange the mono-coordinate catalysts. b) Reaction rate in $\Delta G^{\ddagger}$ against the fluoride charge. **Bottom:** Analysis of the fluoride-alcohol complexes. c) Summation of H-F⁻ BO against the fluoride charge, d) Reaction rate in $\Delta G^{\ddagger}$ against the fluoride charge.

We then investigated the hydrogen-bonded fluoride-alcohol complexes (Scheme 14 – catalysts 4-7).[2] The correlation between the fluoride charge and the sum of hydrogen BOs was once again confirmed (Figure 84c, $R^2 = 0.72$). Finally, a weak correlation between the reaction rate and the fluoride charge was obtained (Figure 84d, $R^2 = 0.04$). From the above we conclude that the summation of the BO is a valuable descriptor.

Based on the above findings we decided that nucleophilicity/electrophilicity will not be considered as a descriptor for our ML models, whereas the BO will be included.

B.2. Steric descriptors

To calculate the B_1, B_5, and L values of the 3,5-bis(trifluoromethyl)phenyl in the blue circle (Figure 85a), with wSterimol,[3] one must define the bond from the N (atom A in yellow) to the H (atom B in yellow). The wSterimol code is designed to assume that atom A is a H (which in Figure 85a is not the case), for this to be true the catalyst has to be 'cut' at atom A and replace

the N with a H (Figure 85b). Then wSterimol calculates the distances between atom A and every atom present in the molecule and defines the maximum and minimum distances. It is obvious that these distances cannot be the same in the two cases of Figure 85.

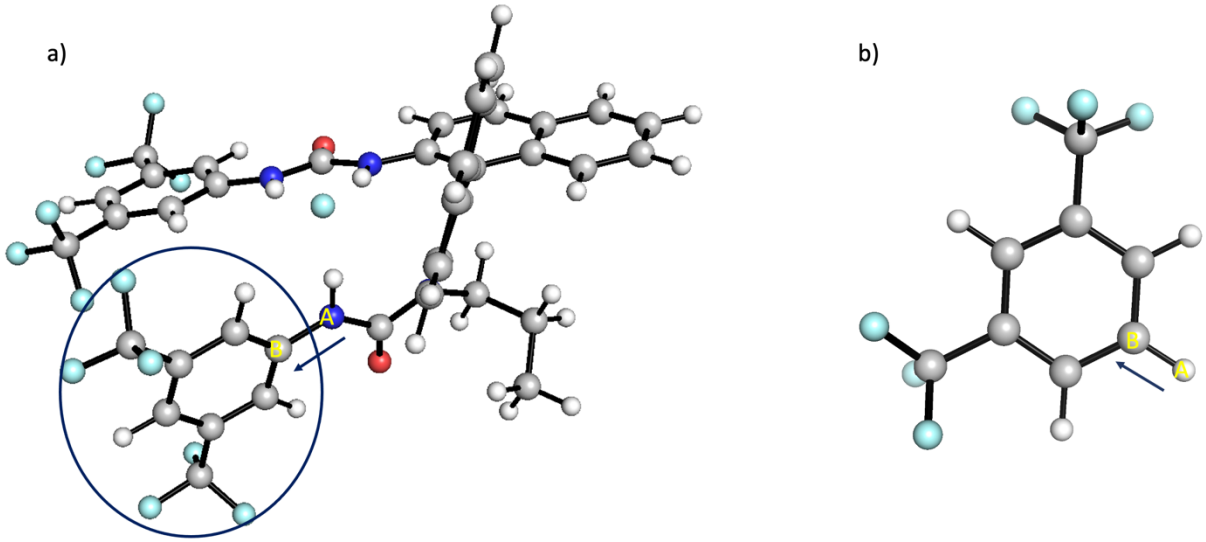


Figure 85: To calculate B_1, B_5, and L values of these two structures we have to identify a bond to scan along. a) The entire catalyst is considered, and the R group of interest is marked in the blue circle. The bond we scan along is between atoms A and B in yellow. b) Only the phenylthingy is considered and the bond we scan along is between atoms A and B in yellow. This time atom A is a H, as the sterimol script assumes.

To identify what B_1, B_5, and L values illustrate when the entire molecule is considered, we generated a script that depicts the atoms as spheres and the vectors as arrows (Figure 86). In this three-figure panel, the same catalyst is shown from different angles of the cartesian space. It seems as the B_5 vector is reaching to the other 3,5-bis(trifluoromethyl)phenyl of the catalyst and not the one for which we scanned the bond along. Such a thing could cause inconsistencies between the descriptors as different catalysts have different sizes and orientation and the B_5 vector can in fact end up showing anywhere on the catalyst.

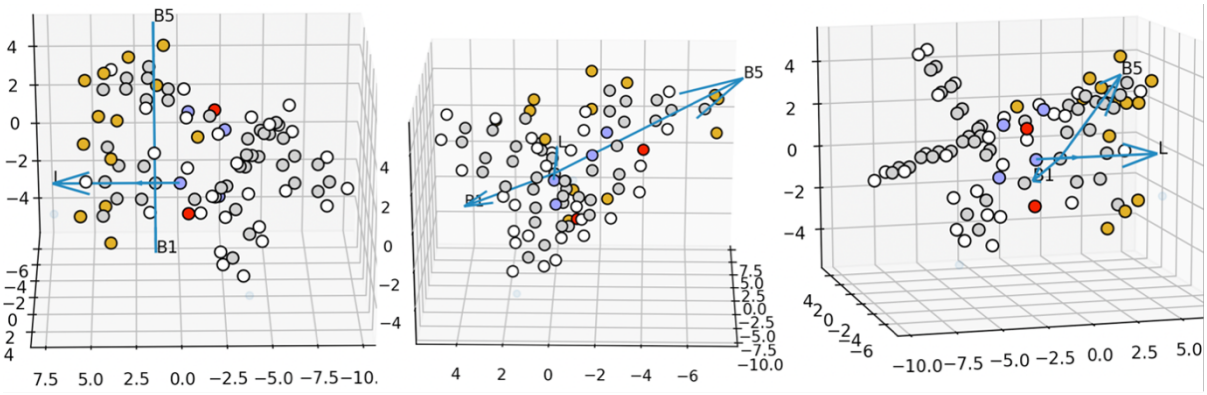


Figure 86: Three different angles of the Cartesian space illustrating the atoms as spheres and the B_1, B_5, and L vectors as arrows. The B_5 vector is pointing away from the R group of interest 3,5-bis(trifluoromethyl)phenyl.

Previous works consider a 'cut' R group and calculate sterimol values only for this group.[4] However, the molecules in these works have a single consistent core, and only specific substitutions take place. The cores of our catalysts differ to great degree ((mono) ureas, thioureas, squareamides) and therefore such a traditional practice cannot be implemented.

To overcome this challenge, we included the idea of directionality along a bond. We decided that using Graph networks would be a quick and efficient method to identify which part of the molecule the calculations should be performed on. We exploited the fact that the Sterimol script already extracts the atoms and bonds from the optimised structures, and we therefore could draw a graph of the molecules where the nodes are atoms, and the edges are the bonds. As the atoms of interest (A-B) are known to us, we can delete the edge between them resulting in two subgraphs (Figure 87). The subgraph which contains atom B is the subset of atoms we are interested in, so all atoms of this subgraph are input for the Stermoil script and the B_1, B_5, and L values are calculated for this subgraph.

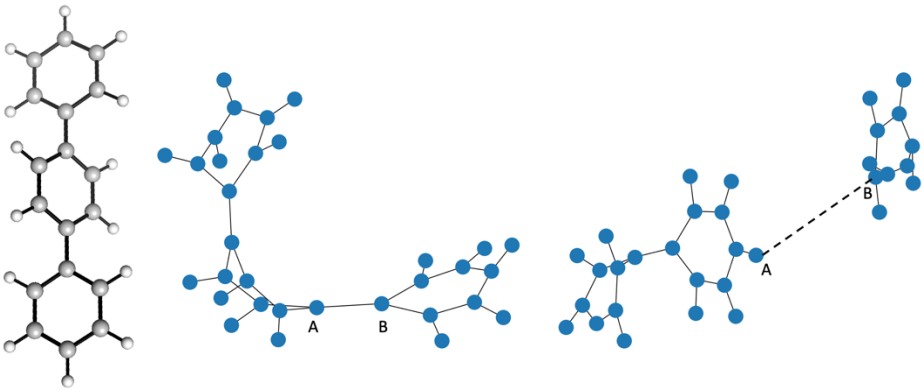


Figure 87: Graphical representation of a graph where the nodes represent the atoms, and the edges represent the bonds. By deleting the edge between atoms, A and B two subgraphs result. The subgraph which contains atom B is the subset of atoms we are interested in.

In short, the wSterimol code was modified to automatically identify the carbon and hydrogen atoms that are connected to the nitrogen atoms of the ureas, thioureas and squareamides, returning a list of atom ids for each catalyst. The script identifies which bond can be deleted based on the atom list given, resulting to two subgraphs of the same molecule. Finally, it calculates the B_1, B_5, and L values for the subgraph of interest. The bonds scanned to extract sterimol values are introduced in detail in each chapter separately.

