KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Uncertainty Quantification for Trustworthy Drug-Target Interaction Predictions

Hannah Rosa Friesacher

Dissertation presented in partial fulfillment
of the requirements for the degree of
Doctor of Engineering Science (PhD):
Electrical Engineering

December 2025

Supervisors:
Dr. Adam Arany
Prof. dr. ir. Yves Moreau
Prof. dr. Ola Engkvist
(AstraZeneca, Sweden &
Chalmers University of Technology, Sweden)

UNCERTAINTY QUANTIFICATION FOR TRUSTWORTHY DRUG-TARGET INTERACTION PREDICTIONS

HANNAH ROSA FRIESACHER

Supervisors:
Dr. Adam Arany
Prof. dr. ir. Yves Moreau
Prof. dr. Ola Engkvist
(AstraZeneca, Sweden &
Chalmers University of Technology, Sweden)

Members of the
Examination Committee:
Prof. dr. ir. Krista De Jonge, chair
Prof. dr. Matthew Blaschko
Dr. Bence Bruncsics
Prof. dr. Günter Klambauer
(Johannes Kepler University Linz, Austria)
Prof. dr. Esther Heid
(TU Wien, Austria)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

December 2025

© 2025 Hannah Rosa Friesacher
Uitgegeven in eigen beheer, Hannah Rosa Friesacher, Your place of residence (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Preface

Looking back on my PhD journey, I am deeply grateful for the support I have received over the past four years. This thesis is, in many ways, a shared accomplishment that would not have been possible without the knowledge, encouragement, and support of those around me. I would like to say thank you to everyone who stood by me during this experience.

To my supervisory and examination committee. First, I would like to thank the *members of my supervisory committee, Prof. Klambauer* and *Prof. Blaschko*, for their valuable insights and guidance throughout the last four years. I also want to thank the *entire examination committee* for reviewing this thesis and providing constructive feedback. My gratitude also goes to *Prof. De Jonge* for kindly serving as chair of the committee.

To my supervisors. *Dr. Adam Arany* – thank you for your invaluable support throughout my PhD. Your endless knowledge and ideas were crucial to the success of my work, and I am sincerely grateful for your patient and trusting guidance over the past years. And thank you so much for all the exciting and wonderful trips to the far (and very snowy;)) corners of Europe! *Prof. Yves Moreau* – thank you for your thoughtful questions and insightful comments, which always encouraged me to think more critically. I am truly grateful for your support over the last years! *Prof. Ola Engkvist* – it was a pleasure to have you as my supervisor. Thank you for the warm welcome in Sweden, which made my stay at AstraZeneca so enjoyable. Thank you as well for always making time for our regular meetings; your experience and expertise were invaluable to my work!

To my colleagues. *Emma* – a big heartfelt thank you to you. I really enjoyed collaborating with you, as well as our regular after-work bouldering sessions. *To the fellow PhD students of the AIDD program* – thank you for making all the PhD schools such unforgettable experiences. A special shout-out to *Peter, Mathias, Vincenzo, and Varvara* for making my time at AstraZeneca so

enjoyable. *To my colleagues at KU Leuven* – although I only spent a part of my PhD in Belgium, I have many great memories of our time together. Thank you for all the great moments in the office and during after-work drinks! *Andras* – having someone in the same PhD-boat, going through the same ups and downs of this journey, meant a lot. It was a pleasure to share this experience with you! *Ashkan* – your kindness and constant willingness to help made me feel so welcome in Leuven and in the group. Thank you for supporting me with so many of the organizational challenges of this thesis!

To my friends. *Markus* – you have been a constant in my life for more than a decade, and I want to thank you for all the funny, beautiful, and insightful moments we have shared. I really enjoyed having you over in Belgium! Our bouldering and concert sessions made Belgium feel like home. *Jessi, Hussein, and Ariane* – thank you for your friendship, for always staying connected despite the distance, and for joining me in all my slightly crazy ideas, from collective trips to Belgium to long-distance book clubs. *To all my friends in Auftakt* – thank you for the incredible concerts and musical moments that balanced out the stressful moments. Your support over the years has meant so much to me, and you have always made me feel as if I never left Vienna.

To my family. *To my parents, Elke and Tom* – thank you for always being incredibly supportive and for encouraging me to move out of my comfort zone. Without you, this journey would not have been possible. Also, thanks for never losing interest in what I do, even if it sounds like gibberish to you ;). *Michi* – I am so happy that you have become part of our family. Also, our trips to Pfarrwerfen were always a highlight of the year, so thank you for the great memories! *Theres* – thank you for all the phone calls and talks over the past four years, encouraging me to grow as a person and pursue my PhD with confidence. Your moral support and constant encouragement helped me to get through the difficult moments (and made the good ones even better!). *To my partner* – Andi, having you by my side, with your humor and endless optimism, made this journey so much easier. Thank you for always being there for me with unwavering support, regardless of how far apart we were scattered across Europe.

Abstract

The development of new therapeutic agents is a time- and resource-consuming process, characterized by high failure rates and development spans of over a decade until a compound can be put on the market. The use of machine learning (ML) approaches can contribute to easing these problems by using the extensive amount of data produced in the drug discovery pipeline to train computational models that can effectively support future projects with their knowledge. Quantitative structure-activity relationship (QSAR) models are well-established in computer-aided drug discovery for identifying compounds with desired features. They enable the prediction of biological activities or properties of chemical compounds based on their molecular structure.

Due to the time-consuming and expensive nature of the drug discovery workflow, the reliability of machine learning predictions is essential. In this framework, uncertainty quantification (UQ) has emerged as a powerful tool that allows model users to assess the risk associated with a model prediction before deciding how to progress in the experimental pipeline. To optimally support an informed decision-making process, it is essential to accurately quantify the uncertainty in machine learning predictions.

Previous work has concluded that ML models, especially modern neural networks, often fail to provide realistic estimates of the uncertainty associated with predictions in classification tasks, leading to poorly calibrated models. Since uncertainty estimates are only meaningful when they accurately reflect the true uncertainty of a prediction, this work focuses on achieving well-calibrated uncertainty estimation. The primary objective of this work is to identify practices that yield reliable uncertainty estimates for drug–target interaction predictions. Using both real-world and public datasets, we evaluate the performance of various uncertainty quantification approaches and highlight challenges that arise in realistic settings. This thesis is structured into two parts, each based on distinct datasets and addressing different research questions.

In the first part (Chapter 3), we use a real-world pharmaceutical dataset to conduct a comprehensive, large-scale evaluation of uncertainty estimation methods, including ensemble-based and Bayesian approaches. Furthermore, we use this realistic setting to systematically assess the distribution shifts in label and descriptor space and their impact on the capability of the uncertainty estimation methods. Finally, we adapt uncertainty quantification regression models to enable learning from censored labels, which are commonly available in pharmaceutical datasets but can rarely be used by conventional machine learning methods.

The second part of this work (Chapter 4) focuses on using Hamiltonian Monte Carlo (HMC) sampling for UQ in QSAR models. Although HMC is regarded as the gold standard for posterior approximation, its high computational cost makes it impractical for most real-world applications. In the first part of this work, we adopt a low-complexity approach that enables HMC-based uncertainty quantification by sampling only parts of the machine learning model. In the second part, we aim to develop a ground truth for uncertainty quantification in drug-target interaction predictions by applying HMC sampling across entire QSAR models. An additional focus is on identifying strategies that yield well-behaved HMC chains that can effectively sample from the posterior.

By applying various UQ methods across different datasets, this work provides a comprehensive study of uncertainty estimation in drug-target interaction predictions. The findings provide valuable insights into how reliable uncertainty estimates can be achieved, ultimately supporting well-informed decision-making and contributing to a more resource-and time-efficient drug discovery pipeline. Furthermore, they highlight the challenges introduced by real-world data, emphasizing the complexity of identifying effective strategies for uncertainty estimation in QSAR models.

Beknopte samenvatting

De ontwikkeling van nieuwe therapeutische middelen is een tijdrovend en kostbaar proces, dat wordt gekenmerkt door hoge mislukkingspercentages en ontwikkelingsperiodes van meer dan tien jaar voordat een geneesmiddel op de markt kan worden gebracht. Het gebruik van machine learning-benaderingen kan deze problemen helpen verlichten door de enorme hoeveelheid gegevens die tijdens het ontwikkelingsproces van geneesmiddelen wordt geproduceerd, te gebruiken om computationele modellen te trainen die met hun deskundige kennis toekomstige projecten effectief kunnen ondersteunen. Quntitative structure-activity relationship (QSAR) zijn een gevestigde waarde in computerondersteunde geneesmiddelenontwikkeling voor het identificeren van verbindingen met gewenste eigenschappen. Ze maken het mogelijk om biologische activiteiten of eigenschappen van chemische verbindingen te voorspellen op basis van hun moleculaire structuur.

Vanwege het tijdrovende en dure karakter van het proces voor de ontwikkeling van geneesmiddelen is de betrouwbaarheid van voorspellingen op basis van machine learning van essentieel belang. In deze context is onzekerheidsquantificering naar voren gekomen als een krachtig hulpmiddel waarmee gebruikers het risico van een voorspelling kunnen beoordelen voordat ze beslissen hoe ze verder gaan in de experimentele pijplijn. Om een weloverwogen besluitvormingsproces optimaal te ondersteunen, is het essentieel om de onzekerheid in voorspellingen op basis van machine learning nauwkeurig te kwantificeren.

Eerder onderzoek heeft aangetoond dat moderne neurale netwerken vaak geen realistische schattingen geven van de onzekerheid die gepaard gaat met een voorspelling in classificatietaken, wat resulteert in slecht gekalibreerde modellen. Aangezien onzekerheidsschattingen alleen zinvol zijn als ze de werkelijke onzekerheid van een voorspelling nauwkeurig weergeven, richt dit onderzoek zich op het ontwikkelen van een goed gekalibreerde onzekerheidsschattingsmethode. Het primaire doel van dit onderzoek is het identificeren van praktijken die betrouwbare onzekerheidsschattingen opleveren voor voorspellingen van

interacties tussen geneesmiddelen en doelwitten. Aan de hand van zowel bedrijfsinterne als openbaredatasets evalueren we de prestaties van verschillende benaderingen voor het kwantificeren van onzekerheid en belichten we uitdagingen die zich in realistische situaties voordoen. Dit proefschrift bestaat uit twee delen, die elk zijn gebaseerd op verschillende datasets en verschillende onderzoeksvragen behandelen.

In het eerste deel (hoofdstuk 3) gebruiken we een echte farmaceutische dataset om een uitgebreide, grootschalige evaluatie uit te voeren van methoden voor onzekerheidsschatting onder realistische temporele distributieverschuivingen. We onderzoeken de prestaties van verschillende methoden voor onzekerheidsschatting, waaronder ensemble-gebaseerde en Bayesiaanse benaderingen. Bovendien gebruiken we deze industriële context om systematischde verschuivingen van de distributie in label- en descriptorruimte en hun impact op de capaciteit van de methoden voor onzekerheidsschatting te beoordelen. Ten slotte passen we regressiemodellen voor onzekerheidsquantificering aan om te kunnen leren van gecensureerde labels, die vaak beschikbaar zijn in farmaceutische datasets, maar zelden kunnen worden gebruikt door conventionele machine learning-methoden.

Het tweede deel van dit werk (hoofdstuk 4) richt zich op het gebruik van Hamiltonian Monte Carlo (HMC)-sampling voor onzekerheidsquantificatie in QSAR-modellen. Hoewel HMC wordt beschouwd als de gouden standaard voor posterieure benadering, maakt de hoge rekenkost het onpraktisch voor de meeste toepassingen in de praktijk. In dit werk willen we HMC gebruiken voor Bayesiaanse onzekerheidsschatting. Ten eerste hanteren we een benadering met lage complexiteit die HMC-gebaseerde onzekerheidsquantificering mogelijk maakt door slechts delen van het machine learning-model te bemonsteren. Ten tweede willen we een grondwaarheid ontwikkelen voor onzekerheidsquantificering in voorspellingen van geneesmiddel-doelwitinteracties door HMC-sampling toe te passen op het gehele QSAR-model. Daarnaast richten we ons op het identificeren van strategieën die goed functionerende HMC-samplers opleveren die effectief samples uit de posterieure kunnen genereren .

Door verschillende benaderingen voor onzekerheidsquantificatie toe te passen op verschillende datasets, biedt dit werk een uitgebreide studie van benaderingen voor onzekerheidsquantificatie voor voorspellingen van interacties tussen geneesmiddelen en doelwitten. In de context van geneesmiddelenontwikkeling biedt dit werk belangrijke inzichten voor het verkrijgen van betrouwbare onze- kerheidsschattingen, waardoor weloverwogen besluitvorming en een efficiënte pijplijn voor de ontwikkeling van nieuwe geneesmiddelen mogelijk worden. Bovendien belicht het de uitdagingen die bedrijfsinterne data met zich meebrengen, wat de complexiteit van het identificeren van effectieve strategieën voor onzekerheidsschatting in QSAR-modellen benadrukt.