Appendix C

C.1. Mordred descriptors

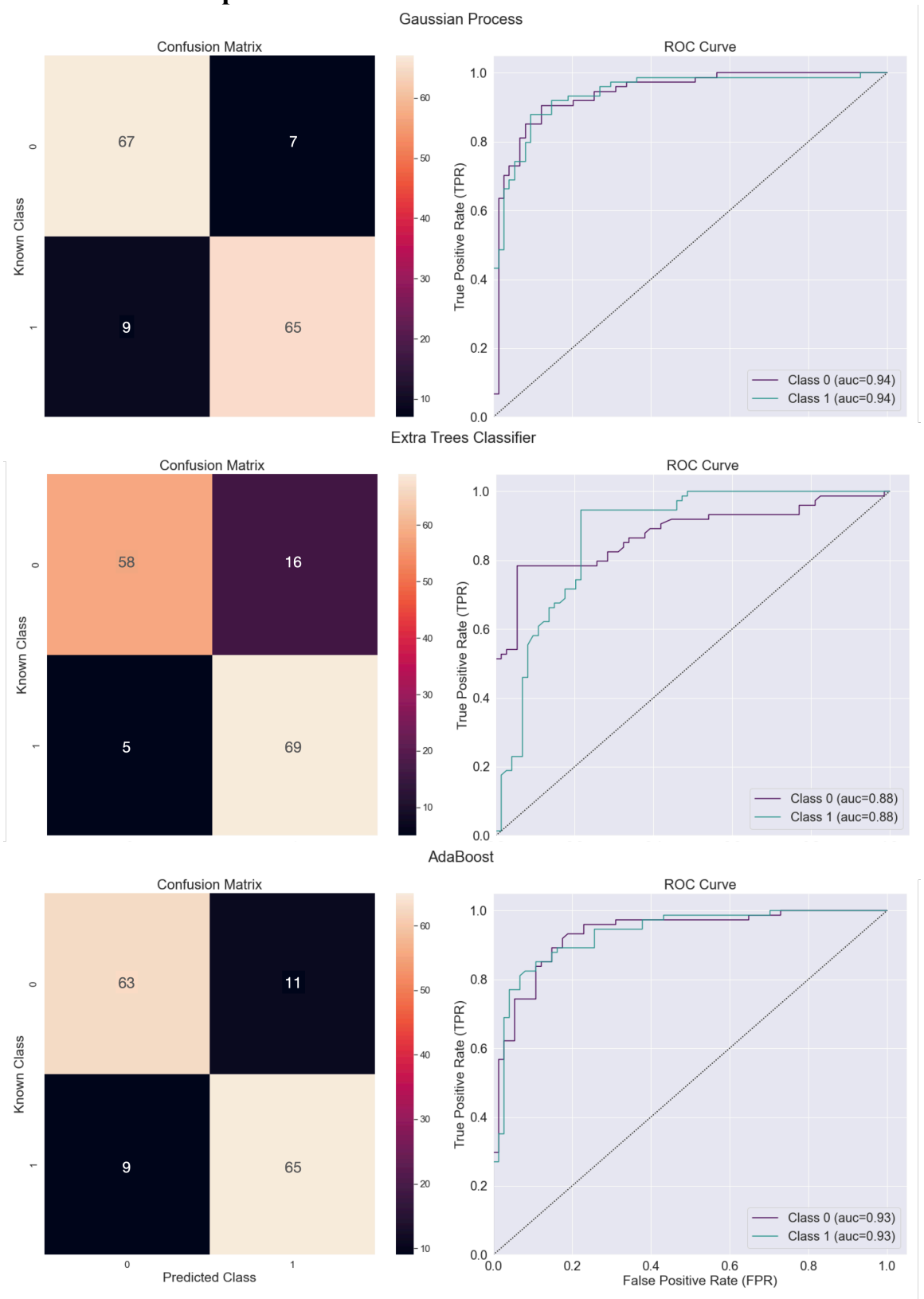


Figure 88: The confusion matrices and the ROC curves for the top three classifiers, with Mordred descriptors, on the training data.

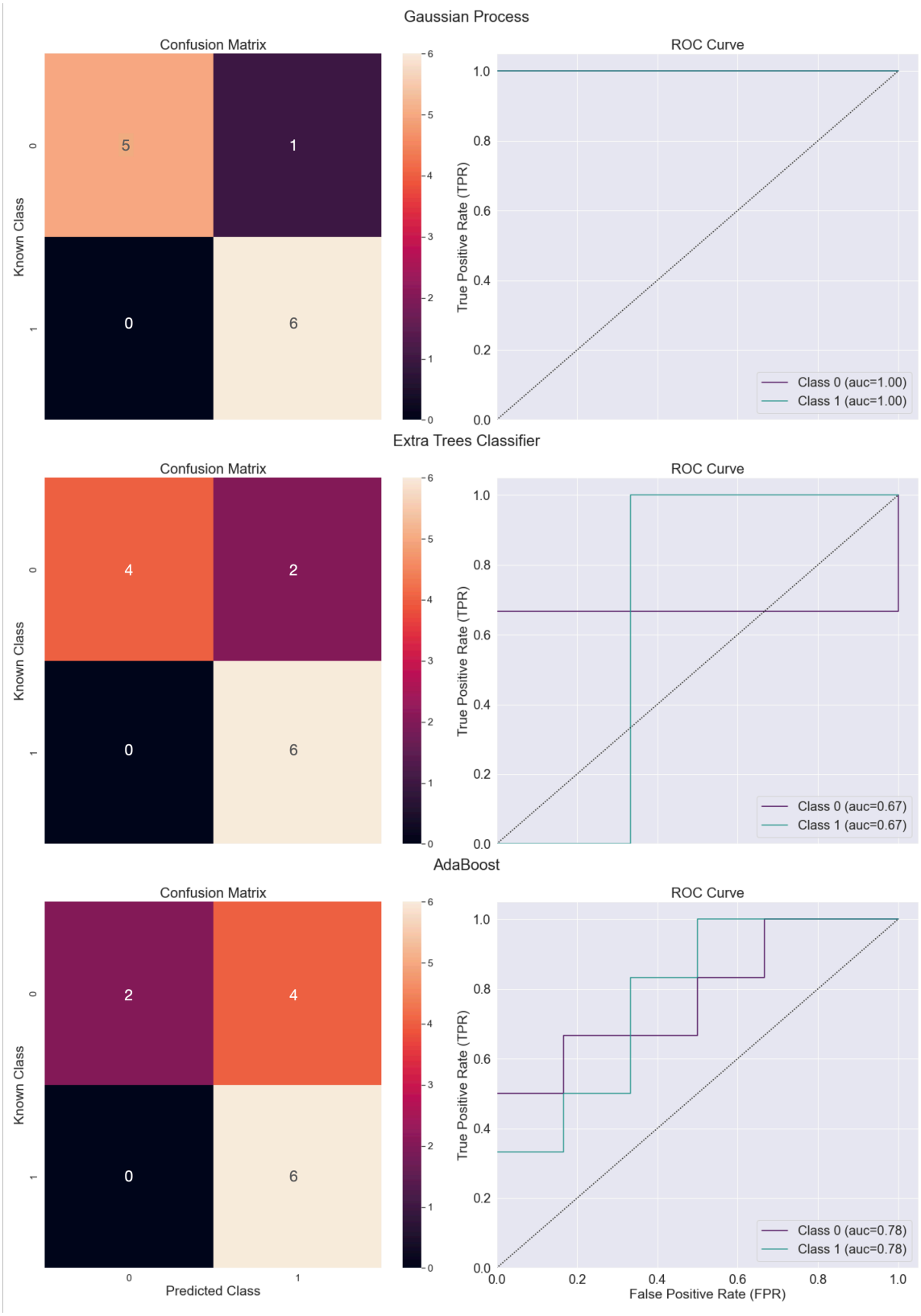


Figure 89: The confusion matrices and the ROC curves for the top three classifiers, with Mordred descriptors, on the test set. The ROC curve for the GP does not show the purple line (AUC class 0) as it overlaps with the green line (AUC class 1).