List of Abbreviations

ACE	adaptive calibration error
Adam	adaptive moment estimation
ADME-T	absorption, distribution, metabolism, excretion and toxicity
AI	artificial intelligence
AUC	area under the receiver operating characteristic (ROC) curve
BBB	Bayes by Backprop
BCE	binary cross-entropy loss
BS	Brier score
CDF	cumulative distribution function
CI	confidence interval
CYP	Cytochrome P450
ECE	expected calibration error
ECFP	extended-connectivity fingerprints
ELBO	evidence lower bound
ENCE	expected normalized calibration error
ESS	effective sample size
HBLL	HMC Bayesian last layer
hERG	human ether-a-go-go-related gene
HMC	Hamiltonian Monte Carlo
HP	hyperparameter
iid	independent and identically distributed
KL	Kullback-Leibler

MAO-A	monoamine oxidase A
MAP	maximum a posteriori
MC-Dropout	Monte Carlo-Dropout
MCMC	Markov chain Monte Carlo
ML	machine learning
MLE	maximum likelihood estimation
MLP	multilayer perceptron
MMD	maximum mean discrepancy
MSE	mean squared error
MVE	mean–variance estimation
NLL	negative log-likelihood
NN	neural network
PC	preferred class
PRF	probabilistic random forest
PS	Platt-Scaling
QSAR	quantitative structure–activity relationship
ReLU	rectified linear unit
RF	random forest
RMSE	root mean squared error
RMV	root mean variance
SD	standard deviation
SMILES	simplified molecular input line entry system
TB	target-based
UQ	uncertainty quantification
VA	Venn-ABERS

Contents

Abstract iii

Beknopte samenvatting v

List of Abbreviations viii

List of Symbols ix

Contents ix

List of Figures xiii

List of Tables xv

1 Introduction 1
- 1.1 Motivation 1
- 1.2 Thesis Overview and Key Contributions 3

2 Scientific Background 7
- 2.1 Introduction to Uncertainty Quantification in Machine Learning 7
  - 2.1.1 Frequentist vs. Bayesian UQ 8
  - 2.1.2 Sources of Uncertainty 10
  - 2.1.3 Perspectives on UQ 13
- 2.2 Probability Calibration and Evaluation of Predictive Uncertainty 15
  - 2.2.1 Introduction to Probability Calibration 15
  - 2.2.2 Probability Calibration in Classification 17
  - 2.2.3 Probability Calibration in Regression 21
- 2.3 Decomposition of Proper Scores for Classification 23
  - 2.3.1 Decomposition into Epistemic and Aleatoric Loss 23
  - 2.3.2 Decompositions into Calibration Loss 24
- 2.4 Introduction to Machine Learning Models 26

2.4.1 Random Forests . . . 27
2.4.2 Multilayer Perceptrons . . . 27
2.5 UQ using Bayesian Inference . . . 33
2.5.1 Introduction to the Bayesian Approach . . . 33
2.5.2 Posterior Approximation Approaches . . . 36
2.5.3 Sampling Approaches . . . 38
2.6 UQ using Ensemble-based Methods . . . 48
2.7 UQ using First-Order Risk Minimization . . . 50
2.8 UQ using Second-Order Risk Minimization . . . 51
2.9 Post hoc Calibration . . . 52
2.10 UQ in Drug Discovery–the State of the Art . . . 54

3 Comparing UQ Methods for Drug-Target Interaction Predictions using Real-World Pharmaceutical Data and Temporal Splits 56
3.1 Introduction . . . 58
3.1.1 Real-World Pharmaceutical Dataset . . . 59
3.2 UQ under Temporal Distribution Shift . . . 63
3.2.1 Methods . . . 64
3.2.2 Results and Discussion . . . 71
3.3 Probability Calibration with Experimental Errors . . . 82
3.3.1 Methods . . . 83
3.3.2 Results . . . 85
3.4 Enhancing UQ with Censored Regression Labels . . . 88
3.4.1 Methods . . . 89
3.4.2 Results and Discussion . . . 98

4 UQ in Drug-Target Interaction Models Using Hamiltonian Monte Carlo Sampling 117
4.1 Introduction . . . 118
4.1.1 Public Dataset . . . 119
4.2 A Comprehensive Calibration Study via HP Tuning and HMC Bayesian Last Layer . . . 120
4.2.1 Methods . . . 121
4.2.2 Results and Discussion . . . 127
4.3 Fully Bayesian NN using HMC Sampling for UQ in Drug-Target Interaction Predictions . . . 135
4.3.1 Methods . . . 137
4.3.2 Results and Discussion . . . 139

5 Conclusion and Outlook 145
5.1 Contributions . . . 145

5.1.1 Chapter 3: Comparing UQ Methods for Drug-Target Interaction Predictions using Real-World Pharmaceutical Data and Temporal Splits . . . 146
5.1.2 Chapter 4: UQ in Drug-Target Interaction Models Using Hamiltonian Monte Carlo Sampling . . . 149
5.2 General Conclusions . . . 150
5.3 Future Research . . . 151

A Appendix: Chapter 3 **155**
A.1 UQ under Temporal Distribution Shift . . . 155
A.1.1 Model Selection . . . 156
A.1.2 Distribution Shift in Label Space . . . 157
A.1.3 Distribution Shift in Descriptor Space . . . 158
A.1.4 Results of the Calibration Study . . . 162
A.2 Enhancing UQ with Censored Regression Labels . . . 166
A.2.1 Temporal distributions . . . 166
A.2.2 Model selection . . . 168
A.2.3 Full ablation study . . . 169
A.2.4 Additional model comparison . . . 171

B Appendix: Chapter 4 **175**
B.1 A Comprehensive Calibration Study via HP Tuning and HMC Bayesian Last Layer . . . 175
B.1.1 Probability Calibration Study . . . 176
B.1.2 Additional Experiments . . . 180
B.1.3 Hyperparameter Tuning . . . 181
B.1.4 Computation Time of Models . . . 182
B.2 Fully Bayesian NN with HMC Sampling for UQ in Drug-Target Interaction Prediction . . . 183
B.2.1 Model Selection. . . . 183
B.2.2 Evaluation of HMC Chains. . . . 184

Bibliography **187**

Funding **209**

Curriculum **211**

List of Publications **213**

List of Figures

2.1 Sources of uncertainties. 10
2.2 Sampling from the typical set. 39
2.3 Generation of samples with HMC. 46

3.1 Temporal split. 62
3.2 Classification models: Data overview. 65
3.3 Overview of the classification models. 68
3.4 Distribution shifts between the training and test datasets over time. 71
3.5 T-SNE plots of the ECFP space. 72
3.6 Summary of BCE and ACE scores for the third temporal setting. 76
3.7 Summary of BCE and ACE scores across all temporal settings. 78
3.8 Summary of ACE scores of post hoc probability calibration approaches using the third temporal setting. 80
3.9 SD of the control compounds. 84
3.10 Model calibration over time. 87
3.11 Overview of regression models. 93
3.12 Ablation study. 100
3.13 Predictive accuracy. 104
3.14 Calibration curves. 106
3.15 Combined accuracy of uncertainty estimation and predictive performance. 109
3.16 Epistemic estimates by Bayes by Backprop. 111
3.17 Aleatoric estimates by Gaussian ensemble. 114

4.1 Generation of the public dataset. 120
4.2 Overview of the architecture of the MLP baseline model and the HP tuning workflow. 124
4.3 Overview of model architectures assessed in the model calibration study. 126

4.4 Architecture of the uncertainty estimation approaches combined with PS. 127
4.5 Results of the model selection study for all targets. 129
4.6 ACE vs AUC across model selection strategies. 130
4.7 Results of the model calibration study for all targets. 132
4.8 ACE vs AUC across model architectures. 134
4.9 Results of post hoc-calibrated UQ methods. 135
4.10 Traceplot of the hERG assay. 140
4.11 $\hat{R}$ of the HMC samples across assays. 141
4.12 Autocorrelation between HMC samples across assays. 142
4.13 Effective sample size across assays. 142

A.1 T-SNE plots of the ECFP space for all TB assays. 158
A.2 T-SNE plots of the ECFP space for all ADME-T assays. 159
A.3 Temporal distribution of the label-space. 166
A.4 Temporal distribution of the feature-space. 167
A.5 Ablation study on observed labels only. 169
A.6 Full ablation study of MSE. 170
A.7 Full ablation study of NLL. 170
A.8 Combined accuracy of uncertainty estimation and predictive performance. 171
A.9 Combined accuracy of uncertainty estimation and predictive performance. 172
A.10 Confidence-based calibration curves. 173
A.11 Coefficient of variation. 174

B.1 Traceplot of the MAO-A assay. 184
B.2 Traceplot of the CYP3A4 assay. 185
B.3 Chain 2–4: Autocorrelation between HMC samples across assays. 186

List of Tables

3.1 Overview of the real-world pharmacological data. 60
3.2 Classification models: Data overview. 67
3.3 Summary of AUC results for the third temporal setting. 75
3.4 Overview of RF model performance based on two example assays. 86
3.5 Regression models: Data overview. 91

4.1 Overview of public assay data. 119
4.2 Comparison of model performance between the HMC model and other UQ methods. 143

A.1 Summary of the HP optimization strategy. 156
A.2 Overview of compound ratios of the PC. 157
A.3 Ratio of scaffold overlap between train and test set. 160
A.4 Quantification of the distribution shifts in descriptor space between the calibration and test set. 161
A.5 Summary of the BCE scores for the baseline models. 162
A.6 Continuation of Table A.5 163
A.7 Summary of the ACE scores for the baseline models 164
A.8 Continuation of Table A.7 165
A.9 Model selection. 168

B.1 Optimization with AUC. 176
B.2 Optimization with accuracy. 177
B.3 Optimization with BCE. 178
B.4 Optimization with ACE. 179
B.5 Performance of models with an increased Number of Layers . . 180
B.6 Model performance on the training vs test set. 180
B.7 Overview of the parameter space screened during HP tuning of the baseline MLP model. 181
B.8 Overview of the parameter space used for HBLL tuning. 181

B.9 Time required for models included in the study. 182
B.10 HP tuning of the baseline MLP. 183
B.11 Hyperparameter configuration of the base MLPs. 183

Chapter 1

Introduction

1.1 Motivation

The development of safe and effective drugs is a challenging task associated with high development costs, a high risk of adverse effects or lack of efficacy, which can lead to the failure of a drug candidate or to long approval processes until a drug can be brought to the market (Schlander et al., 2021; Wouters et al., 2020). Machine learning (ML) models can improve the productivity of pharmaceutical companies by leveraging insights from vast amounts of pharmacological data to make predictions and guide experts through the drug discovery process. ML methods are applicable to a wide range of tasks that play a role in drug design (Dara et al., 2022), including the identification and validation of molecular targets (Kamya et al., 2024; Madhukar et al., 2019), synthesis planning (Coley et al., 2018, 2019; Genheden et al., 2020), or the prediction of protein structure with tools like AlphaFold (Jumper et al., 2021). Furthermore, ML models are frequently employed in molecular property prediction tasks, which aim to predict a specific feature from the molecular structure of a compound (Heid et al., 2023a; Stokes et al., 2020). Models that predict activities against biological targets are referred to as quantitative structure–activity relationship (QSAR) models (Cherkasov et al., 2014), which can be used in various applications (Muratov et al., 2020), such as toxicology predictions or drug design. Consequently, these computational techniques have emerged as valuable tools in drug discovery, supporting a shift towards a more time- and resource-efficient pipeline (Chen et al., 2018; Kim et al., 2020; Sarkar et al., 2023).

The ML model generation workflow typically involves numerous steps, ranging from the initial collection of data and selection of an appropriate model architecture to the actual training of the model. In each of these steps, errors are accumulated, which can impair model performance and increase the uncertainty of a model during inference (Begoli et al., 2019). While a main priority in the scientific community has been to accurately predict the features of new instances, the correct quantification of the uncertainty associated with the prediction has often been neglected and has only recently received increasing interest (Mervin et al., 2021a). However, quantifying these uncertainties is paramount for assessing the extent to which a model user can rely on the prediction of a model (Begoli et al., 2019).

Uncertainty quantification (UQ) is considered a valuable tool for assessing the uncertainty associated with ML model predictions, particularly in drug discovery, where incorrect decisions may compromise safety or lead to a waste of resources (Mervin et al., 2021a; Begoli et al., 2019). For example, uncertainties could be leveraged to improve decisions about which candidates to pursue across a candidate portfolio, by selecting compounds that are certain to possess a certain feature or by filtering out those that are likely to trigger adverse drug reactions (Mervin et al., 2021a). Furthermore, Mervin et al. (2021a) argue that UQ is essential in order to successfully realize more automation in the design-make-test-analyze cycle, which plays an important role in designing and improving therapeutic agents.

In this context, the reliability of the uncertainty estimate is paramount for ensuring that the prediction is meaningful and trustworthy. It is important to highlight that high prediction accuracy does not necessarily imply trustworthy uncertainty estimates, as accurate neural networks often fail to provide realistic estimates of their predictive uncertainty (Guo et al., 2017; Mervin et al., 2021a; Wang, 2025). These models are called poorly calibrated, which implies that the model's confidence does not reflect the true probability of making a prediction error. Poorly calibrated models prevent a realistic assessment of the risk and fail to accurately inform decision-making processes. In the drug discovery workflow, poorly calibrated predictions can lead to misguided decisions, increasing both time and resource demands. Since the causes of poor probability calibration are often multifaceted and difficult to control in realistic settings, it is essential to develop strategies that can handle these challenges and yield reliable uncertainty estimates.

1.2 Thesis Overview and Key Contributions

The goal of this work is to evaluate and compare different approaches for drug–target interaction modeling to identify methods that produce well-calibrated uncertainty estimates. It aims to examine a range of factors that influence uncertainty estimation, including model architectures, UQ techniques, data characteristics, and model selection strategies. By addressing these different aspects, this thesis seeks to provide a comprehensive study of uncertainty estimation for bioactivity predictions in a realistic setting.

The thesis is structured into three parts: Chapter 2 outlines the scientific background, while Chapters 3 and 4 present the methodologies and findings of the conducted studies. The studies presented in Chapter 3 and Chapter 4 differ both in the datasets employed to evaluate UQ methods and in the research questions that guide each analysis.

Chapter 2: Scientific Background

In Chapter 2, we present the theoretical foundations of UQ, probability calibration, and proper scores to assess the quality of uncertainty estimates generated by ML models. Furthermore, we discuss the fundamental principles underlying the training and prediction of models considered in this work, and we cover the mathematical background for evaluating the quality of uncertainty estimates using calibration errors and proper scoring rules. Finally, we introduce common UQ approaches and review the literature on their application in drug discovery.

Chapter 3: Comparing UQ Methods for Drug-Target Interaction Predictions using Real-World Pharmaceutical Data and Temporal Splits

In this chapter, we evaluate various UQ approaches for drug–target interaction prediction using temporal splitting on real-world pharmaceutical data. Our contributions include a study of the evolution of this internal dataset over time, a systematic comparison of UQ methods under realistic distribution shifts, and an examination of the integration of lower-quality bioactivity data in QSAR models.

Section 3.2: UQ under Temporal Distribution Shift. In Section 3.2, we leverage the temporal splitting strategy on the internal pharmaceutical dataset to simulate the evolution of drug discovery data within a pharmaceutical company and to investigate distribution shifts in both label and descriptor space over time. Within this context, we systematically compare the performance of widely used UQ methods applied to QSAR models trained on real-world data. We further examine how the performance of these methods evolves over time by evaluating models trained at different time points of the assay history. Finally, we assess the impact of temporal distribution shifts on the effectiveness of post hoc calibration methods in correcting uncertainty estimates, providing insights into the limitations and reliability of current UQ approaches under realistic conditions.

Section 3.3: Probability Calibration with Experimental Errors. The work in Section 3.3 investigates whether incorporating data uncertainty through probabilistic labels can improve uncertainty estimation in classification random forest models. Additionally, we compare the performance and calibration of standard random forest and probabilistic random forest models, both with and without post hoc calibration, using two different calibration approaches. Finally, we leverage a temporal evaluation setting to examine how the evolution of the test set over time impacts model calibration, providing insights into the reliability of uncertainty estimates under realistic conditions.

Section 3.4: Enhancing UQ with Censored Regression Labels. While computational methods for QSAR modeling often suffer from limited data and sparse experimental observations, additional information may be available in the form of censored labels that provide thresholds rather than precise values of observations. In Section 3.4, we develop extended versions of ensemble-based models, Bayesian models, and Gaussian mean–variance estimation (MVE) that can incorporate additional information from censored regression labels. We also adapt evaluation methods to compare models trained with and without these censored labels. Building on this, we conduct a large-scale, temporal evaluation of the resulting censored regression models using internal pharmaceutical assay data. Finally, we demonstrate how the predictions from these models can be applied in practice to support and streamline the drug discovery process through a case study.

Chapter 4: UQ in Drug-Target Interaction Models Using Hamiltonian Monte Carlo Sampling

Bayesian approaches quantify uncertainty in ML models by estimating posterior distributions over model parameters rather than point estimates. Hamiltonian Monte Carlo (HMC) is considered a gold standard for posterior approximation, as it efficiently exploits the posterior's geometry. In Chapter 4, we use HMC sampling for models trained on public data to quantify predictive uncertainty. The HMC models are obtained either by sampling the entire neural network or by sampling parts of the network to improve UQ in conventional models.

Section 4.2: A Comprehensive Calibration Study via HP Tuning and HMC Bayesian Last Layer. Section 4.2 presents a thorough investigation of probability calibration of neural network–based drug–target interaction predictions. We use a computationally efficient Bayesian uncertainty estimation approach that uses HMC trajectories to sample the parameters of a Bayesian logistic regression model applied to the hidden layer of the baseline network. Furthermore, we evaluate different hyperparameter (HP) tuning strategies, including metrics based on accuracy and calibration, to identify approaches that produce the most well-calibrated models.

Section 4.3: Fully Bayesian NN using HMC Sampling for UQ in Drug–Target Interaction Predictions. The preliminary work presented in Section 4.3 focuses on developing a full Bayesian neural network (NN) by applying HMC sampling to whole QSAR models. In the first part, the retrieved samples are analyzed to ensure that the HMC chains are sampling from the true posterior and to detect pathological behaviours of the chain. The second part compares the uncertainty estimates of popular UQ approaches to the predictions of the HMC models, which serve as a ground truth for UQ.

Chapter 2

Scientific Background

2.1 Introduction to Uncertainty Quantification in Machine Learning

As a subfield of artificial intelligence (AI), machine learning (ML) provides a framework that allows algorithms to learn from data in order to generalise to unseen data and predict future outcomes. Among the various ML methods, neural network (NN) have emerged as a successful approach, as their flexibility enables them to handle complex tasks and adapt to a wide range of applications. Consequently, these algorithms have experienced a surge in popularity and are increasingly applied across diverse fields that handle large volumes of data and require modeling of complex problems. One of these fields is drug discovery, where an elemental challenge lies in screening the vast chemical space for promising molecular compounds, which are subsequently optimized to achieve the desired properties (Hertzberg and Pope, 2000). The large-scale and complex nature of this early-stage screening makes it an ideal application for ML models, given their high scalability and predictive abilities (Sadybekov and Katritch, 2023; Bleakley and Yamanishi, 2009; Klambauer et al., 2019; Hochreiter et al., 2018; Engkvist et al., 2023).

Despite their success, NNs have some considerable drawbacks, as they often lack transparency and are hard to interpret due to their black-box nature. Furthermore, the majority of these methods provide a single value to predict the outcome of a future event, also referred to as point estimates, which do not provide information about the uncertainty associated with the predictions. However, taking into account these uncertainties facilitates the assessment of

how certain the model is about its prediction, and, consequently, gives insight into the reliability of a predicted value and its usefulness when basing decisions on it. Probability theory provides a convenient framework for quantifying uncertainties and has emerged as a common tool for modeling them in ML. Given the observed dataset $D = \{x, y\}_{n=1}^{N}$, with x_n, y_n denoting feature, label pairs, the aim is to generate a distribution over the prediction of an unseen datapoint x^*:

$$\mathbb{P}(y^*|x^*, D) \tag{2.1}$$

instead of only providing a point estimate for y^*.

Accounting for uncertainties in model predictions can help to detect challenging instances, such as out-of-domain samples (Hendrycks and Gimpel, 2017), distribution shifts in the data (Ovadia et al., 2019), or adversarial samples (Szegedy et al., 2013). In general, uncertainty quantification (UQ) enables safer and more reliable deployment of ML models in real-world applications by increasing the confidence of humans in the models (Apostolakis, 1990). In drug discovery, UQ is particularly valuable as the field is highly complex, and its experiments are time-consuming, expensive, and high-risk. Furthermore, poor conclusions regarding which experiments to make can be highly wasteful (Mervin et al., 2021a). Therefore, the drug discovery process can significantly benefit from UQ, enabling more informed decision-making and a more efficient allocation of resources.

It should be emphasized that the reliability of uncertainty estimates is crucial to ensure the trustworthiness of ML models and the utility of uncertainty estimates. However, uncertainty estimates often fail to represent the true uncertainty of a model prediction.

The following sections discuss the concepts and methods used in the frequentist and Bayesian frameworks, which are the two major strategies for statistical inference. Furthermore, sources of uncertainty are examined, and the popular categorization into epistemic and aleatoric uncertainty is critically analyzed.

2.1.1 Frequentist vs. Bayesian UQ

The frequentist and the Bayesian setup are two major strategies for statistical inference, which differ in how they model data and the uncertainties associated with ML predictions. Although a strictly dualistic view of these approaches is not always appropriate, given their methodological similarities and the blurred boundaries discussed at the end of this section, a polarizing debate between their advocates has persisted for decades, extending into the ML community as well.

The frequentist approach aims to find the best choice for model parameters to obtain a model that is likely to have produced the data. A common tool is maximum likelihood estimation (MLE), which finds point estimates for the model parameters θ that maximize the probability of the data D:

$$\theta_{MLE} = \underset{\Theta}{\operatorname{argmax}}\ \mathbb{P}(D|\theta) \tag{2.2}$$

MLE serves as the foundation for defining many commonly used loss functions in ML training. Because frequentist approaches obtain point estimates for model parameters, they do not intrinsically quantify the uncertainty associated with a prediction. Thus, UQ approaches exist that aim to generate uncertainty estimates in a frequentist framework. These methods include first-order risk minimization approaches, which include the traditional model training using MLE, as well as post hoc calibration methods that aim to generate true uncertainties from probability-like scores.

Bayesian approaches aim to obtain posterior distributions $\mathbb{P}(\theta|D)$ over model parameters rather than point estimates. The Bayesian framework naturally provides an estimation of the uncertainty, using the distribution over model parameters to generate posterior predictive distributions.

Note that the strict theoretical separation between the frequentist and Bayesian treatment contrasts with the connections that can be drawn between these two frameworks. For example, a Bayesian interpretation of MLE exists, in which the prior distribution is tweaked such that the Bayesian solution reduces to the frequentist treatment. Furthermore, for some UQ approaches, the classification to either one of the frameworks is ambiguous, leaving the boundaries between these paradigms unclear. Ensemble models, for example, combine the predictions of multiple base estimators, which are often trained using MLE. Ensemble models are not solely frequentist, as they consider multiple hypotheses, which can be viewed as draws from the parameter space. However, they are not truly Bayesian because the base estimators correspond only to the modes of the posterior distribution and do not represent the whole posterior. Thus, the frequentist and Bayesian ideologies should not be seen as a strict dualistic system, but rather as a continuum, which provides the opportunity to connect and combine them to benefit from both viewpoints.

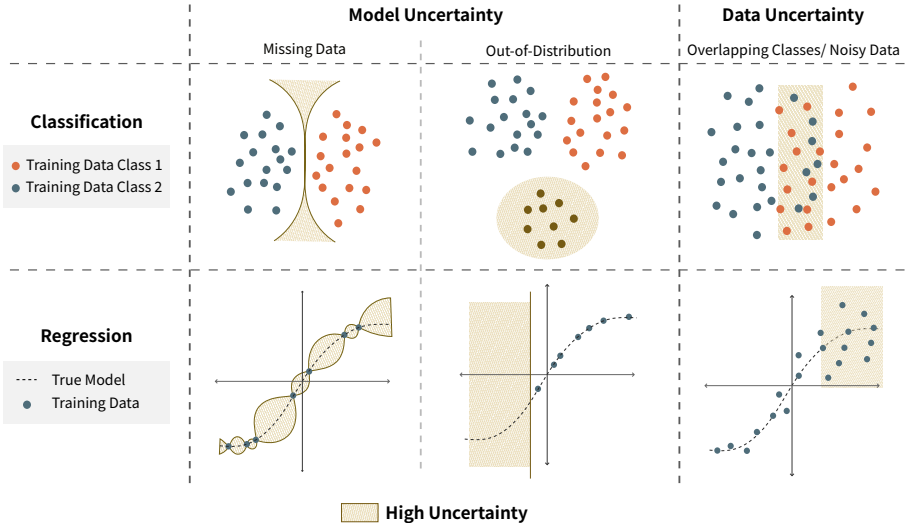

Figure 2.1: **Sources of uncertainties.** Sources of uncertainties in the model (epistemic uncertainty) and in the data (aleatoric uncertainty) are illustrated for classification and regression models. Adapted from Gawlikowski et al. (2023).

2.1.2 Sources of Uncertainty

Ideally, estimates of predictive uncertainty should represent the total uncertainty originating from all sources. Among the methods that enable UQ for ML models, some quantify the total predictive uncertainty, while others provide separate estimates of the uncertainties coming from the respective sources. The UQ methods used in this work are addressed in Section 2.5.

The total uncertainty associated with a prediction can arise from various sources, including errors in the data, the model structure, or the training process. A common classification of sources of uncertainty is the distinction between aleatoric and epistemic uncertainty (Gruber et al., 2023; Hüllermeier and Waegeman, 2021; Kendall and Gal, 2017). Figure 2.1 shows the causes of uncertainties originating from aleatoric and epistemic sources. Although this distinction is widely accepted in scientific literature, recent studies have shown that these concepts are much more vague in practice (Mucsányi et al., 2024). This chapter addresses the various sources of uncertainty and explains why quantifying them separately can be beneficial. Furthermore, the limitations of this theoretical framework are presented in the last section.

Aleatoric Uncertainty. Aleatoric or data uncertainty is induced by the uncertainty related to data and data acquisition. It refers to the random variability in the data, which may arise from both systematic and unsystematic errors during data collection (Figure 2.1). Noisy data can impair model performance, as it prevents the model from learning correctly from the data. Furthermore, in classification tasks, overlapping classes can lead to high aleatoric uncertainty (Figure 2.1). Aleatoric uncertainty is also often referred to as irreducible uncertainty, implying that it cannot be reduced by adding more data samples. Note that the notion of irreducibility only holds unless new aspects of the process are taken into account. For example, the outcome of a fair dice roll will always be random, thus uncertain, and driven by aleatoric sources. However, if additional information about the process is gathered, such as the positioning or movement of the hand before the throw, the uncertainty will become epistemic and can be reduced (Hüllermeier and Waegeman, 2021).

Epistemic Uncertainty. The epistemic uncertainty is related to deficiencies in the model and is therefore also referred to as model uncertainty. Epistemic uncertainty can be attributed to a lack of knowledge in the model, stemming from out-of-domain samples or distribution shifts between the training and test datasets (Figure 2.1). If the model is trained on additional data originating from the same distribution as the test samples, this uncertainty diminishes or vanishes entirely. Other causes of epistemic uncertainty include model misspecification or errors in the training procedure, which can lead to pathologies such as model overfitting. Uncertainties from these causes are more difficult to mitigate, but can, in theory, also be reduced.

Some approaches exist that model uncertainty originating from distribution shift as a separate class in addition to model and data uncertainty (Malinin and Gales, 2018). In this work, however, the distributional uncertainty is not modeled separately, but instead estimated together with the epistemic uncertainty, or the total predictive uncertainty. As discussed above, epistemic uncertainty can be reduced by adding more data or modifying the model. However, understanding which adjustments are needed requires further dissection of the predicted uncertainty (Gruber et al., 2023).

Disentangling Uncertainty Sources. Knowing the source of the uncertainty can provide valuable insight into the model and guide improvements towards a more confident forecasting tool. For the classification setting, Malinin and Gales (2018) provide an intuitive explanation to distinguish between epistemic and aleatoric uncertainty. Given an ensemble model with M base estimators, the epistemic uncertainty can be quantified by the extent to which the predictions

of the individual base estimators for a new test instance differ from one another. Thus, strong disagreement among them indicates high epistemic uncertainty. A different scenario arises in the case of high aleatoric uncertainty, which will lead to base estimators agreeing on an ambiguous prediction, for example, when the probability of belonging to a class is 0.5 in a binary prediction. It is often stated that an agreement among models automatically implies low epistemic uncertainty. However, note that underfitting models can also result in agreement on an ambiguous prediction.

In drug discovery, awareness of the aleatoric uncertainty in molecular property prediction can lead to better risk management by recognizing and quantifying the unpredictable nature of certain properties or parts of the chemical space (Yang et al., 2019). Quantified epistemic uncertainty, on the other hand, can be used during drug discovery to guide the search through the chemical space by redirecting data collection (Heid et al., 2023b). If the parts of the epistemic uncertainty that relate to missing data or distribution shift can be effectively separated from the remaining model uncertainty, it can also aid in developing the ML model Malinin and Gales (2018). It is important to emphasize that the distinction between aleatoric and epistemic uncertainty sources does not always make sense in practice, where the boundaries of these two concepts become vague. The next section discusses the limitations of the aleatoric-epistemic dichotomy.

Reevaluating the Aleatoric-Epistemic Dichotomy. Although the distinction between aleatoric and epistemic uncertainty works nicely in theory, separating those sources can be more difficult in practice. Considerable discrepancies have recently been reported in how epistemic and aleatoric uncertainties are perceived in the scientific community, indicating a lack of clear definitions for these concepts (Kirchhof et al., 2025). The authors also showed that epistemic uncertainty estimates can be contracting, depending on whether epistemic uncertainty is perceived as a measure of disagreement between single models or as the number of possible models that generated the data. Furthermore, in classification problems, aleatoric uncertainty can arise when class boundaries overlap due to noise in the data, making accurate prediction more difficult. Hüllermeier and Waegeman (2021) showed that by introducing additional dimensions, the classes can become separable, reducing the aleatoric and increasing the epistemic uncertainty. As a result, more data is now required to fit the model to the more complex data. This underlines that the separation between the two classes can become unclear, with aleatoric and epistemic uncertainties potentially shifting into one another. Therefore, distinguishing between these sources only makes sense when restricted to a model of analysis, as reported by Der Kiureghian and Ditlevsen (2009).