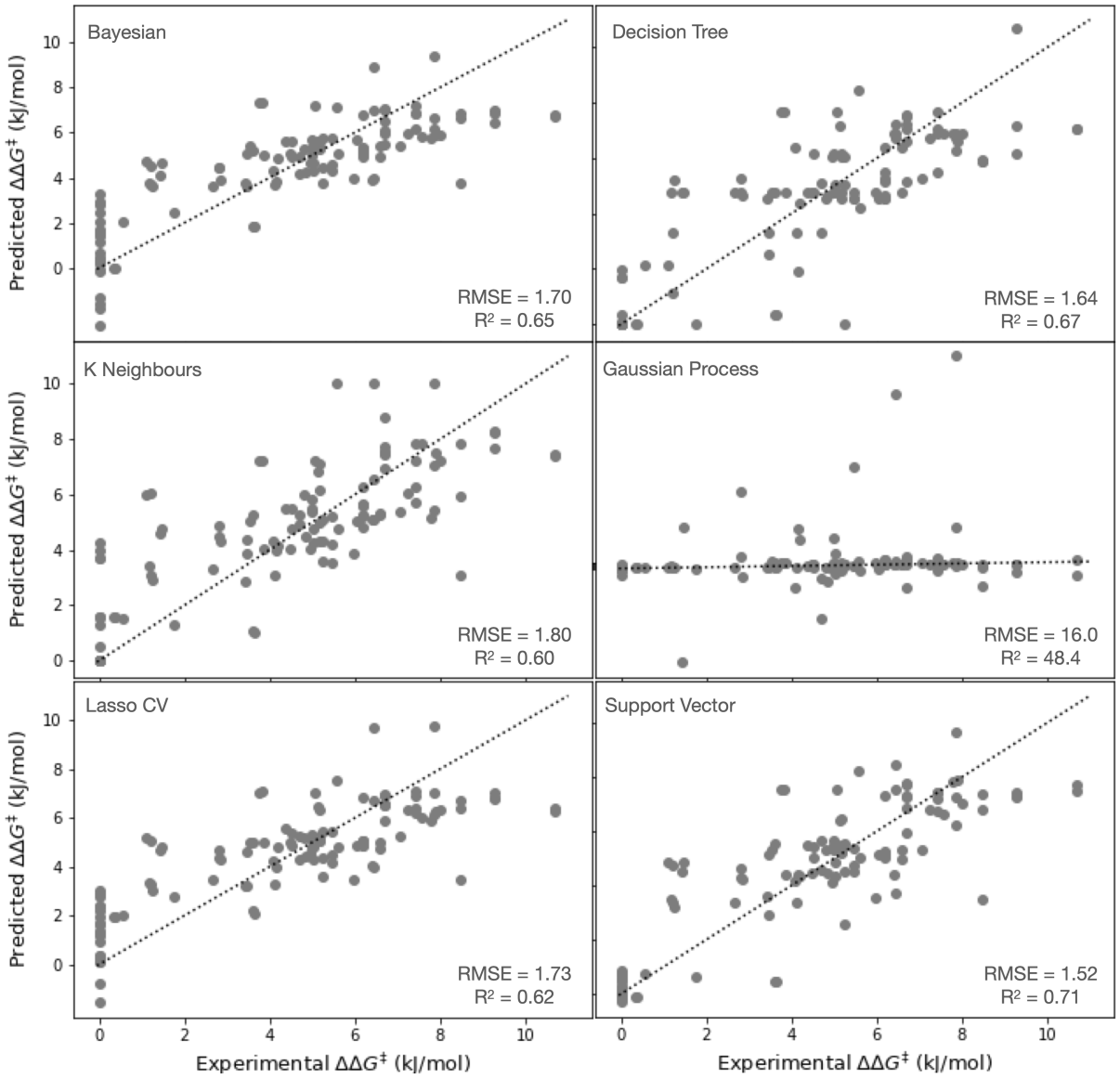


Figure 90: Regression models for the training set with Mordred descriptors.

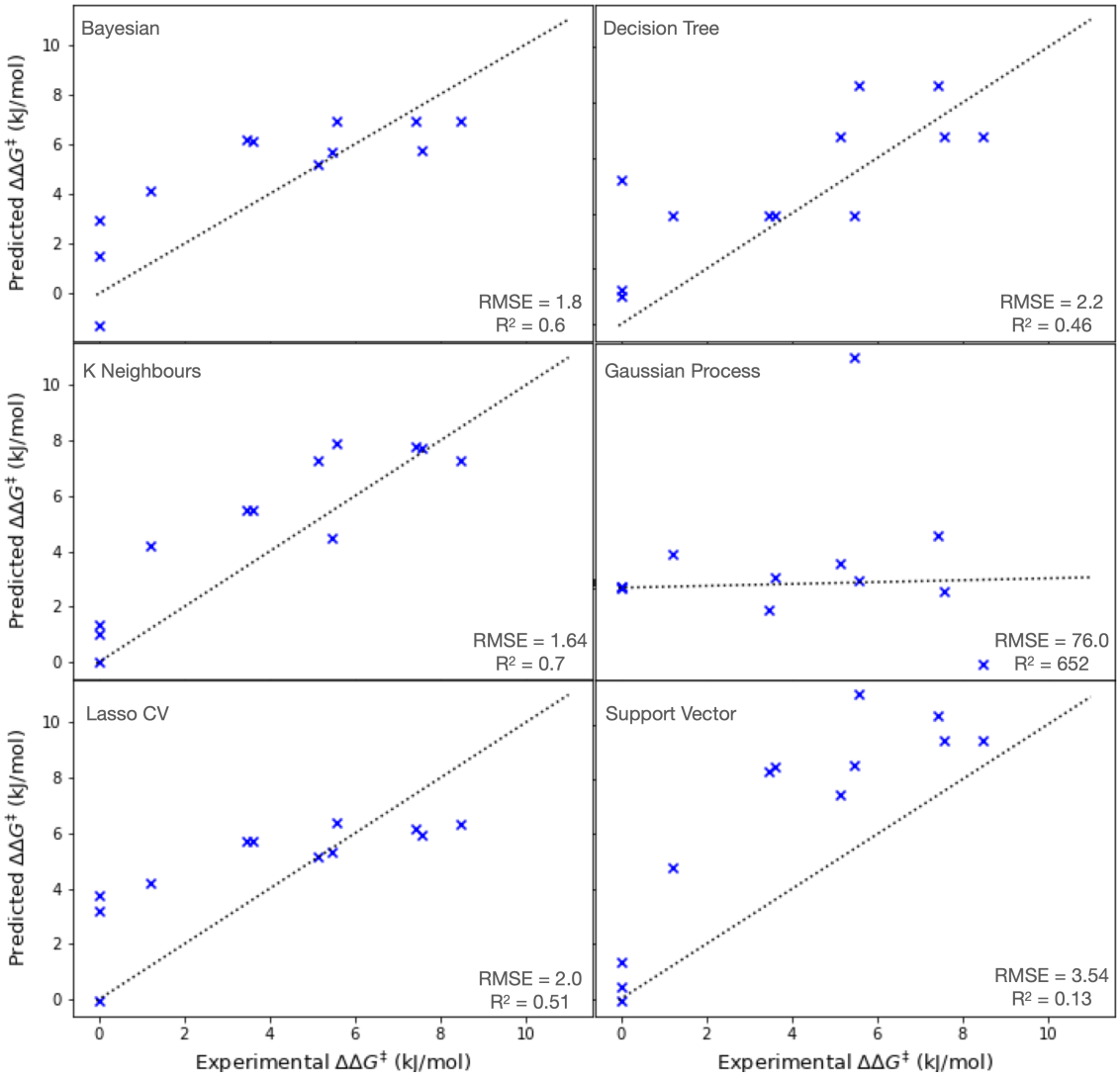


Figure 91: Regression models for the test set with Morgan descriptors.

C.2. Morgan fingerprints

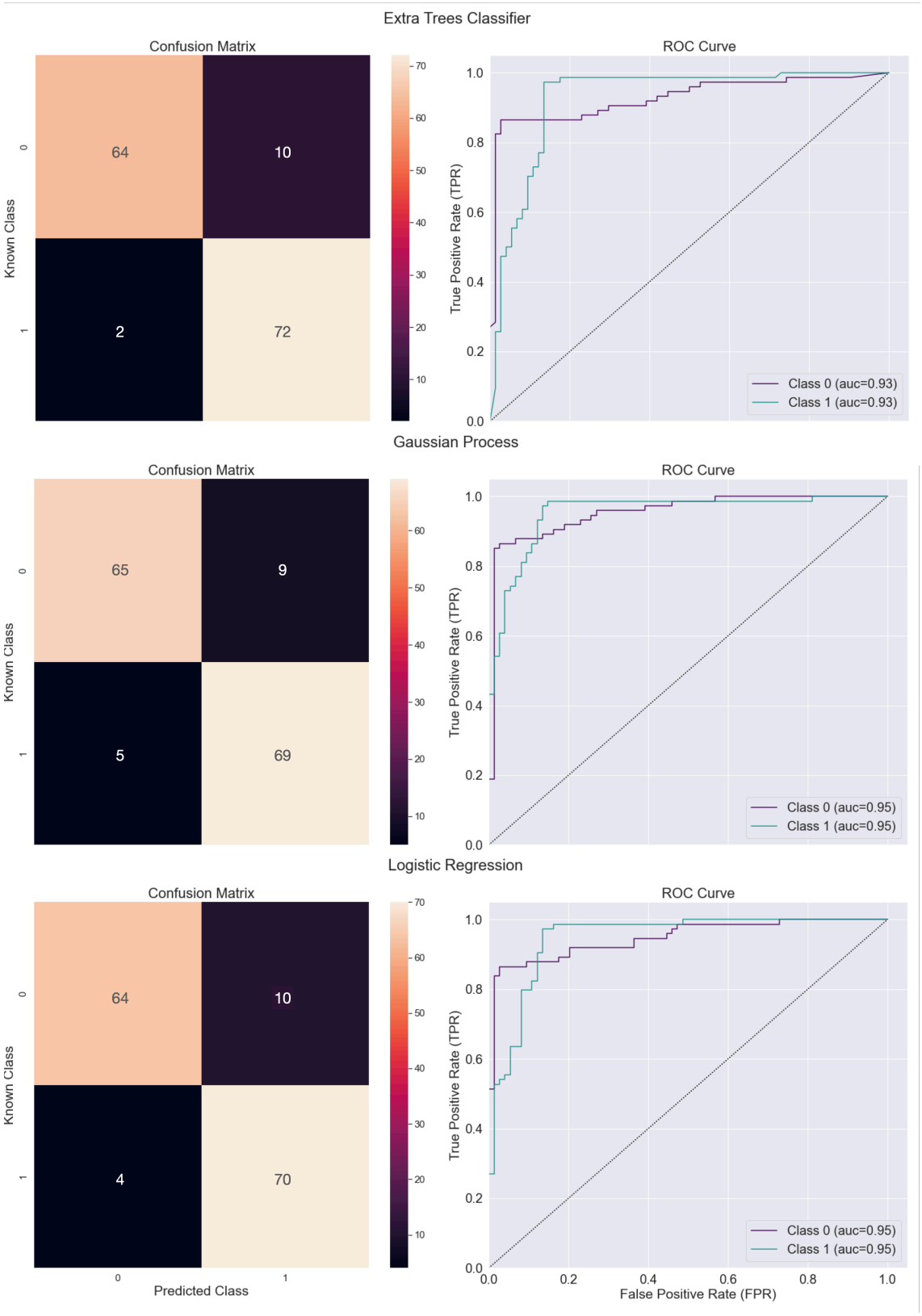


Figure 92: The confusion matrices and the ROC curves for the top three classifiers, with Morgan fingerprints, for the training set.