In addition, recent studies have reported that many widely used methods for disentangling aleatoric and epistemic uncertainty often fail to do so effectively. Mucsányi et al. (2024) benchmarked a range of methods that aim to separately quantify epistemic and aleatoric uncertainty. The authors report a high correlation between aleatoric and epistemic uncertainty estimates, implying that the methods fail to disentangle the uncertainties correctly. Furthermore, the authors recommend abandoning the one-size-fits-all view of uncertainty and its sources, and instead tailoring the UQ approach to individual tasks.

2.1.3 Perspectives on UQ

The Bayesian Perspective

The Bayesian approach is a natural way to account for uncertainty. Building on Bayes' theorem, model parameters θ are treated as random variables with associated posterior distributions $\mathbb{P}(\theta|D)$ rather than point estimates (Bishop and Nasrabadi, 2006), allowing to take into account model uncertainty. Based on the Bayesian concepts, a posterior predictive distribution over a test instance $\{x^*\}$ can be determined by marginalizing out the parameters θ:

$$\mathbb{P}(y^*|x^*, D) = \int \underbrace{\mathbb{P}(y^*|x^*, \theta)}_{aleatoric} \underbrace{\mathbb{P}(\theta|D)}_{epistemic} d\theta \tag{2.3}$$

The first term in the Equation 2.3 represents the aleatoric uncertainty, corresponding to the posterior predictive distribution over possible outcomes given specific model parameters. The second term denotes the posterior distribution over the parameters $\mathbb{P}(\theta|D)$, which describes the epistemic uncertainty capturing the uncertainty in the model (Malinin and Gales, 2018; Hüllermeier and Waegeman, 2021). Note that the distribution over parameters is often not directly accessible, especially in high-dimensional parameter spaces. The Bayesian approach, including the posterior predictive distribution, is addressed in more detail in Section 2.5.

Multiple ML approaches exist that address this challenge, aiming to model the parameters as probability distributions. The theoretical background of these approaches is provided in Sections 2.5.2 and 2.5.3.

The Information-Theoretic Perspective

Another way to estimate the uncertainty of a prediction is via the entropy of its distribution. In the case of a test instance x^* with a discrete outcome, the entropy denotes the amount of information gained upon observation of the true label y^* (Bishop and Nasrabadi, 2006). A common way to measure the amount of information carried in y^* is the Shannon entropy $H(y^*)$. For discrete label spaces, the Shannon entropy can be calculated as expectation over class labels $k = (1, \ldots, K)$:

$$H[\mathbb{P}(y^*|x^*, D)] = -\sum_{k=1}^{K} \mathbb{P}(y^* = k|x^*, D) \ \log[\mathbb{P}(y^* = k|x^*, D)] \tag{2.4}$$

In regression problems, the sum in 2.4 is replaced by an integral to obtain the differential entropy.

Note that a uniform distribution $\mathbb{P}(y^*|x^*, D)$ corresponds to the lowest information content in the posterior predictive distribution, leading to the highest entropy. Thus, in this case, revealing the true label y^* results in the largest information gain. Intuitively, when the model assigns a probability of 0.5 to each class in a binary prediction task, its uncertainty is maximal, and observing the true label y^* provides considerable information.

The Shannon entropy can be further decomposed into the conditional entropy and the mutual information I, which represent the aleatoric and epistemic components:

$$H[\mathbb{P}(y^*|x^*, D)] = \underbrace{\mathbb{E}_{\mathbb{P}(\theta|D)}[\, H[\mathbb{P}(y^*|x^*, \theta)]\,]}_{aleatoric} + \underbrace{I(y^*, \theta|x^*, D)}_{epistemic} \tag{2.5}$$

The first term represents the conditional entropy, which denotes the entropy of the prediction assuming the true model parameters are known. By fixing θ, the epistemic sources are omitted, thus, the conditional only takes into account aleatoric sources. Since the true parameters are generally unknown, the expectation over the posterior distribution of the parameters is taken. The second term calculates conditional mutual information, which measures the amount of information shared between the new instance and the model parameters. More specifically, the mutual information estimates how much the uncertainty of one variable would be reduced if observing another and is therefore a measure for the epistemic uncertainty associated with the prediction of a new instance. If the two variables are conditionally independent, the

mutual information is zero, meaning that the epistemic component of the total uncertainty vanishes. The mutual information is obtained by calculating the following expected Kullback-Leibler (KL) divergence:

$$I(y^*, \theta | x^*, D) = \mathbb{E}_{\mathbb{P}(\theta|D)}[\, \mathrm{KL}[\, \mathbb{P}(y^*|x^*, \theta) \,||\, \mathbb{P}(y^*|x^*, D)] \,] \tag{2.6}$$

If the expected KL divergence is high, the posterior predictive distribution differs on average from the distribution of y^* conditioned on a specific θ. This implies that the uncertainty about θ would be reduced if the label for y^* were observed, and that the epistemic uncertainty is high.

In general, the entropy and related concepts have been used in numerous papers to quantify the uncertainty in ML predictions (Smith and Gal, 2018; Li and Gal, 2017; Mukhoti et al., 2023; Depeweg et al., 2018). Since this work emphasizes the application of Bayesian methods and Bayesian-inspired approaches, a detailed discussion of the underlying information-theoretic principles is beyond the scope of this thesis. More details can be found in Schweighofer et al. (2024); Bishop (1994).

2.2 Probability Calibration and Evaluation of Predictive Uncertainty

The reliability of a model forecast is crucial for producing meaningful uncertainties that effectively support decision-making processes. This is particularly important in high-stakes domains, such as the drug discovery pipeline, where experiments are costly and poor decisions inevitably prolong the development of new drugs and consume additional resources. For a probabilistic forecast to be reliable, the model must produce calibrated probabilities that reflect the true uncertainty associated with the model predictions. This chapter gives an introduction to probability calibration for classification and regression tasks, along with examples of estimators that assess calibration in probabilistic ML models.

2.2.1 Introduction to Probability Calibration

In general, well-calibrated models refer to models whose probabilistic predictions correspond to the true likelihood that an event occurs. For example, in molecular property prediction, a binary classification model is well-calibrated if, among compounds predicted to be active with 70% probability, about 70% are indeed

observed to be active. Poor model calibration, on the other hand, implies that the model's confidence in its predictions does not accurately reflect the true probability of making a prediction error. In this context, confidence is defined as the degree to which the model believes the prediction is correct. In the case of binary classification, a prediction of 0.5 implies minimum confidence, while the confidence increases with increasing shifts toward the extremes of the probability interval. An overconfident model is too confident in its predictions, which are skewed toward the extremes of the probability range. In contrast, underconfident models generate probabilities that cluster too closely around 0.5, reflecting higher uncertainty. Under- and overconfidence can be quantified using calibration errors, which will be discussed later in this work.

The predictive uncertainty should summarise the total uncertainty associated with the prediction, considering all sources of uncertainty. However, the predicted probabilities have been reported to differ from their ground truth, preventing a reliable risk assessment (Guo et al., 2017; Mervin et al., 2020). In 2017, Guo et al. (2017) drew attention to the inability of classifying NNs to estimate uncertainties of predictions correctly. They reported that despite their high accuracy, large NNs are poorly calibrated, resulting in inaccurate probability estimates. In their paper, the authors linked poor probability calibration to model overfitting, leading to increased probabilistic errors rather than affecting the model's ability to correctly classify test instances. Furthermore, they concluded that model calibration and model accuracy are also likely to be optimized by different HP settings (Guo et al., 2017). While Guo et al. (2017) proposes that the growing size of modern NNs contributes to poor probability calibration, Minderer et al. (2021) found that poor calibration is more related to the model family used (*e.g.* multilayer perceptron (MLP), convolutional NNs) than model size. Wang et al. (2021) lists three major factors that impair the probability calibration of models, including large model size and over-parametrization of models, lack of model regularization and data quality and quantity, as well as imbalanced label distribution in classification. In addition, differences between training and test data distributions have been reported to influence model calibration, as distribution shifts make achieving calibrated UQ more challenging (Ovadia et al., 2019; Minderer et al., 2021).

Different approaches exist that aim to obtain well-calibrated probabilities for ML models. In this work, we distinguish between two types of calibration techniques, which differ fundamentally in how they tackle uncertainty estimation. First, post hoc calibration methods correct the predicted uncertainties after model training using a calibrating function. These methods are usually easy to train, however, they require a separate calibration dataset to train the calibrator function, which is used in the additional post-processing step. Second, other approaches employ more sophisticated techniques to drive the model to learn

calibrated uncertainties already during network training. They include Bayesian methods, or methods inspired by the Bayesian setup, as well as other approaches that rely on modified loss functions to estimate uncertainty correctly.

The following sections give an introduction to probability calibration in classification and regression modeling and explain how probabilities can be assessed using calibration errors.

2.2.2 Probability Calibration in Classification

Probabilistic classifiers assign probabilities to all possible outcomes. Link functions, such as the softmax function, allow the transformation of a general classification score to a class label probability, which can be directly interpreted as an uncertainty estimate. This chapter defines calibration in classification models and describes how to quantitatively assess it. Furthermore, proper scores are discussed, which enable a more extensive assessment of the model performance.

Definition of Calibration in Classification. Let the random variables X and Y denote the descriptor and the target variable. The target variable takes values in a set of classes $1, 2, ..., L$. Further, consider a probabilistic classifier $f : \mathcal{X} \rightarrow \Delta^L$ that maps the descriptors X from the feature space $\mathcal{X}$ to the probability simplex. The resulting forecasting scheme is a L-dimensional random variable, which takes values from the simplex and is represented as a vector $\hat{\mathbf{y}} = \{p_1, p_2, ...p_L\}$ with $p_l \in [0, 1]$ and $\sum p_l = 1$. Following Bröcker (2009); Kumar et al. (2019); Zadrozny and Elkan (2002), a model $f(X)$ is calibrated, if its predictions $\hat{\mathbf{Y}} = f(X)$ are calibrated:

$$\mathbb{P}(Y = l|\hat{\mathbf{Y}} = \mathbf{p}) = \mathbf{p}_l \quad \forall\, \mathbf{p} \in \Delta^L \quad \forall\, l = 1, \ldots, L. \tag{2.7}$$

Therefore, if the predicted probability vector is $\mathbf{p}$, Y should be distributed according to $\mathbf{p}$ in a calibrated setting.

The problems addressed in this thesis are modeled as binary classification tasks, in which case the target variable $Y \in \{0, 1\}$. In this framework, a prediction is calibrated if

$$\mathbb{P}(Y = 1|\hat{Y} = p) = p \quad \forall p \in [0, 1]. \tag{2.8}$$

For example, a binary classification QSAR model is considered calibrated if, among the compounds predicted to have a specific property with 70% probability, approximately 70% actually exhibit that property. The term $\mathbb{P}(Y = 1|\hat{Y} = p)$ denotes the calibrated probability distribution given the forecast $\hat{Y}$, which will be referred to as Y_{cal} throughout this work.

Measuring Calibration in Binary Classification

As discussed in the previous sections, uncertainty-aware models ideally assign calibrated probabilities to events. In this framework, methods are needed that assess if the predicted probabilities are correct or to what extent the predicted distribution differs from the calibrated distribution. The L_p calibration error CE_p measures the difference between these two distributions (Kumar et al., 2019; Naeini et al., 2015):

$$CE_p = \mathbb{E}\left[\left\|\hat{Y} - \mathbb{E}[Y|\hat{Y}]\right\|_p^p\right]^{\frac{1}{p}} \tag{2.9}$$

where $\|.\|_p$ denotes the p-norm of a vector. The computation of the calibration error requires the conditional expectation $\mathbb{E}[Y|\hat{Y}]$, which denotes the calibrated distribution and will be referred to as Y_{cal} in this work. Intuitively, Y_{cal} is the ratio of those compounds with $y = 1$, for which the model has predicted the same $\hat{y}$. Since $\hat{Y} = f(X)$ and $f(X)$ is continuous, the calculation of this conditional expectation term is, in general, not possible unless additional smoothness assumptions are made. Several binning approaches exist that circumvent the exact computation of the conditional expectations by providing an estimate of them, and subsequently of the calibration error.

The following sections summarize commonly used strategies for assessing model calibration in binary classification. Further methods exist that estimate the calibration error in a multi-class setting (Kumar et al., 2018, 2019; Gupta et al., 2021; Widmann et al., 2019). Since this thesis focuses exclusively on QSAR modeling as either a binary classification or regression problem, the estimation of calibration error for multi-class settings falls outside its scope and will not be further addressed.

Estimation of Calibration Errors via Binning. Binning strategies partition the probability interval into bins $b = \{1, \ldots, B\}$ and take a weighted average of the errors over all bins (Naeini et al., 2015; Nixon et al., 2019). The most frequently used form of calibration error is the expected calibration error (ECE) (Naeini et al., 2015), which is Equation 2.9 using the L_1 norm. For a dataset with N instances, the ECE measures the average error over B equally-spaced bins. The error in each bin b is weighted according to the number of instances n_b in each bin:

$$\text{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left| \frac{\sum_{i \in b} y_i}{n_b} - \frac{\sum_{i \in b} \hat{y}_i}{n_b} \right|. \tag{2.10}$$

The first term represents the fraction of positives in bin b given the prediction, while the second term denotes the mean of predictions in this bin.

One significant disadvantage of the ECE originates from the binning strategy of this estimate. For skewed distributions, this approach can lead to sparsely populated bins, causing some instances to contribute disproportionately to the error compared to others. Furthermore, fixing the number of bins implies different bias-variance trade-offs in the bins. With an increasing number of measurements in a bin, the bias of the ECE estimate increases, while the variance in this bin decreases.

To alleviate some of the drawbacks, other binning strategies have been proposed. Nixon et al. (2019) introduced the adaptive calibration error (ACE), which calculates the calibration error similarly to the ECE, except with an equal-mass binning approach. In contrast to equally-spaced binning, equal-mass binning ensures that all bins contain approximately the same number of instances. Furthermore, Popordanoska et al. (2024) proposed to use Dirichlet kernel density estimates to estimate the L_p calibration error. In this work, we will use the ECE and ACE for the evaluation of the results.

Proper Scoring Rules for Binary Classification While the previously discussed calibration errors assess how well calibrated the predicted probabilities are, they do not provide insights into the model's accuracy. However, both the calibration and accuracy of computational models are crucial to minimize the risk $\mathcal{R}$, which is the expected loss $\mathcal{L}$ associated with the prediction $f(X)$:

$$\mathcal{R}(f) = \mathbb{E}_{(X,Y)}[\mathcal{L}(f(X), Y)] \tag{2.11}$$

Calibration errors, like the ECE or ACE, fail to account for all critical aspects that contribute to the risk of a model's predictions. For instance, a binary classifier that always predicts the base rate of actives in the dataset may be perfectly calibrated, although it is a poor approximation of the true underlying model and yields predictions that lack meaningful insight. To obtain a more genuine estimation of the risk associated with a model, the use of proper scores (Gneiting and Raftery, 2007) has been proposed, which take into account both calibration and accuracy. In the remainder of this chapter, we present the theoretical foundations for model evaluation using proper scores.

A scoring rule $s(\hat{Y}, y)$ measures the quality of probabilistic predictions $\hat{Y}$ for the observation y. The expected score compares the predicted distribution $\hat{Y}$ with the true, hence calibrated, distribution Y_{cal}:

$$S(\hat{Y}, Y_{cal}) = \mathbb{E}_{y \sim Y_{cal}}[s(\hat{Y}, y)] \tag{2.12}$$

Proper scoring rules, or proper scores, are a class of scoring rules that reach the optimum if a model predicts the underlying distribution of Y. In this work, we treat proper scores as negatively oriented, in which case they are also referred to as proper loss, implying that lower values correspond to better predictive performance.

We denote the divergence d between the predicted distribution $\hat{Y}$ and the calibrated distribution Y_{cal} as:

$$d(\hat{Y}, Y_{cal}) = S(\hat{Y}, Y_{cal}) - S(Y_{cal}, Y_{cal}). \tag{2.13}$$

A scoring rule is proper if it is always non-negative. If $d = 0$ implies $\hat{Y} = Y_{cal}$, then the score is strictly proper.

Strictly proper scores measure the difference between two probability distributions. They can be related to Bregman divergences, which are addressed in detail in Ovcharov (2018); Hendrickson and Buehler (1971). Hence, every proper score can be associated with a Bregman divergence $D_\phi(\hat{Y}, Y_{cal})$ which is induced by a convex generalized entropy function ϕ. The Bregman divergences also play a role in the decomposition of proper scores into a calibration and sharpness term (Bröcker, 2009; Ovcharov, 2018). The calibration errors obtained from the decomposition of proper scores are called proper calibration errors (Gruber and Buettner, 2022).

2.2.3 Probability Calibration in Regression

Unlike classification models, regression models output continuous predictions with labels $Y \in \mathbb{R}$. The predictions are typically aligned with the scale and units of the observed data and, therefore, cannot be immediately interpreted as probability. In regression tasks, uncertainty is typically represented by the standard deviation (SD) of the prediction. This chapter defines calibration for regression models and gives an overview of methods to evaluate calibration in the regression case.

Definition of Calibration in Regression. Consider a regression model $f : \mathcal{X} \rightarrow \mathcal{Y}$ with descriptor variable X and target Y. To introduce the probabilistic aspect, the prediction $\hat{y}_i$ is typically modeled as a Gaussian distribution:

$$\hat{y}_i \sim \mathcal{N}(f(x_i), \sigma_i). \tag{2.14}$$

Further, we define F_X to be the cumulative distribution function (CDF) of this probability distribution. Following Kuleshov et al. (2018), a regression model is calibrated if

$$\mathbb{P}(\hat{Y} \leq F_X^{-1}(p)) = p, \text{ for all } p \in [0, 1]. \tag{2.15}$$

Thus, if the empirical CDF aligns with the predicted CDF, the model is perfectly calibrated.

Furthermore, Kuleshov et al. (2018) extend the definition in 2.15 to confidence intervals:

$$\mathbb{P}(F_X^{-1}(p_1) \leq \hat{Y} \leq F_X^{-1}(p_2)) = p_2 - p_1, \text{ for all } p_1, p_2 \in [0, 1]. \tag{2.16}$$

Measuring Calibration in Regression

A popular approach to evaluate if a regression model is calibrated and Equation 2.15 holds is to assess the confidence-based calibration curve. The confidence-based calibration curve is obtained by computing the z% confidence interval (CI) for every predicted uncertainty in the test set. Next, the observed fraction of errors within each CI is computed for several expected fractions between 0 and 1. The resulting confidence-based calibration curve can be subsequently compared to the identity function, which corresponds to perfect calibration

(Hubschneider et al., 2019; Wang et al., 2021; Heid et al., 2023b; Yang and Li, 2023).

While the confidence-based calibration curve solely considers the calibration of the predictions, other metrics evaluate the intertwined predictive performance and calibration of uncertainties. These metrics include the Gaussian negative log-likelihood (NLL) and the expected normalized calibration error (ENCE) (Levi et al., 2022). The ENCE metric is derived from the error-based calibration plot proposed by Levi et al. (2022), which is made from a binned representation of the root mean squared error (RMSE) and the root mean variance (RMV), i.e., the predicted uncertainty. Computationally, the errors and corresponding predicted uncertainties are ordered based on increasing predicted uncertainty and split into $1, 2, ..., B$ bins. For each bin b of size n_b the RMSE and RMV are calculated as,

$$\mathrm{RMSE}_b = \sqrt{\frac{1}{n_b}\sum_{i\in b}(y_i - \mu_i)^2}, \quad \mathrm{RMV}_b = \sqrt{\frac{1}{n_b}\sum_{i\in b}\sigma_i^2}. \tag{2.17}$$

The bins are then summarized to give the ENCE metric as follows,

$$\mathrm{ENCE} = \frac{1}{N}\sum_{b=1}^{B}\frac{|\mathrm{RMSE}_b - \mathrm{RMV}_b|}{\mathrm{RMV}_b} \tag{2.18}$$

with N denoting the number of total instances.

Other metrics have been proposed and used to evaluate uncertainty estimates in drug discovery applications, such as Spearman's Rank Correlation Coefficient between predicted uncertainties and corresponding errors (Hirschfeld et al., 2020; Wang et al., 2021; Dutschmann et al., 2023; Yin et al., 2023). However, this score has been criticized due to the stochasticity and unreliability of the result (Rasmussen et al., 2023). Statistically, a data point with high predicted uncertainty can still result in a prediction with low error and vice versa. Therefore, we discard the metric from our analysis.

2.3 Decomposition of Proper Scores for Classification

Several decompositions of proper scores exist (Kull and Flach, 2015; Brier, 1950), which reflect different aspects of uncertainty estimates. Kull and Flach (2015) present two types of proper loss decompositions, including the decomposition into epistemic and aleatoric loss, and into calibration and refinement loss. The studies presented in this work primarily focus on components derived from the calibration–refinement decomposition to evaluate model performance, assessing calibration error and ranking ability. However, the research presented in Chapter 3.4 also provides a disentanglement of epistemic and aleatoric sources of uncertainty.

As the individual components of the decompositions are examined to assess model performance in subsequent sections of this thesis, this chapter presents a detailed explanation of the decomposition of proper scores. A definition of proper scores can be found in Section 2.2.2. We consider the following setup to decompose proper losses: Let X and Y be random variables representing the descriptor and the target variable. In a binary classification setting, $Y \in \{0, 1\}$ with predictions $\hat{\mathbf{y}} \in [0, 1]$. Furthermore, the calibrated distribution $\mathbb{E}(Y|\hat{Y})$ is denoted with Y_{cal}, while d denotes the divergence associated with the proper score (e.g., KL-divergence for binary cross-entropy loss (BCE)).

2.3.1 Decomposition into Epistemic and Aleatoric Loss

Following (Kull and Flach, 2015), proper losses $S(\hat{Y}, Y_{cal})$ can be decomposed as following:

$$S(\hat{Y}, Y_{cal}) = \underbrace{\mathbb{E}[d(\hat{Y}, Z)]}_{\text{Epistemic}} + \underbrace{\mathbb{E}[d(Z, Y)]}_{\text{Aleatoric}} \tag{2.19}$$

where $Z = \mathbb{E}[Y|X]$ in the binary classification case, which is the expected label for all instances with features X and intuitively quantifies the uncertainty in Y. The first term measures the difference between the predictions and the expected label and quantifies epistemic uncertainty, i.e., the error due to the shortcomings of the model. The second term is also referred to as aleatoric, which measures the expected divergence between the labels and the expected labels. It is also called irreducible loss (Kull and Flach, 2015), since it is model-independent and cannot be reduced by adding more data.

While in theory, the distinction between aleatoric and epistemic sources works out nicely, the disentanglement of these uncertainty types can be more problematic in practice, as described in Section 2.1.2 of this work.

2.3.2 Decompositions into Calibration Loss

In this section, we will address the decomposition of proper scores into calibration loss and other terms by following the work of Bröcker (2009).

Decomposition into Calibration and Refinement Loss. Equation 2.13 illustrates how the distance between two distributions can be calculated using the expected score S. Since the score $S(Y_{cal}, Y_{cal})$ between a probability distribution and itself is the entropy $\mathrm{e}(Y_{cal})$ (Gneiting and Raftery, 2007), we use 2.13 to express the score as:

$$S(\hat{Y}, Y_{cal}) = \mathrm{e}(Y_{cal}) + d(\hat{Y}, Y_{cal}) \tag{2.20}$$

By the law of total expectation, the expected score can be written as

$$\mathbb{E}_{X,Y}[s(\hat{Y}, y)] = \mathbb{E}_{X,Y}[\underbrace{\mathbb{E}[s(\hat{Y}, y)|\hat{Y}]}_{S(\hat{Y}, Y_{cal})}]. \tag{2.21}$$

Taking the expectation of 2.20 and substituting the expectation according to 2.21 yields the first decomposition of the expected score into a refinement and a calibration term

$$\mathbb{E}_{X,Y}[s(\hat{Y}, y)] = \underbrace{\mathbb{E}_{X,Y}[\mathrm{e}(Y_{cal})]}_{\text{Refinement}} + \underbrace{\mathbb{E}_{X,Y}[d(\hat{Y}, Y_{cal})]}_{\text{Calibration}} \tag{2.22}$$

where $y \sim Y$ denotes the observed labels, $\hat{Y} = f(X)$ the predicted distribution, and Y_{cal} the calibrated distribution.

The refinement measures the entropy of the calibrated distribution, which denotes the loss due to non-trivial predictions. The calibration loss quantifies the divergence between the predicted and the calibrated distribution, which is the ratio of positives among instances with the same predictions.

Decomposition into Calibration Loss and Sharpness. Following Bröcker (2009), the previously derived refinement term can be further decomposed into

$$\mathbb{E}_{X,Y}[\mathrm{e}(Y_{cal})] = \mathrm{e}(\bar{Y}) - \mathbb{E}_{X,Y}[d(\bar{Y}, Y_{cal})]. \tag{2.23}$$

where $\bar{Y}$ denotes the background probability, therefore, the probability of encountering an observation in the dataset. For the interested reader, the decomposition is explained in more detail by Bröcker (2009). Consequently, the final decomposition of the expected score consists of three terms:

$$\mathbb{E}_{X,Y}[s(\hat{Y}, y)] = \underbrace{\mathrm{e}(\bar{Y})}_{\text{Uncertainty}} \underbrace{- \mathbb{E}_{X,Y}[d(\bar{Y}, Y_{cal})]}_{\text{Sharpness}} + \underbrace{\mathbb{E}_{X,Y}[d(\hat{Y}, Y_{cal})]}_{\text{Calibration}} \tag{2.24}$$

.

To conclude, the expected score can be expressed as Expected Score = Uncertainty − Sharpness + Calibration. The first term, which is referred to as uncertainty, is the entropy of the background probability, which remains constant for each dataset and provides information on the complexity of the prediction tasks. If the entropy is high, the prediction task will be more challenging than if the classes are imbalanced. The second term measures the sharpness, thus the divergence between the background probability and the calibrated distribution. Note that this term contributes negatively to the expected score. This implies that non-trivial predictions are preferred, which are dissimilar from the background probability. The calibration loss should be minimized, as a large divergence leads to a higher score, indicating poor calibration.

Examples of Proper Score Decompositions. The Brier score (BS) (Brier, 1950) is a popular proper score for which the corresponding decomposition is well established (Bröcker, 2009). Let y_i and $\hat{y}_i$ be realizations of the random variables $Y \in \{0, 1\}$ representing binary labels, and $\hat{Y} \in [0, 1]$, representing the probabilistic predictions. The BS is calculated with

$$\text{Brier Score} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{2.25}$$

with N denoting the dataset size.

Analogously to Equation 2.24, the BS can be decomposed into an uncertainty, sharpness, and calibration term. The sharpness term $\mathbb{E}[d(\mathbb{E}[Y|\hat{Y}], \bar{Y})]$ and the calibration error $\mathbb{E}[d(\mathbb{E}[Y|\hat{Y}], \hat{Y})]$ both use binning strategies to estimate

the conditional expectation, which denotes the calibrated distribution Y_{cal}. The probability interval is partitioned into K bins. In each bin, the mean of predictions $\hat{y}_k$ and the ratio of observations y_k are calculated. In addition, the background probability $\bar{y}$ is obtained as the ratio of observations in the dataset. Subsequently, the BS can be decomposed into

$$\frac{1}{N}\sum_{i=1}^{N}(y-\hat{y})^2 = \underbrace{\bar{y}(1-\bar{y})}_{\text{Uncertainty}} - \underbrace{\frac{1}{N}\sum_{k=1}^{K} n_k(y_k-\bar{y})^2}_{\text{Sharpness}} + \underbrace{\frac{1}{N}\sum_{k=1}^{K} n_k(\hat{y}_k-y_k)^2}_{\text{Calibration Error}}. \quad (2.26)$$

The decomposition reveals that, in the case of the BS, the L_2 norm variant of the ECE is an appropriate estimator of the calibration error.

Other proper scores are induced by the KL divergence, including the BCE:

$$\text{NLL} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i).$$

The BCE decomposes into a KL-based calibration error and the Shannon entropy as uncertainty term. Popordanoska et al. (2024) introduced an estimator for all proper calibration errors, including KL-based errors, and refinement terms.

2.4 Introduction to Machine Learning Models

The general aim of ML is to learn patterns from data and utilize this knowledge to facilitate informed decision-making. In this work, supervised learning approaches are used (Goodfellow et al., 2016; Alloghani et al., 2020), which leverage datasets composed of feature vectors and their associated labels to train predictive models, with the aim of generalizing to previously unseen instances. Other approaches exist, including unsupervised (Alloghani et al., 2020; Naeem et al., 2023), semi-supervised (Chapelle et al., 2006), and reinforcement learning (Kaelbling et al., 1996; Sutton et al., 1998) strategies, which are outside the scope of this work and will not be discussed further.

A great number of ML methods have been developed, which differ greatly in their learning approaches and applicability to specific tasks. Bishop and Nasrabadi (2006); Goodfellow et al. (2016) provides a detailed review of the theoretical background of ML methods. All modeling approaches used in this work build on either a random forest (RF) or a MLP model, which are further extended to generate more sophisticated UQ approaches. The following sections provide an introduction to both model types, as well as to their training.