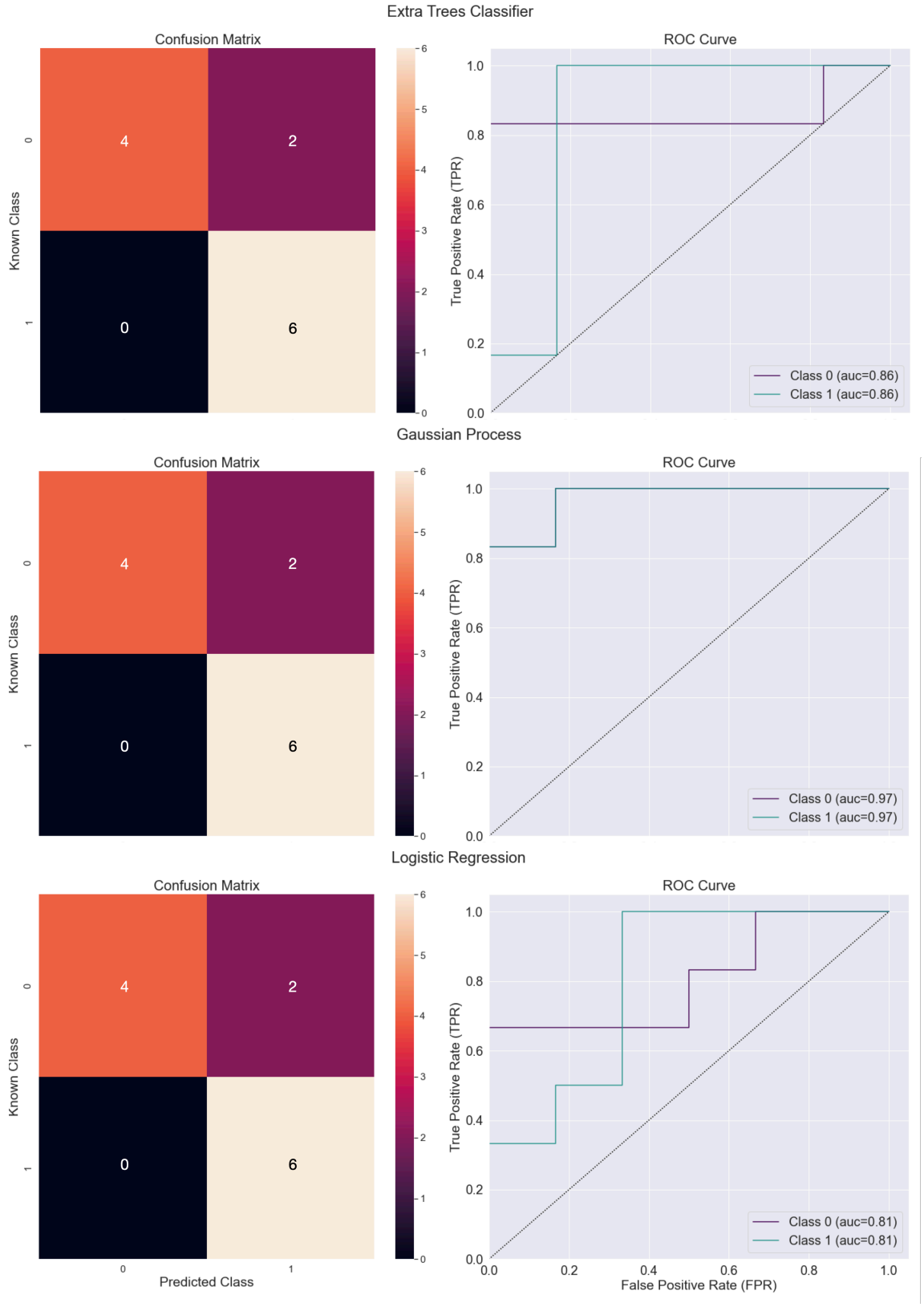


Figure 93: The confusion matrices and the ROC curves for the top three classifiers, with Morgan fingerprints, for the test set. The ROC curve for the GP does not show the purple line (AUC class 0) as it overlaps with the green line (AUC class 1).

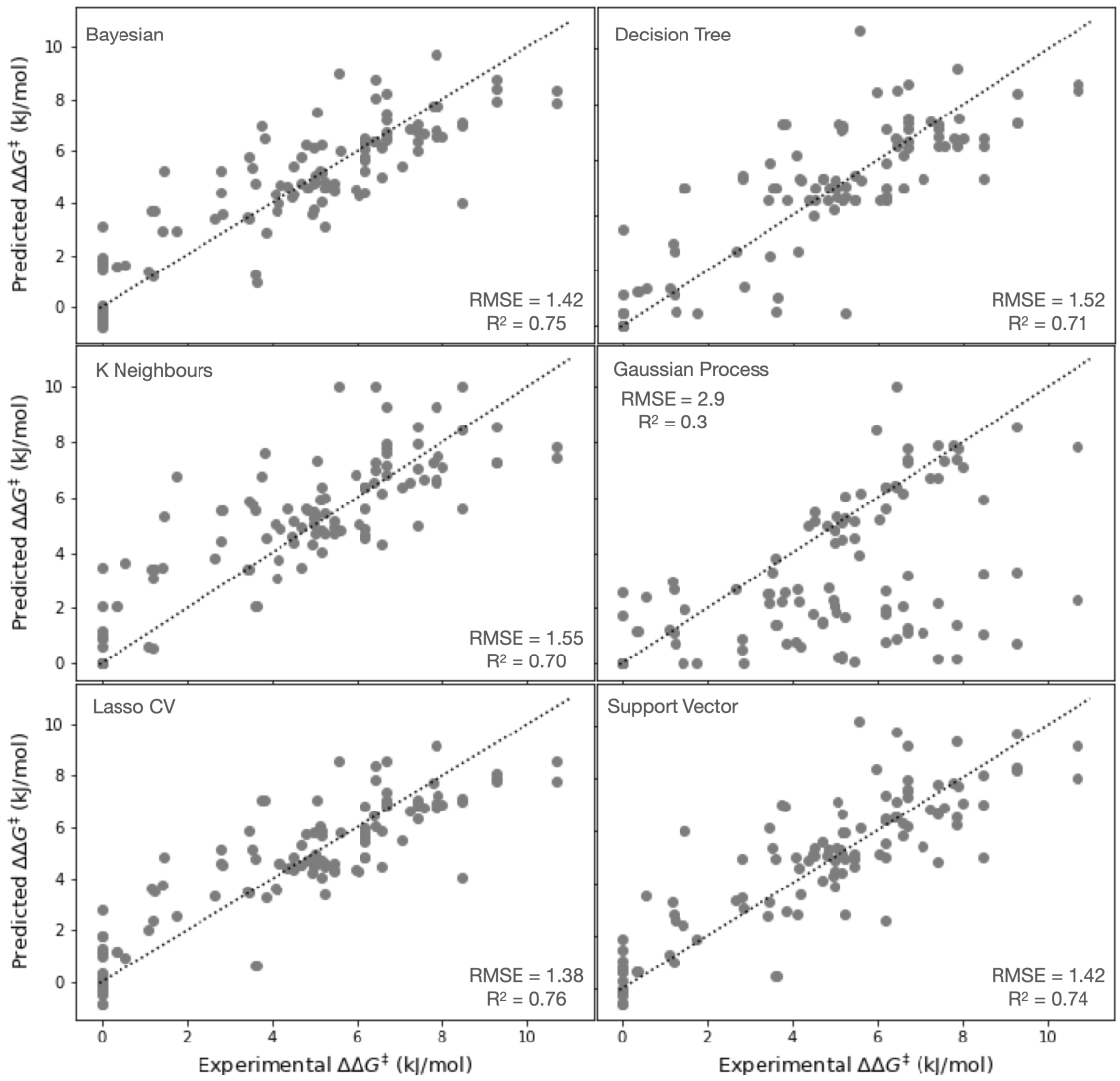


Figure 94: Regression models for the training set with fingerprints.