2.4.1 Random Forests

A simple and commonly used ML approach are RF models Breiman (2001), which involves the generation and aggregation of multiple decision trees. Decision trees repeatedly split the data based on selected features, aiming to create groups whose data points are most similar to one another. RF models build an ensemble of decision trees, each of which is trained on different subsets of the training dataset and considers different features to generate the splits. To obtain a point prediction, the average of the individual decision trees is taken. In classification, the resulting prediction is a probability-like score that can be considered an uncertainty estimate. In regression RFs, the SD of the decision tree predictions can be used to quantify the uncertainty.

The training of RF models involves optimizing the data splits performed in each decision tree to build subsets of similar datapoints. In this work, the Gini impurity is used as a loss function during RF classifier training. The Gini impurity measures the quality of the splits by calculating how mixed the class distribution of a subset is. In regression, the mean squared error (MSE) between the mean and the individual members of a resulting split is used to measure the similarity, corresponding to minimizing the variance within each split.

2.4.2 Multilayer Perceptrons

Over the last decades, artificial NNs have become increasingly popular. Inspired by the way the human brain processes signals, they can flexibly learn complex, nonlinear patterns, which makes them applicable to a wide range of problems. The most common NN approaches include MLP, which are also referred to as feed-forward NNs, convolutional NNs (CNN), recurrent NNs (RNN), graph NNs (GNN), and transformers. Among these architectures, MLPs represent the fundamental building blocks of NNs, serving as the foundation for more complex and specialized models. This thesis focuses on the uncertainty quantification of MLP-based predictions for molecular property prediction tasks. The following sections provide an introduction to the architecture of MLPs and to the theoretical background of MLP training.

Introduction to MLPs

Architecture of MLPs. MLPs are feedforward NN models that consist of multiple layers of interconnected nodes that apply nonlinear functions to a linear combination of the inputs (Bishop and Nasrabadi, 2006; Goodfellow et al., 2016). MLPs typically consist of several hidden layers, with the first layer matching the dimensionality of the feature space, and a smaller output layer. In binary classification or regression problems, the output layer usually consists of one neuron, which outputs the predicted value. On the other hand, models that output parameters characterizing a probability distribution to quantify uncertainty, such as MVE or evidential deep learning, require more neurons in the last layer. The layers in between the input and output layer are called hidden layers, which can incorporate a varying number of neurons. Generally, a layer consists of a specified number of nodes, the corresponding weights and biases connecting two consecutive layers, as well as activation functions, which introduce nonlinearity.

MLPs are fully-connected networks, which means that all nodes of one layer are connected to the neurons of the next one. For the input variables $X := \{\mathbf{x_1}, ..., \mathbf{x_M}\}$, the first layer (1) of hidden size M is constructed by taking the linear combinations of these inputs weighted by the weights $w_{ji}^{(1)}$ and adding biases $b_{j0}^{(1)}$, with $j = 1, 2, ...M$:

$$z_j^{(1)} = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + b_j^{(1)}. \tag{2.27}$$

Subsequently, a nonlinear activation function $h(.)$ is applied to retrieve the output activations:

$$a_k^{(1)} = h(z_k). \tag{2.28}$$

Common activation functions include rectified linear unit (ReLU), sigmoid, or tanh. The retrieved output activations can again be transformed by additional layers, following the same concept. These layers take the activated output of the previous layer as input and apply a nonlinear activation function to the linear combinations of input, weighed by the model parameters θ.

Forward pass. A forward pass through the MLP refers to passing input data through the network's layers to compute the output using the current model parameters θ. For example, the output y_m for the feature vector $\mathbf{x}_m$ of a MLP with one hidden layer can be obtained with

$$y_m(\mathbf{x}_m, \theta) = \sigma(\sum_{j=1}^{M} w_j^{(2)} h(\sum_{i=1}^{D} x_i w_{ji}^{(1)} + b_j^{(1)}) + b^{(2)}). \tag{2.29}$$

During inference, forward passes are performed to process input data through the network and generate predictions. Furthermore, forward passes are an integral part of the NN training process, used to compute the training loss, which is then employed to adjust the model parameters through backpropagation. The concept of backpropagation is introduced later in this Chapter.

Predicting with MLPs. The prediction with MLPs requires passing the features of the unknown instances through the NN to obtain the output. Generally, the outputs of MLPs are real-valued. In multiclass classification, the Softmax function is typically used to map the outputs to a probability-like range, whereas in binary classification tasks, the logistic sigmoid function $\sigma(.)$ is usually employed. For regression tasks, on the other hand, where $Y \in \mathbb{R}$, no additional mapping step is required.

Training of Neural Networks

The training of NNs usually requires splitting the available data into three datasets. The training dataset is used to obtain the gradients to update the weights as described in the following sections. The validation dataset serves to monitor the model performance between the training epochs and to detect pathological behaviours, like model overfitting. Furthermore, it is used to tune the HPs of the model or to improve model training with methods such as early stopping. Lastly, a test dataset is required to assess the model performance on unseen data. The following chapters address model training using backpropagation and provide an introduction to the theoretical background of common training losses, optimization algorithms, and regularization approaches.

Error Backpropagation The error backpropagation step is a key component of NN training, enabling the network to update its parameters and learn from data (Goodfellow et al., 2016). The goal is to minimize the error between the predicted output and the target value by adjusting the parameters of the model. The error can be measured through a loss function. In the next chapter, we discuss widely used loss functions and their practical applications.

During model training, forward passes are used to obtain predictions for the training data and calculate the error. Subsequently, the error is used in the backpropagation step to modify the model's parameters. Backpropagation relies on the chain rule to compute the gradients of the loss function with respect to each parameter. Thus, using loss $\mathcal{L}$, for a single weight in layer l, we can compute the gradient as

$$\frac{\partial \mathcal{L}}{\partial w_{jk}^{(l)}} = \frac{\partial \mathcal{L}}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial w_{jk}^{(l)}} \tag{2.30}$$

in which

$$z_j^{(l)} = \sum_{k=1}^{M} w_{jk}^{(l)} a_k^{l-1} + b_j^l. \tag{2.31}$$

We now can define the error δ_j^l in neuron j in layer l as

$$\delta_l^j = \frac{\partial \mathcal{L}}{\partial z_j^{(l)}} \tag{2.32}$$

Using the chain rule, the error can be propagated through the network starting from the output layer. This step is also referred to as the backward pass. The error associated with each layer l is stored as δ^l, which is used to efficiently calculate the local gradients for the parameters in the previous layers. Subsequently, the weights $\mathbf{W}^l$ of a layer l can be updated with

$$\mathbf{W}^l = \mathbf{W}^l - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{W}^l} \tag{2.33}$$

where η denotes the stepsize. The same procedure can be used to update the values for the biases.

Loss Functions

Loss functions are one of the cornerstones for assessing the quality of NN predictions by quantifying the difference between the predictions and the true labels. Since the loss function has a significant impact on the training procedure of NNs, it should be chosen carefully, depending on the task that needs to be solved. Common loss functions include the BCE for binary classification and the MSE for regression tasks. Most algorithms used in this work use one of these loss functions in the optimization process, except the evidential learning and second-order risk minimization approaches, which require more complex loss functions. Below, the BCE and MSE losses are defined, while more specialized losses are addressed in Chapter 2.5, in the sections describing the respective methods.

Binary-cross Entropy Loss. The binary cross-entropy loss (BCE) quantifies the divergence between probability distributions of binary variables. It is commonly used in binary classification settings, where it serves as a loss function to measure the discrepancy between the predicted probability-like score and the label. For a predicted value $\hat{y}_n$ with $n = 1, 2, ..., N$ and the corresponding true label y_n with $Y \in \{0, 1\}$, the BCE is calculated as:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{n=1}^{N}[y_n \log(\hat{y}_n) + (1 - y_n)\log(1 - \hat{y}_n)]. \tag{2.34}$$

Thus, the loss increases as the predicted value and the observed label diverge. One advantage of the BCE is that it belongs to the group of proper losses, so that minimizing the BCE not only optimizes the accuracy but also the uncertainty estimate of the models.

Mean Squared Error. In regression problems, the mean squared error (MSE) is a popular loss function to assess the quality of the model. For instances $n = 1, 2, ..., N$, the MSE is the mean squared distanced between the predictions $\hat{y}_n$ and true labels y_n with $Y \in \mathbb{R}$:

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2 \tag{2.35}$$

Optimization Algorithms

The aim of model optimization is to adjust the model parameters such that the model is capable of generalizing to unseen data and can be used for inference. The optimization algorithm finds minima in the loss surface, corresponding to parameters that fit the data well and minimize the loss. A natural approach to optimization is via gradient descent, where the gradients in the loss surface are used as guidance to find minima. To do so, the gradients are first computed in a backpropagation step as described earlier in this chapter. In the subsequent optimization step, the parameters are updated by the optimizer. This update depends on the predefined learning rate, controlling the convergence towards the local minima.

In this work, the adaptive moment estimation (Adam) optimizer is used, which builds on the stochastic gradient descent approach and enables faster convergence compared to simple stochastic gradient descent (Kingma, 2014).

Optimization with Adam. The Adam optimizer was first introduced by Kingma (2014) and provides an extension to stochastic gradient descent. It includes the momentum by maintaining an exponentially decaying moving average of past gradients. This approach facilitates faster convergence along dimensions with continuously large gradients. Furthermore, Adam also uses the exponentially decaying average of the past squared gradients to adapt the learning rate individually for each dimension. In dimensions with strongly fluctuating gradients, the learning rate will be small, while in the case of stable gradients, the learning rate will be larger.

Regularization Approaches

Due to the high flexibility of large NNs, these models tend to overfit the data by learning not only the relevant patterns but also memorizing the data, including noise. Overfitting leads to overly complex models with impaired generalizability and increased loss on unknown data (Dietterich, 1995; Nowlan and Hinton, 2018). To counteract overfitting in NN, several regularization methods exist, which are reviewed in detail in Bejani and Ghatee (2021); Ying (2019). The regularization techniques used in this work are discussed in the following chapters.

Weight Decay. Weight decay (Krogh and Hertz, 1991) is a popular method to inhibit overfitting by penalizing large weights. More specifically, a penalty is applied to the weights, which increases with the size of the weights. Because

large weights can increase a model's sensitivity to noise, weight decay encourages smaller weights, helping to prevent memorization of the data and unnecessary model complexity. Thus, weight decay prevents overfitting and improves the generalization ability of the model (Bishop and Nasrabadi, 2006; Bejani and Ghatee, 2021; Ying, 2019).

Early Stopping. In early stopping (Bishop and Nasrabadi, 2006; Bejani and Ghatee, 2021; Ying, 2019), the model loss on validation data is monitored during the training process. The validation loss after each training epoch is collected, and the training process is stopped as soon as the validation loss does not improve for a specific number of epochs. This leads to selecting model parameters that correspond to a minimum in the validation loss landscape. Due to its straightforward implementation and simple application, early stopping is a convenient regularization method that has been reported to prevent overfitting effectively (Lodwich et al., 2009).

Dropout. Another popular regularization method is dropout (Hinton et al., 2012; Srivastava et al., 2014), which disables a fraction of the neurons that are not updated in the following training set. In more detail, a dropout rate p is fixed before training, which specifies the probability of disabling neurons and setting them to 0. Dropout can be applied to every layer except the output layer, and encourages neurons to be closer to 0.

2.5 UQ using Bayesian Inference

The Bayesian framework treats model parameters as random variables, aiming to determine their probability distributions rather than their point estimates. This chapter provides an introduction to the Bayesian approach and its key concepts. Furthermore, it addresses methods for generating Bayesian NNs.

2.5.1 Introduction to the Bayesian Approach

The primary objective of Bayesian techniques is to determine uncertainty in the model parameters by modeling them as random variables with associated probability distributions. In the Bayesian treatment, posterior predictive distributions are obtained by marginalizing over the model parameters. Relying on marginalization instead of optimization represents one of the fundamental advantages of the Bayesian approach, as emphasized by Wilson

and Izmailov (2020). Considering various possible parameter configurations can particularly improve the accuracy and calibration of NN model, which are often underspecified, implying many equally good solutions for a given task. One of the core pillars of the Bayesian treatment is Bayes' theorem, which facilitates the computation of these probability distributions and, therefore, builds the theoretical foundation for the probabilistic framework.

Bayes' theorem. Given the data D, the posterior distribution $p(\theta|D)$ of the model parameters θ can be computed with the Bayes' theorem:

$$\mathbb{P}(\theta|D) = \frac{\mathbb{P}(D|\theta)\mathbb{P}(\theta)}{\mathbb{P}(D)}. \tag{2.36}$$

Hence, Bayes' theorem can be expressed in terms of three components that contribute to the posterior: the likelihood, the prior, and the evidence.

The likelihood $\mathbb{P}(D|\theta)$ denotes the probability that the data was predicted by a model parameterized by θ. Different likelihood forms give rise to different loss functions, such as the BCE or the MSE loss.

The term $\mathbb{P}(\theta)$ denotes the prior distribution, thus, the belief about the distribution of the model parameters before any data is observed. If knowledge from prior analysis or experiments is available, an informative prior can be constructed. However, in many cases, there is not enough information available to build an informative prior. Noninformative priors can be chosen instead, which have a flat density and are often called improper, as they do not integrate to 1. An alternative to uninformative priors are weakly-informative priors, which aim to impose some regularization on the posterior without making too strong assumptions (Gelman et al., 2013).

The model evidence $\mathbb{P}(D)$ is obtained by calculating the marginal likelihood:

$$\mathbb{P}(D) = \int \mathbb{P}(D|\theta)\mathbb{P}(\theta)d\theta. \tag{2.37}$$

The evidence measures how well the chosen model fits the data, and acts as a normalization constant, which stays fixed for the chosen data and model class.

Intractable High-Dimensional Posteriors. Calculating the model evidence requires integrating over the parameter space, a task that becomes increasingly difficult as the dimensionality of the posterior distribution grows. When dealing

with high-dimensional posteriors, the integral in the denominator in Equation 2.36 can usually not be solved numerically.

Although high-dimensional posteriors are generally difficult to compute, exact solutions are still possible when the prior and posterior belong to the same parametric family, given a particular likelihood function. A prior distribution that satisfies this condition is referred to as a conjugate prior. Specific examples include the Beta prior, which is a conjugate prior to the Bernoulli likelihood function.

In practice, conjugacy is rarely given, especially in the context of ML models, where likelihood functions are inherently complex and non-linear. Thus, viable alternatives are needed for more complex models, which approximate the posterior distribution without requiring the calculation of the intractable evidence term in Equation 2.36. In these cases, the posterior is usually approximated to avoid calculating the exact posterior. Many of these approaches use the unnormalized posterior as a workaround, which omits the calculation of the evidence term and is proportional to the normalized posterior:

$$\mathbb{P}(\theta|D) \propto \mathbb{P}(D|\theta)\mathbb{P}(\theta). \tag{2.38}$$

There are two main strategies to approximate the Bayesian framework. Posterior approximation approaches, including variational methods, approximate the posterior distribution by optimizing a simpler surrogate distribution. Sampling approaches, on the other hand, retrieve samples from the posterior distribution, which are used to approximate the integral of the posterior predictive distribution.

The Posterior Predictive Distribution. Having obtained the posterior distribution over parameters, there are different approaches to get a prediction for a new test instance. For example, the posterior distribution can be used to obtain the posterior predictive distribution for a new test instance x^*, which can be calculated by marginalizing over θ:

$$p(y^*|x^*, D) = \int_\theta p(y^*|x^*, \theta)p(\theta|D)\, d\theta. \tag{2.39}$$

Other approaches, like the maximum a posteriori (MAP), collapse the posterior distribution to a point estimate as described in Chapter 2.5.2. However, these approaches do not account for model uncertainty, as they use only a single model that maximizes the posterior distribution.

2.5.2 Posterior Approximation Approaches

Approximation of the posterior includes deterministic methods that optimize a simpler surrogate distribution to best represent the posterior distribution. The surrogate distributions are chosen to be tractable, so that they can be solved analytically, leading to an exact solution of the approximate. Methods that approximate the posterior include variational approximation methods, which are also used in this work and are discussed in more detail at the end of this chapter.

Posterior Approximation using Point Estimates

One approach to UQ in ML models is via point estimates of the model parameters. Instead of using the whole posterior distribution to obtain a posterior predictive distribution, these methods use a single estimate of the model parameters and a point prediction for new instances. Intuitively, these methods seem to find the model parameters that are most likely to have generated the data.

The maximum a posteriori (MAP) estimates one of the most straightforward approaches to approximate a posterior distribution, It involves finding the mode of the posterior distribution over the model parameters θ and using it as a point estimate in the model:

$$\theta_{MAP} = \underset{\Theta}{\operatorname{argmax}}\ \mathbb{P}(\theta|D)\mathbb{P}(\theta) \tag{2.40}$$

The MAP estimate is closely related to MLE, which is a common strategy to find model parameters in traditional NN training (Bishop and Nasrabadi, 2006). MLE involves obtaining a point estimate for the model parameters, the likelihood of the data D given the model parameters θ is maximized (Goodfellow et al., 2016).

Variational Inference

Variational inference approaches optimize a surrogate distribution $q(\theta)$ to approximate the complex posterior distribution $p(\theta|D)$. The surrogate distribution comes from a class of simple tractable distributions. In the optimization step, the KL divergence between the posterior and the surrogate distribution is used:

$$KL[q(\theta|D)||p(\theta|D)] = \mathbb{E}_{q(\theta|D)}\left[\log \frac{q(\theta|D)}{p(\theta|D)}\right] \tag{2.41}$$

The Evidence Lower Bound. The KL divergence (Kullback and Leibler, 1951) between the surrogate and the posterior can be expressed in terms of an evidence term and the evidence lower bound (ELBO):

$$KL[q(\theta|D)||p(\theta|D)] = \underbrace{\log p(D)}_{\text{Evidence}} - \underbrace{\mathbb{E}_{q(\theta|D)}\left[\log \frac{p(D|\theta)p(\theta)}{q(\theta|D)}\right]}_{ELBO} \tag{2.42}$$

The computation of the KL divergence requires the calculation of the posterior, thus, the evidence term in the denominator in Equation 2.36, which is usually intractable in higher dimensions. To avoid calculating the posterior, the ELBO in 2.42 can be used for optimization. The ELBO is the lower bound on the true marginalized log likelihood:

$$\log p(\theta) \geq \text{ELBO} \tag{2.43}$$

Since the evidence is constant for a chosen model, the ELBO can be maximized instead of minimizing the KL divergence.

The ELBO loss. Variational approximation approaches use a loss function derived from the ELBO:

$$\mathcal{L}(D, \theta) = KL[q(\theta|D)||p(\theta)] - \mathbb{E}_{q(\theta)}[log(p(D|\theta))] \tag{2.44}$$

The ELBO loss comprises a regularizing KL term and a likelihood term. By minimizing the ELBO loss, the KL divergence is minimized to ensure that the surrogate distribution resembles the prior distribution. The likelihood term is maximized, so that the model fits the data well.

Bayes by Backprop

Bayes by Backprop (BBB) (Blundell et al., 2015) provides a quick solution for obtaining samples from the approximate posterior distribution of NN weights using a variational approximation scheme and a backpropagation framework to update the parameters. The ELBO loss in Equation 2.44 is used as a computable loss function. In general, when sampling from the surrogate distribution, the introduced stochasticity prevents the use of a backpropagation scheme. To enable the computation of gradients, the local reparameterization trick is employed, which converts a sample of parameter-free noise $\epsilon \sim N(0, 1)$ into a sample from the posterior using the learnable parameters of the surrogate distribution. Given the surrogate distribution with mean μ and standard deviation σ, the noise is rescaled by σ and shifted by μ to obtain a sample $\theta_i = \mu_i + \sigma_i * \epsilon$.

2.5.3 Sampling Approaches

Methods that approximate the integral required for computing the posterior predictive distribution (Equation 2.39) include stochastic approaches, which use random sampling to approximate the exact solution. More specifically, instead of using a parametric model for the approximation, sampling methods retrieve a set of random samples from the target distribution. Because they use repeated, random sampling, these methods are often referred to as Monte Carlo methods. After retrieving the parameter samples θ_m, with $m = 1, 2, .., L$, the integral in Equation 2.39 can be approximated with

$$\hat{p}(y^*|x^*, D) = \frac{1}{M} \sum_{m=1}^{M} p(y^*|x^*, \theta_m). \tag{2.45}$$

Given that the samples are drawn independently from the target distribution $p(\theta|D)$, the sampler will converge to the true expectation as the sample size increases:

$$\lim_{M \to \infty} \mathbb{E}[\hat{p}(y^*|x^*, D)] = \mathbb{E}[p(y^*|x^*, D)] \tag{2.46}$$

In a sampling process, the aim is to retrieve samples from the part of the target distribution $\mathbb{P}(\theta)$ that contributes most to any expectation. This part of the distribution is called the typical set, which is addressed in the following section.

Sampling from the Typical Set. In sufficiently high dimensions, the tails of probability distributions are high in volume $d\theta$, but feature negligible densities $f(\theta)$. On the other hand, the density around the mode is high, but the volume is small. This observation stems from the geometrical properties of high-dimensional spaces, which feature much more volume outside a given neighbourhood than inside (Betancourt, 2017). Consequently, the area between these two extremes with significant volume and density contributes most to expectations, and is determined by the product of the volume and the density $f(\theta)d\theta$ as shown in Figure (2.2). This area, also referred to as the typical set, should be ideally explored by the sampler (Betancourt, 2017).

In most cases, the target distributions are too complex to sample from directly. Therefore, simple proposal distributions $q(\theta)$ are used, which allow straightforward sampling. The retrieved samples are subsequently processed to ensure that they represent the target distribution. A straightforward approach is to reweigh the samples, which is referred to as importance sampling and discussed in the next section.

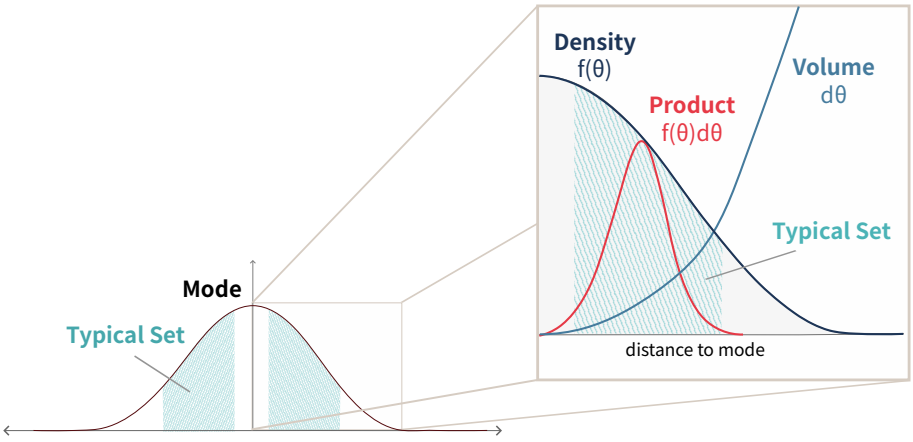

Figure 2.2: **Sampling from the typical set.** Illustration of the typical set of a Gaussian distribution. The area between the mode (high density, low volume) and the tails (low density, high volume) of a distribution contributes most to its expectation. The contribution of any region to the expectation is determined by the product of the probability density and the volume.

Importance Sampling. Importance sampling is used to retrieve weights for samples from a simpler proposal distribution q to correct them towards the target distribution p. With the aim of determining the expectation of a function $f(\theta)$, the contribution of the proposed samples θ_m^* with $m = 1, \ldots, L$ to the expectation is adjusted according to the likelihood of θ_m^* under the proposal and target distribution:

$$\mathbb{E}(f) = \sum_{m=1}^{L} \frac{p(\theta_m^*)}{q(\theta_m^*)} f(\theta_m^*). \tag{2.47}$$

The weights $\frac{p(\theta_m^*)}{q(\theta_m^*)}$ are called importance weights, and are used to alleviate the bias introduced by sampling from the proposal instead of the target distribution.

Sampling from High-Dimensional Posteriors. As dimensions increase, the typical set becomes narrower and sampling from the desired area becomes more difficult. As a result, algorithms that sample randomly from the proposal distributions encounter difficulties in efficiently retrieving samples from the typical set, resulting in low acceptance rates. This is the reason why, for high-dimensional spaces, Markov chain Monte Carlo (MCMC) (see next section)

approaches are often employed as they scale effectively with the increasing dimensionality of the posterior distribution (Bishop and Nasrabadi, 2006). Furthermore, approaches that use the geometry of the target distribution are often more successful in proposing new samples, making the sampling workflow more efficient. These approaches include HMC samplers, which are discussed in more detail at the end of this chapter. Other approaches exist (Bishop and Nasrabadi, 2006), that enable sampling from complex target distributions. However, a detailed discussion of these methods falls outside the scope of this work.

This chapter provides an introduction to MCMC sampling, including the theoretical foundations of Markov chains and diagnostic metrics for assessing sample quality. It concludes with two examples of MCMC, namely Metropolis-Hastings and HMC sampling.

Introduction to MCMC approaches

As discussed previously, the rejection and importance sampling methods suffer from inefficiency as the dimensionality of the posterior space increases. To sample from the parameter space of NNs, a different approach is needed that handles these high-dimensional posteriors more efficiently. Markov chain Monte Carlo (MCMC) samplers are powerful methods that enable sampling from complex posteriors. Instead of retrieving independent samples, MCMC approaches construct a Markov chain of samples $\theta_1, \ldots, \theta_L$, in which in every sampling step m, the proposed sample θ_m is generated based on the proposal distribution $q(\theta_m|\theta_{m-1})$, which depends on the previous sample. Since Markov chains are a core component of MCMC, the following section provides a detailed explanation of their underlying theory.

Markov Chains. The fundamental concept of Markov chains is to form a sequence of samples in which the probability of the current sample depends on the chain of previous samples. Conveniently, this probability is only dependent on the previous sample, a characteristic also known as the Markov property. The conditional probabilities $q(\theta_m|\theta_{m-1})$ for the following samples are also called transition probabilities $T(\theta_{m-1}, \theta_m)$. Transition probabilities are called homogeneous if they remain constant over time, indicating that the probabilities depend only on the previous state and not on the time at which the transition occurs.

The aim of a Markov chain is to retrieve samples that are derived from the desired distribution. This aim is achieved if the Markov chain has converged, so that each step in a Markov chain leaves the distribution $q(\theta)$ invariant. More

specifically, given homogeneous transition probabilities, the marginal probability of a discrete-state Markov chain of θ_m is invariant if

$$q(\theta_m) = \sum_{\theta_{m-1}} T(\theta_{m-1}, \theta_m) q(\theta_{m-1}). \tag{2.48}$$

Note that for continuous parameter spaces, T becomes a transition kernel and the sum in Equation 2.48 is replaced by an integral.

An invariant distribution is also referred to as stationary. Markov chains will converge to a unique stationary distribution as $m \to \infty$, regardless of the initial state of the chain, if the chain is ergodic (Bishop and Nasrabadi, 2006). As with importance and rejection sampling approaches, MCMC methods also include an acceptance step in which the samples are accepted based on the likelihood that the proposal comes from the target distribution p.

Ideal and Pathological Behavior of MCMC Chains. Ideally, Markov chains pass through three phases as the Markov chain progresses (Betancourt, 2017):

1. During the initial phase, the Markov chain is biased from the starting point of the chain. As samples are generated, the chain converges to its stationary distribution. This initial phase is also called the burn-in period, which is eliminated from the final chain used for prediction.

2. In the second phase, the Markov chain explores the typical set by efficiently generating samples derived from the target distribution that are independent of the initialization of the chain. Since the samples are derived from the typical set, they are very valuable and contribute to the calculated expectations, which improve rapidly in accuracy as the number of samples increases.

3. In the third phase, the sampler continues to sample from the typical set. The calculated expectations continue to improve in quality, but the pace of this improvement decelerates as the chain has mostly traversed the target distribution.

Note that Markov chains often do not adhere to this ideal behavior due to challenging probability landscapes that lead to pathological behaviors of the chain. For instance, areas of high curvatures are particularly challenging, as the sampler gets stuck in the vicinity of these regions (Betancourt, 2017). The following chapter discusses widely used metrics for assessing the quality of samples obtained from a Markov chain.

Diagnostic Metrics for MCMC Chains

To evaluate the quality of the simulation, several performance diagnostics exist that provide insight into the behaviour of the chain and the quality of the derived samples. A chain behaves well if it efficiently explores the target distribution without getting stuck, resulting in disproportionately many samples from an area of the probability space. This property, also referred to as mixing, is typically quantified by assessing the autocorrelation of the samples and the effective sample size (ESS) of the chain. Furthermore, convergence diagnostics can be used to evaluate if the chain has converged to the desired target distribution.

Autocorrelation of the Chain. The autocorrelation of a chain is an indicator of its mixing behavior (Neal et al., 2011). A common analysis is the illustration in an autocorrelation plot that shows the autocorrelation between the chain and itself for increasing lags l. In case of a well-mixing chain, the autocorrelation curve will decay fast from 1 at $l = 0$ and will fluctuate around 0 as l increases. This indicates low autocorrelation between samples, hence a well-mixing chain, which explores different regions of the target distribution. Conversely, a slowly decaying autocorrelation curve indicates that the chain might be stuck or moving slowly. Note that, despite a slow exploration, the chain can still explore the whole probability space. However, the high autocorrelation of the samples will result in low precision (Betancourt, 2017).