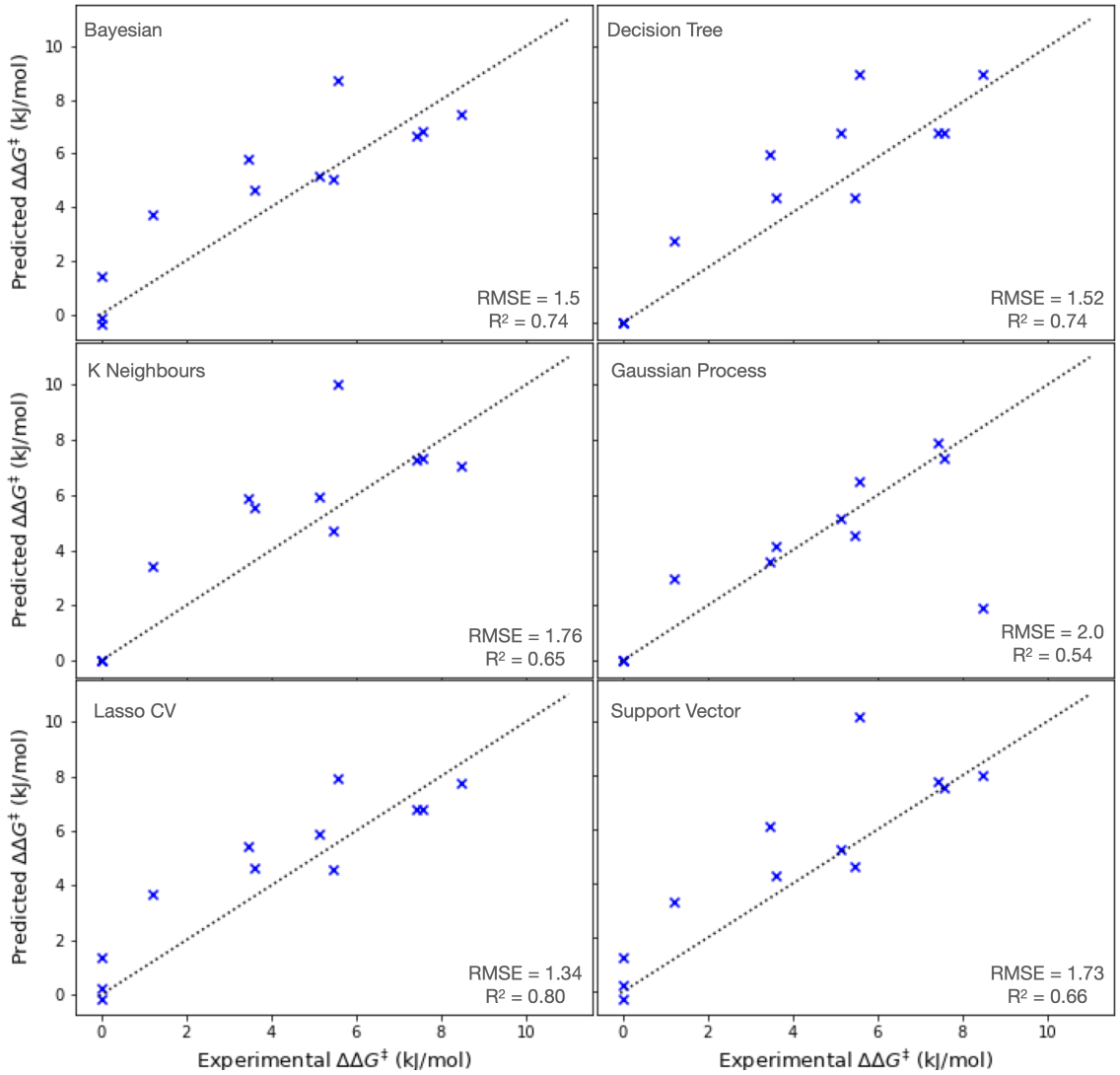


Figure 95: Regression models for the test set with fingerprints.

C.3. DFT descriptors

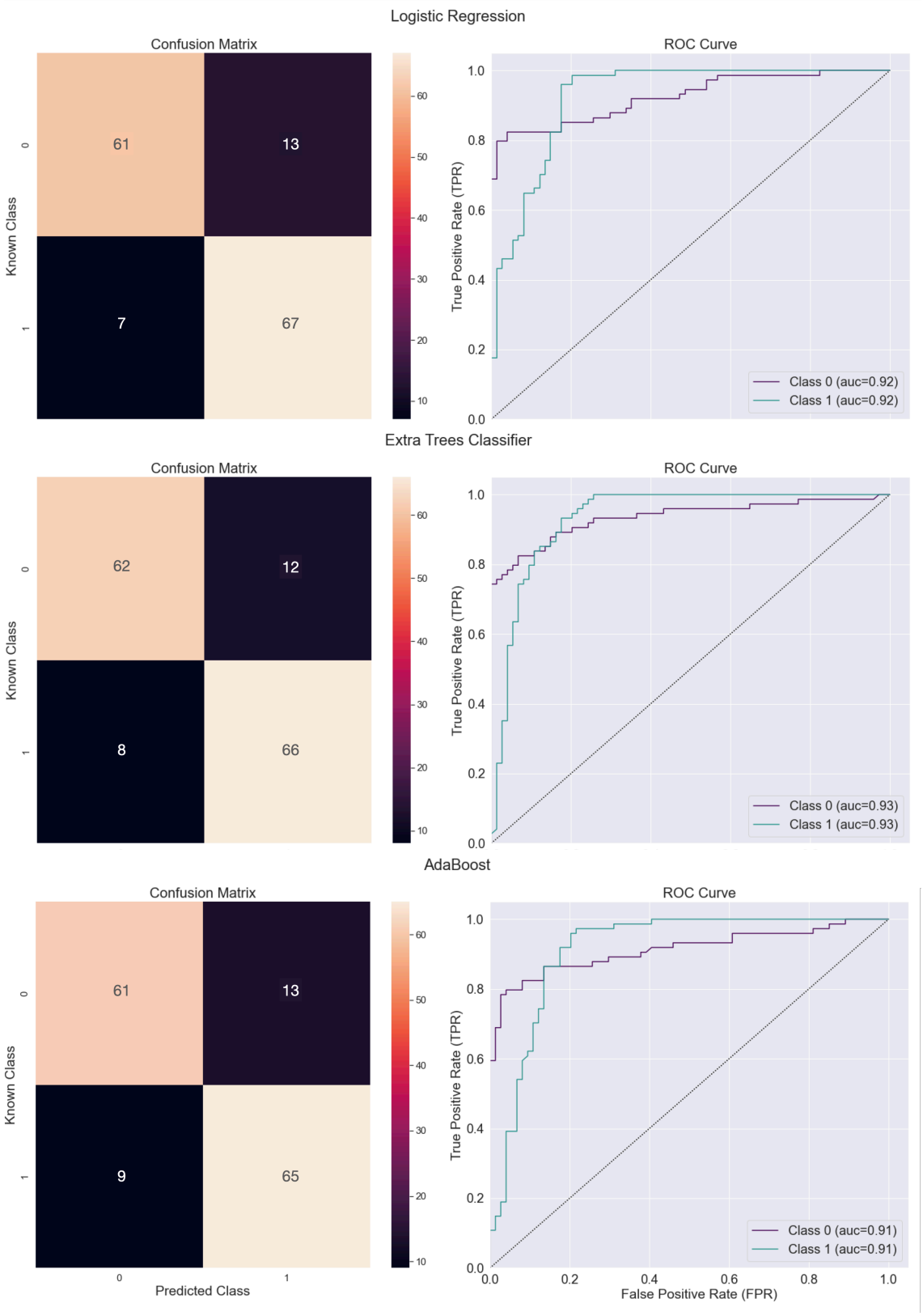


Figure 96: The confusion matrices and the ROC curves for the top three classifiers, with DFT descriptors for the training set.

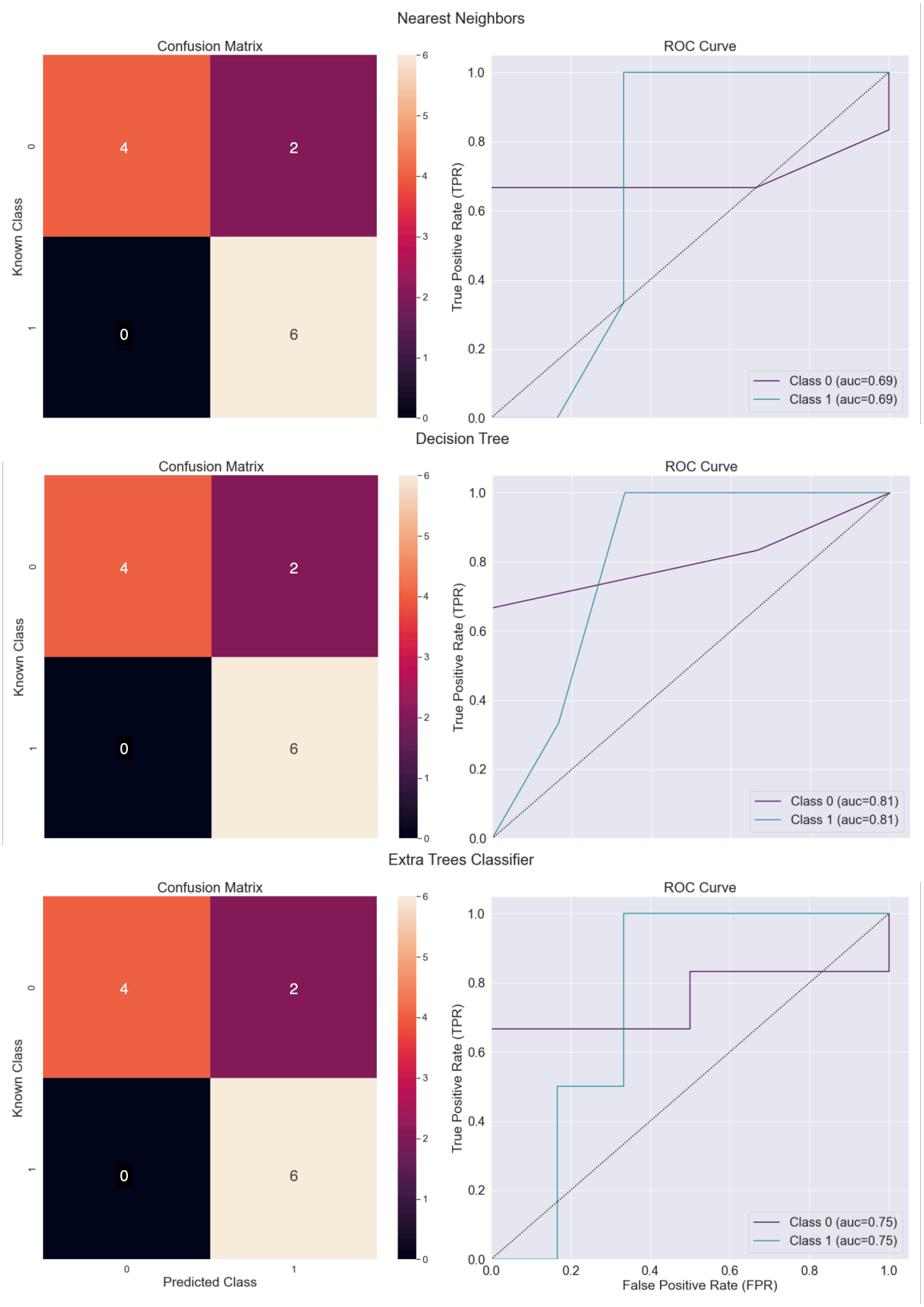


Figure 97: The confusion matrices and the ROC curves for the top three classifiers, with DFT descriptors, for the test set.

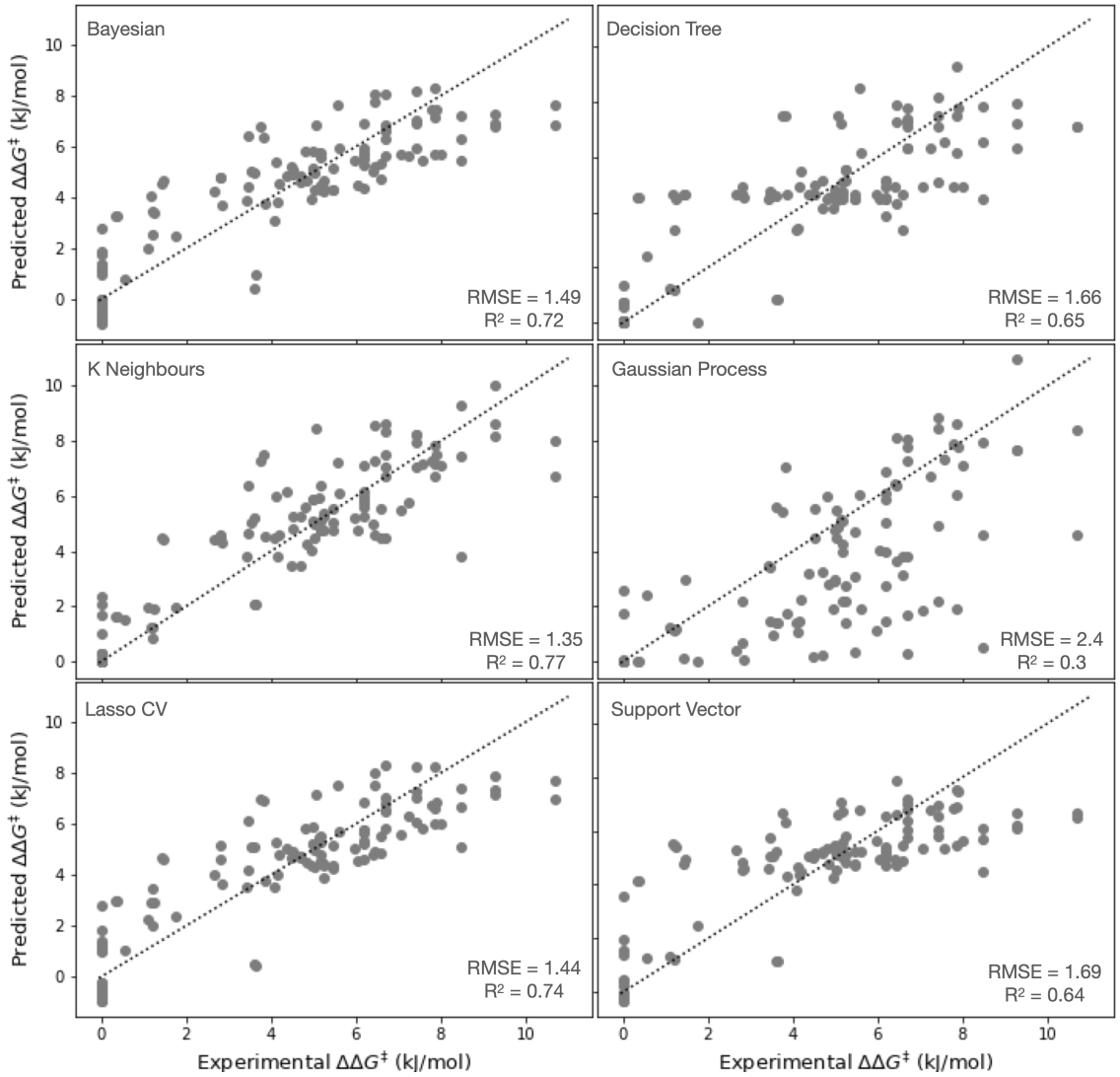


Figure 98: Regression models for the training set with DFT descriptors.

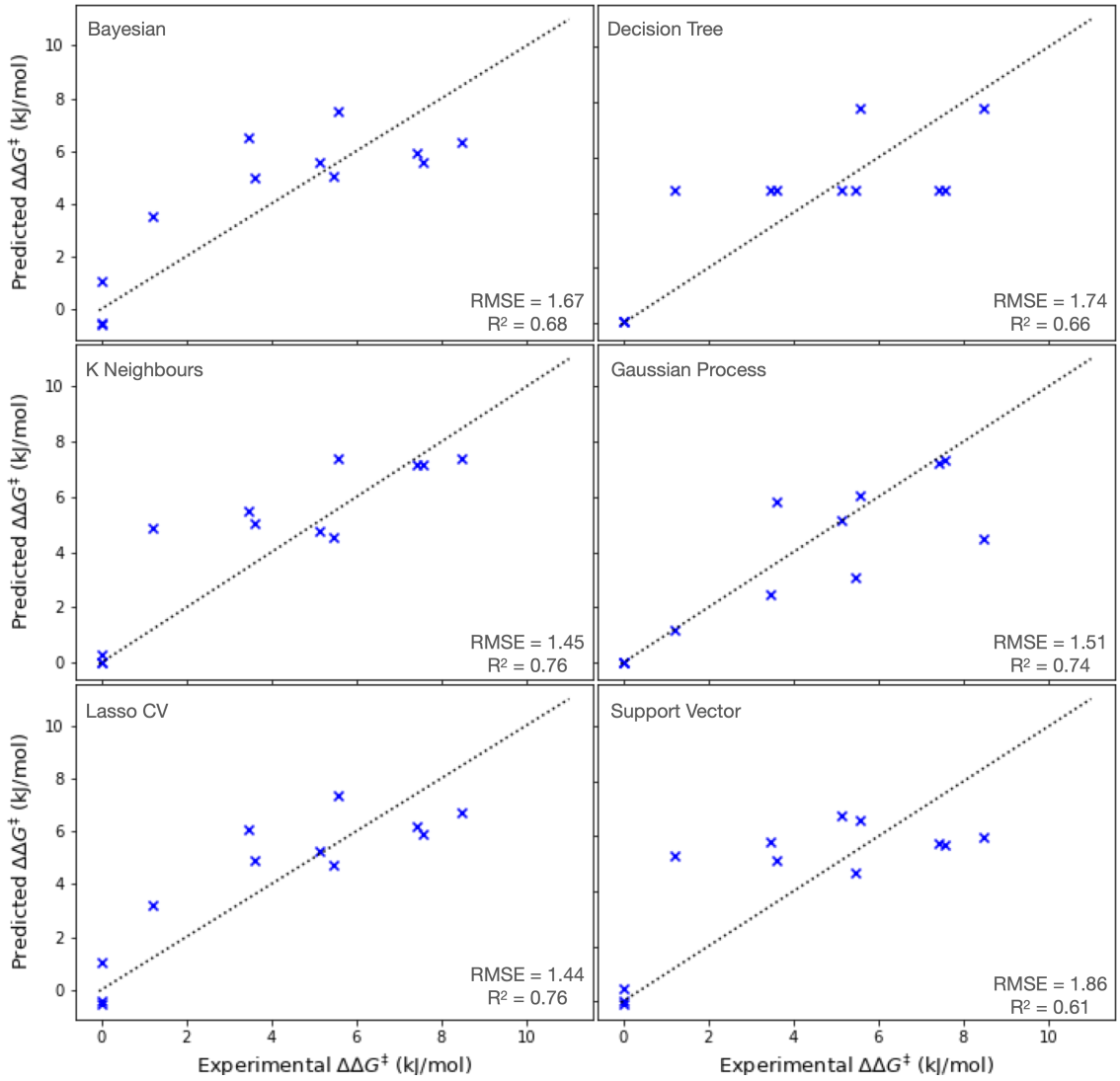


Figure 99: Regression models for the test set with DFT descriptors.

Appendix D

D.1. Morgan fingerprints

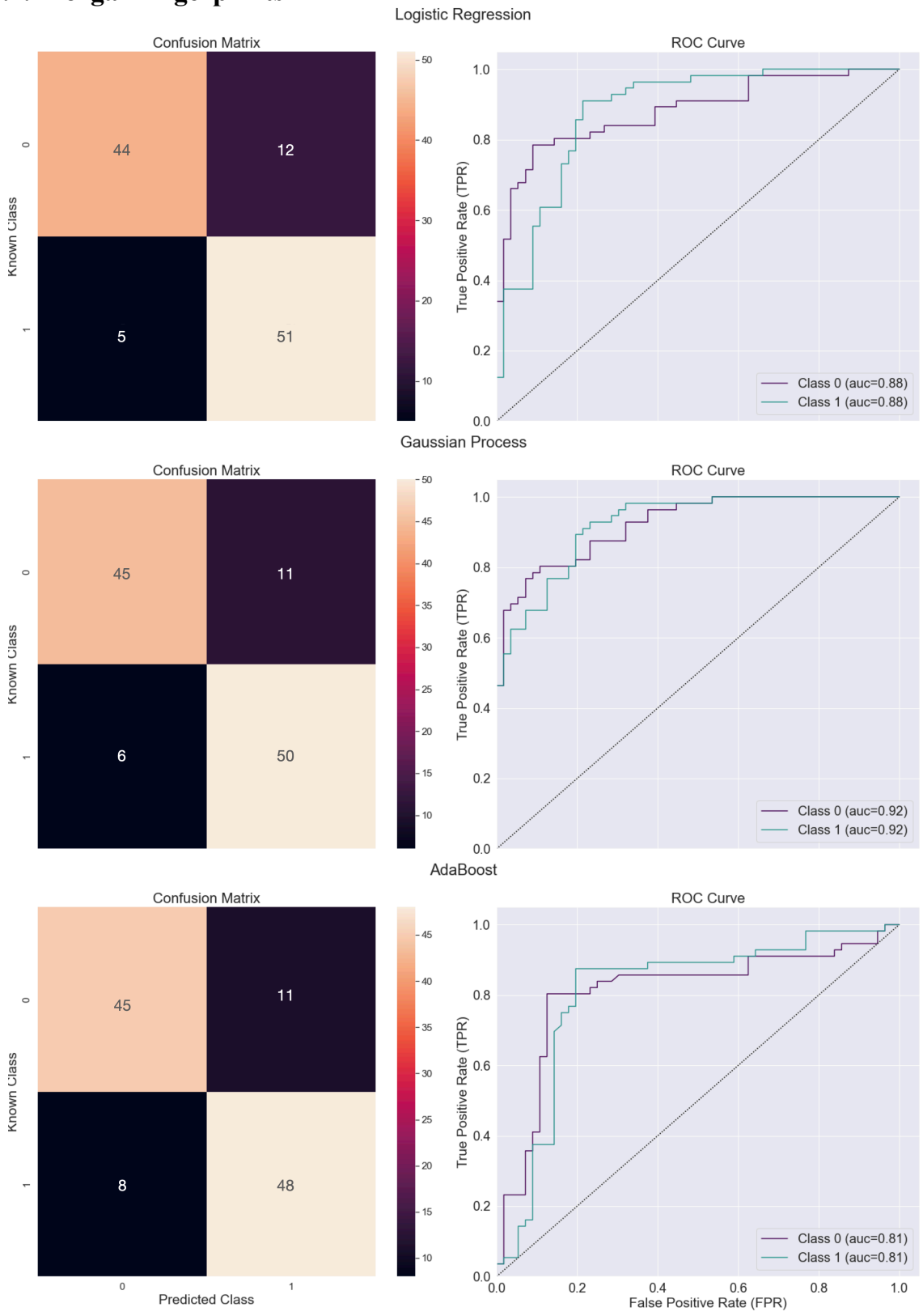


Figure 100: The confusion matrices and the ROC curves for the top three classifiers, with Morgan fingerprints, for the training set.

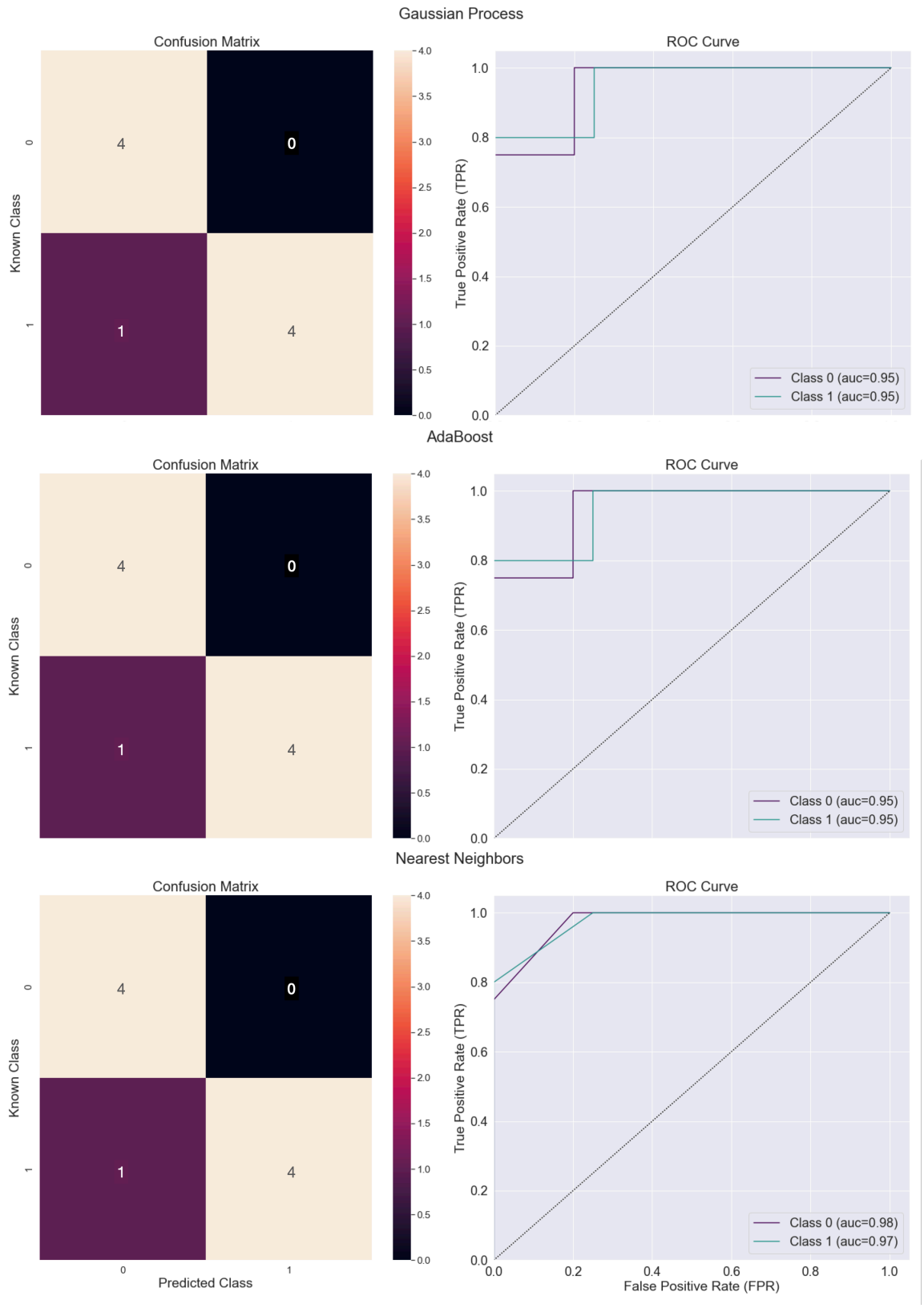


Figure 101: The confusion matrices and the ROC curves for the top three classifiers, with Morgan fingerprints, for the test set.

D.1. DFT descriptors

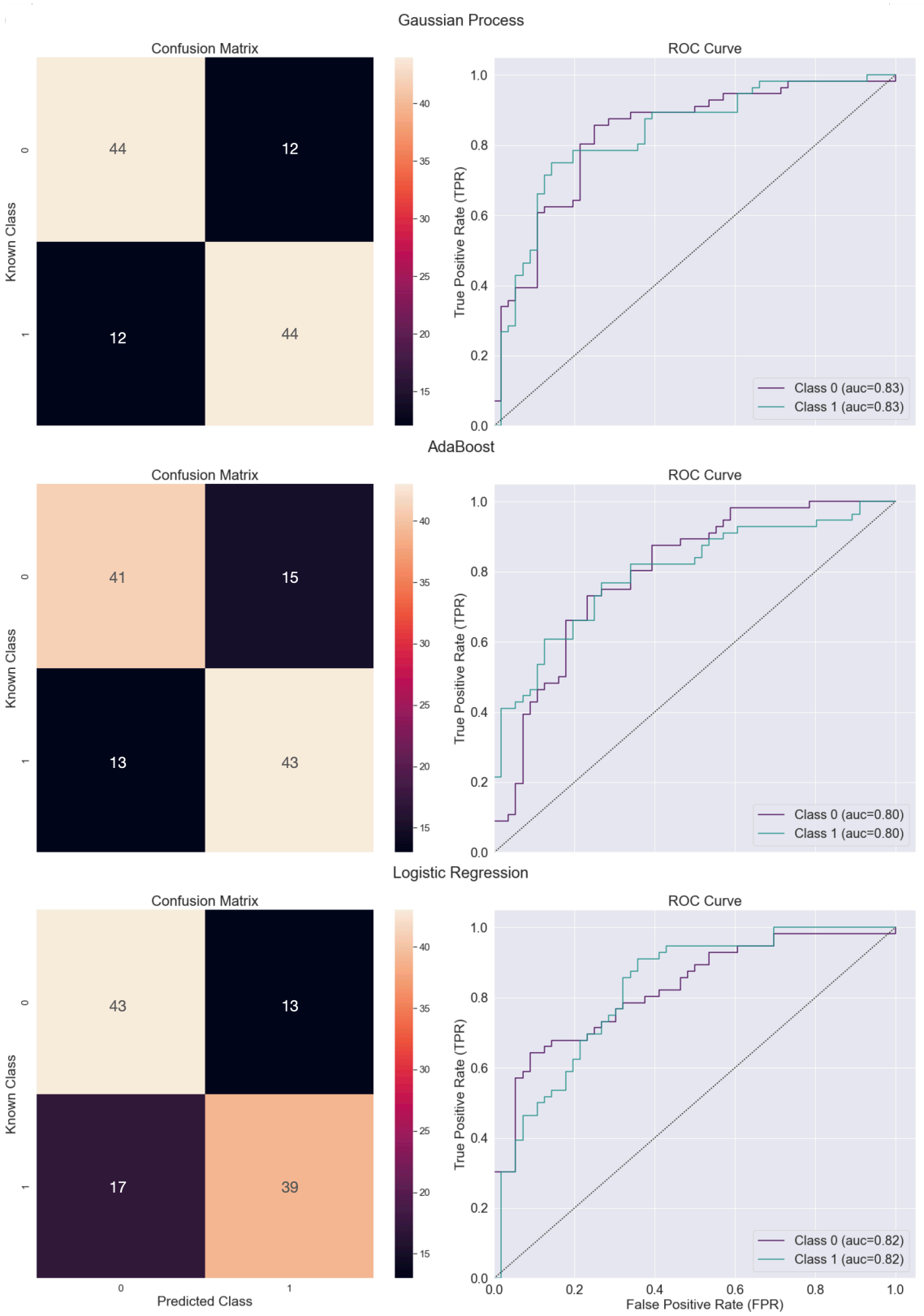


Figure 102: The confusion matrices and the ROC curves for the top three classifiers, with DFT descriptors, for the training set.

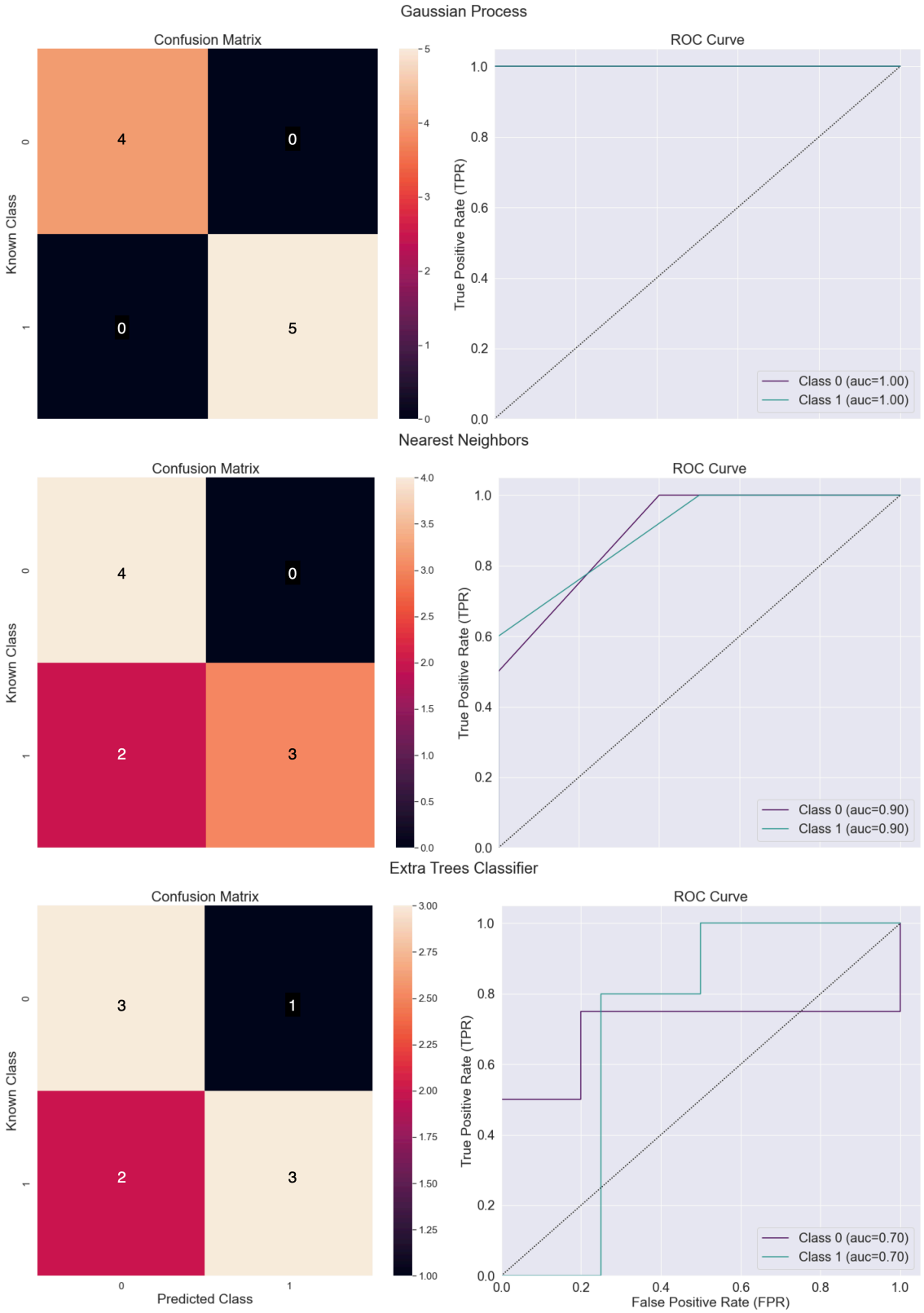


Figure 103: The confusion matrices and the ROC curves for the top three classifiers, with DFT descriptors, for the test set. The ROC curve for the GP does not show the purple line (AUC class 0) as it overlaps with the green line (AUC class 1).

Appendix E

E.1. GNN with one graph

An alternative way to build the GNN is to generate one graph that represents both the anion and the cation. This practically means that the SMILES strings for the two molecules are concatenated in one SMILES string. RDKit[5] features are extracted in the same manner to create the feature vectors of the nodes and edges in the IL. The GNN architecture in this case is shown in Figure 104. For the graph convolutional a GINEConv was employed in two layers in combination with a GRU and fingerprint dimension of 64. The structure of the MLP_{TP} is not varied and set to three layers with 66 (two additional dimension for the temperature and pressure), 32, and 1 neurons. The following training hyperparameters are applied: initial learning rate 0.001, learning rate decay of 0.8 with a patience of 3 epochs, batch size 64, maximum number of epochs 100 for viscosity and 120 for solubility, optimizer adam, early stopping patience of 25 epochs, dropout rate in the MLP_{TP} of 0.05. Training of this model was faster than the two graphs model (7 minutes and 24 minutes respectively in CPU).

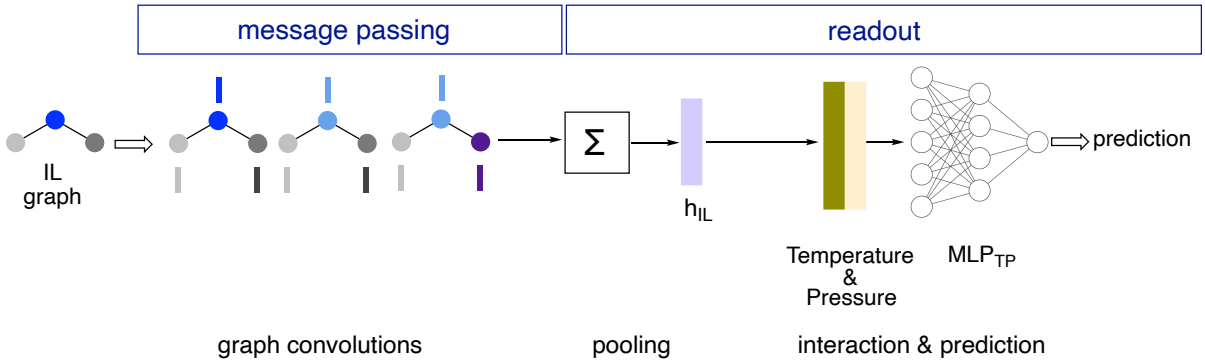


Figure 104: Architecture of GNN model, when one graph is considered both for the anion and the cation.

This model showed to be less accurate compared to the one suggested in our main Thesis for the viscosity dataset, especially for the *Extrapolation* model (Tables 27 & 28). We assume that this architecture does not allow the model to learn the interactions between the anions and the cations.

Table 27: Average metrics for Generalization results for viscosity for 40 models.

Data set	*MAE*	*RMSE*	R^2
Train	0.34	0.52	0.89
Validation	0.34	0.53	0.89
Test	0.35	0.54	0.88

Table 28: Extrapolation results for viscosity.

Data set	*MAE*	*RMSE*	R^2
Train	0.38	0.57	0.87
Validation	0.64	1.10	0.66
Test	0.90	1.19	0.36

E.2. Regression model with Morgan fingerprints

Before we attempted building a GNN for the prediction of viscosity and solubility of ILs we wanted to ensure that a simple regression model with fingerprints could not perform adequately. We therefore created a model on the viscosity data, that uses Morgan fingerprints for each anion and cation and LASSOCV. Information about the temperature was also included. The dataset was split randomly to 80% train set and 20% test set and yield indeed adequate results but no better than other published literature has achieved. For the test set the correlation was relatively high, however the errors were significantly higher compared to the GNN model ($R^2 = 0.83$, MAE = 0.47, RMSE = 0.64). This model was not considered further.

Appendix bibliography

(1) Pfeifer, L.; Engle, K. M.; Pidgeon, G. W.; Sparkes, H. A.; Thompson, A. L.; Brown, J. M.; Gouverneur, V. Hydrogen-Bonded Homoleptic Fluoride-Diarylurea Complexes: Structure, Reactivity, and Coordinating Power. *J. Am. Chem. Soc.* **2016**, *138* (40), 13314–13325.

(2) Engle, K. M.; Pfeifer, L.; Pidgeon, G. W.; Giuffredi, G. T.; Thompson, A. L.; Paton, R. S.; Brown, J. M.; Gouverneur, V. Coordination Diversity in Hydrogen-Bonded Homoleptic Fluoride-Alcohol Complexes Modulates Reactivity. *Chem. Sci.* **2015**, *6* (9), 5293–5302.

(3) Brethomé, A. V.; Fletcher, S. P.; Paton, R. S. Conformational Effects on Physical-Organic Descriptors: The Case of Sterimol Steric Parameters. *ACS Catal.* **2019**, *9* (3), 2313–2323.

(4) Harper, K. C.; Bess, E. N.; Sigman, M. S. Multidimensional Steric Parameters in the Analysis of Asymmetric Catalytic Reactions. *Nat. Chem.* **2012**, *4* (5), 366–374.

(5) *RDKit*. RDKit: Open-source cheminformatics; http://www.rdkit.org. https://www.rdkit.org/ (accessed 2021-04-01).