Effective Sample Size. Another useful diagnostic to determine the quality of a chain is the effective sample size (ESS) (Betancourt, 2017). Intuitively, the ESS gives an estimate of the number of uncorrelated samples in a chain. With this number of samples, the chain of correlated samples can be represented without losing information. A large ESS indicates that sufficient independent draws have been obtained from the posterior, ensuring the stability of the estimators of interest.

For a chain of length N, the ESS can be estimated with an initial positive sequence estimator (Geyer, 2011):

$$ESS_{ips} = \frac{N}{\hat{\tau}}. \tag{2.49}$$

To obtain the nominator $\hat{\tau}$, the sums of the autocorrelation values $\hat{P}_t = \rho_{2t} + \rho_{2t+1}$, hence all neighboring pairs, are calculated. In the process, the autocorrelation values of the pairs are added:

$$\hat{\tau} = -1 + 2\sum_{t=0}^{k} \hat{P}_t \tag{2.50}$$

The addition of the autocorrelation values is stopped as soon as $\hat{P}_t < 0$.

For MCMC runs with multiple chains, the numerator is multiplied by the number of chains M, resulting in:

$$ESS_{ips} = \frac{NM}{\hat{\tau}}. \tag{2.51}$$

Convergence Diagnostics. Convergence diagnostics are useful for assessing whether the chain has converged. One of the most widely used diagnostics is the potential scale reduction factor $\hat{R}$, which was proposed by Gelman and Rubin (1992). The fundamental concept is to assess the variance between chains, which should be similar to the within-chain variance if the chain is mixing well.

To calculate $\hat{R}$, the number of total chains M and samples in a chain N is calculated. Consequently, the number of total samples $S = MN$. Following the notation in Stan (2024), we denote $\theta_{(n,m)}$ as the nth sample from chain m. Furthermore, the sample means over the chain $\bar{\theta}_{(m)} = \frac{1}{N}\sum_{n=1}^{N}\theta_{(n,m)}$ and the grand mean over all of the samples $\bar{\theta} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}_{(m)}$ are calculated to obtain the within-chain variance W

$$W = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{N-1}\sum_{n=1}^{N}(\theta_{(nm)} - \bar{\theta}_{(m)}) \tag{2.52}$$

and the between-chain variance B

$$B = \frac{N}{M-1}\sum_{m=1}^{M}(\bar{\theta}_{(m)} - \bar{\theta}) \tag{2.53}$$

Subsequently, the final estimator can be calculated with:

$$\hat{R} = \sqrt{\frac{\frac{N-1}{N}W + \frac{1}{N}B}{W}} \tag{2.54}$$

As $N \to \infty$, W approaches the numerator in Equation 2.54 and $\hat{R}$ approaches 1. In the initial work of Gelman and Rubin (1992), they reported a threshold

of 1.1 for $\hat{R}$ to be sufficient for convergence. Recently, this threshold was revised in the work of Vehtari et al. (2021), who adopted a more conservative threshold of 1.01. The authors report that the revised threshold enables the detection of a common phenomenon in MCMC chains, in which $\hat{R}$ is fooled by the decreasing within-chain variance of the chains, but fails to detect the subsequent dispersion of the chains after this initial phase, implying that the chains are not yet converged.

Further improvements of the $\hat{R}$ diagnostic have been suggested to detect pathological behaviours of the chain. Since, under specific conditions, the traditional $\hat{R}$ fails to detect non-stationary chains, split-$\hat{R}$ was suggested (Gelman et al., 2013), for which the chains are split into two parts. Furthermore, Vehtari et al. (2021) propose a ranked normalized version of split-$\hat{R}$, that can detect additional unfavorable behaviours, like chains with the same mean, but different variances.

In general, assessing the convergence with $\hat{R}$ does not guarantee that the generated samples enable stable inference or uncertainty estimation. Vehtari et al. (2021) advise the combined use of ESS and a convergence diagnostic to assess the performance of a chain.

MCMC Sampling with Metropolis-Hastings

A commonly used method to ensure the retrieval of samples from a target distribution p is the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970). In the Metropolis-Hastings acceptance step, the likelihood of the sample being accepted depends on the probability of the previous sample and the proposed sample assigned by the target distribution. Furthermore, for asymmetrical proposal distributions, an additional correction term is added involving the transition probabilities. Thus, the acceptance probability of proposed sample θ^* is accepted with probability $A(\theta^*, \theta_n)$:

$$A(\theta^*, \theta_n) = \min[1, \frac{p(\theta^*)}{p(\theta_n)} \frac{q(\theta_n|\theta^*)}{q(\theta^*|\theta_n)}] \tag{2.55}$$

Similar to importance sampling, MCMC methods propose a new sample based on a proposal distribution. There are numerous approaches to propose new samples, including random walk methods that randomly draw proposal samples from the proposal distributions. However, as the number of dimensions increases, it becomes more challenging to retrieve samples, leading to inefficient samplers due to high rejection rates of the randomly proposed samples. Other approaches are required that use the geometry of the posterior space to propose new samples

more efficiently. These approaches include HMC sampling, which is also used in this work. A comprehensive theoretical overview of sampling using HMC is presented in the following chapter.

MCMC Sampling with Hamiltonian Monte Carlo

In this work, we employ Hamiltonian Monte Carlo (HMC) sampling, which scales well to high dimensions. It leverages the principles of Hamiltonian dynamics and uses the shape of the target distribution to achieve efficient Markov transitions through the posterior. It is often considered the gold standard for high-dimensional problems, as it retrieves samples directly from the target distribution and, unlike variational and Laplace approximations, does not rely on approximating the posterior. This chapter focuses on the theoretical foundation of HMC. It is strongly guided by the work of Betancourt (2017), which provides a comprehensive introduction to HMC and is highly recommended to readers seeking a deeper understanding of this topic.

Intuition behind the Hamiltonian framework. HMC chains efficiently sample from the posterior by simulating non-random walks through the log-probability space, guided by information from the posterior landscape. Recall that sampling from the typical set is desirable. This region lies between the extremes of the distribution, has high volume and density, and contributes most significantly to expectation values. To achieve sampling from the typical set, using the gradient of the posterior solely does not suffice, as it will direct the chain to local minima, instead of sampling from the desired area.

To achieve a chain that explores the typical set, the Hamiltonian dynamics can be used, for which an additional momentum variable is introduced. To give an intuition of this framework, Betancourt (2017) reframes the setup, with the modes being planets with surrounding orbits, representing the typical set. Furthermore, a satellite is introduced that moves along its orbit, representing the trajectories along which samples are retrieved. Equivalently to a realistic satellite, the imaginary one requires a specific momentum that needs to be chosen correctly, so that the satellite follows the orbit, thus the typical set, without crashing into the planet, or moving away from the orbit. In the HMC framework, momentum is introduced as an auxiliary variable to make use of Hamiltonian dynamics.

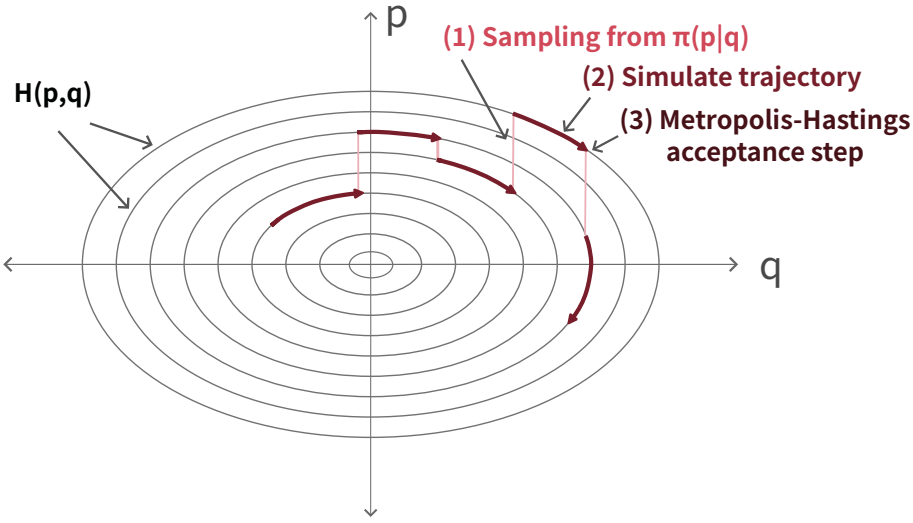

Figure 2.3: **Generation of samples with HMC.** Samples are obtained in three steps: (1) drawing a momentum p from the canonical distribution (shown in light red), (2) simulating Hamiltonian trajectories while preserving the total energy $H(p, q)$(shown in dark red), (3) a Metropolis-Hastings acceptance step. The illustrations were adapted from Betancourt (2017)

Hamiltonian equation. The Hamiltonian of a system defines the total energy as the sum of the kinetic K and potential energy V. This formulation of total energy is used in HMC to efficiently obtain trajectories that follow the typical set through the posterior. The parameter space is extended to include the position variable q and momentum variable p, with their joint probability distribution $\pi(q, p)$, also referred to as canonical distribution. Subsequently, the Hamiltonian can be written as

$$H(p, q) = -\log \pi(p, q) = K(p, q) + V(q). \tag{2.56}$$

The potential energy depends on the position, while the kinetic energy depends on both the position and the momentum. While the potential energy is defined by the posterior distribution, the kinetic energy is defined by the user. A common choice is the Euclidean-Gaussian kinetic energy, which uses a mass matrix M to correlate or decorrelate the posterior space for a more efficient exploration. Betancourt (2017) gives a thorough introduction into different choices of kinetic energies.

Creating Trajectories. To simulate trajectories aligning with the typical set, the partial derivatives of the Hamiltonian are used. The change of position over time is obtained with

$$\frac{\partial H}{\partial p} = \frac{\partial K}{\partial p} \tag{2.57}$$

and the change in momentum over time with

$$-\frac{\partial H}{\partial q} = -\frac{\partial K}{\partial q} - \frac{\partial V}{\partial q}. \tag{2.58}$$

The iterative update of q and p for L steps results in a trajectory that corresponds to the typical set. Note that the volume of the momentum-position phase space must be constant in order to conserve the total energy throughout the simulation. Therefore, at the end of each simulation, the proposed sample needs to pass a Metropolis-Hastings correction step in order to be accepted.

Generation of samples with HMC. The HMC sampling approach is summarized in Algorithm 1 and illustrated in Figure 2.3. The workflow consists of three major steps:

1. Drawing the momentum from the conditional canonical distribution $\pi(p|q)$. Using the Euclidean-Gaussian kinetic energy, the distribution over the momentum is independent of the position, so that ultimately the momentum is sampled from a Gaussian centered at 0 and with M as the covariance matrix.
2. The next step involves simulating the Hamiltonian trajectories with integration time ϵL using the leapfrog integrator. Due to its symplecticity, the leapfrog integrator preserves the energy by intertwining the updates of q and p.
3. To ensure sampling from the target distribution, a Metropolis-Hastings step is applied. If the total energy is maintained, the sample $\tilde{p}$ is accepted.

The choice of HPs L and ϵ has a significant impact on the acceptance rate, thus the efficiency of the sampler. A small stepsize ϵ will result in low integration errors and a high acceptance rate, while a large L ensures low autocorrelation of the samples. However, due to limited computational resources and time, a careful tuning of these HPs is needed to achieve a good balance between low autocorrelation of the samples, a high acceptance rate, and efficiency of the sampler.

Algorithm 1 Sampling with Hamiltonian Monte Carlo

Input Data: $f(x)$, mass matrix M, step size ϵ, number of steps L, initial position q, negative log likelihood V
for t = 1,2,.. **do**
Draw momentum from canonical distribution:
Sample $p^t \sim \mathcal{N}(0, M)$
$(q_0, p_0) = (q^t, p^t)$

Generate Hamiltonian trajectories with leapfrog:
for i = 1 to L **do**
$q_i, p_i \leftarrow$ Leapfrog $(q_{i-1}, p_{i-1}, \epsilon)$
end for
$(\tilde{q}, \tilde{p}) = (q_L, p_L)$

Metropolis-Hastings correction step:
$\rho = e^{(H(\tilde{q},\tilde{p}) - H(q^t,p^t))}$
Sample $u \sim$ Uniform$[0, 1]$
if $u < \min[1, \rho]$ **then** $\tilde{q} = q^{t+1}$
end for

Function Leapfrog(q_i, p_i, ϵ)
$p_{i+\frac{1}{2}} \leftarrow p_i - \frac{\epsilon}{2} \frac{\partial V}{\partial q}(q_i)$
$q_{i+1} \leftarrow q_i + \epsilon\, M^{-1} p_{i+\frac{1}{2}}$
$p_{i+1} \leftarrow p_{i+\frac{1}{2}} - \frac{\epsilon}{2} \frac{\partial V}{\partial q}(q_{i+1})$
return q_{i+1}, p_{i+1}

2.6 UQ using Ensemble-based Methods

Ensemble-based methods aim to obtain numerous base estimators to improve uncertainty estimation and model performance (Hüllermeier and Waegeman, 2021; Lakshminarayanan et al., 2017; Gal, 2016). They are inspired by the Bayesian framework, as the base estimators correspond to samples from the posterior space, which facilitates accounting for uncertainty in the model. A common way to obtain model uncertainty is via the variance of the predictions of the individual base estimators (Malinin and Gales, 2018). Low variance implies low model uncertainty, while contradicting predictions leads to high variance, indicating high model uncertainty.

In this work, a prediction of an ensemble-based model $f(x)$ is obtained by averaging over the predictions of N base estimators $f_i(x)$:

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x) \tag{2.59}$$

Hence, the underlying idea is to consider the predictions of multiple base estimators, which represent different local minima in the loss landscape, corresponding to different modes in the posterior. This leads to increased predictive accuracy compared to considering only one model. Dietterich et al. (2002) claims that ensembles are better at generalizing to new data since they incorporate different local minima, which leads to better predictions. Furthermore, the use of multiple models prevents overfitting to the data. Note that the performance of an ensemble model can suffer if the individual base estimators are not diverse enough or do not perform well (Polikar, 2006).

MLP Ensembles

For UQ with MLP ensembles, multiple base estimators are trained, starting from different weight initializations of the network. To retrieve a single probability estimate, the predictions of the different base estimators are averaged, as described earlier. It is assumed that because of the strong non-convex nature of the error landscape, most of these models reach different local minima Fort et al. (2020). It is, therefore, expected that such sets of base estimators represent different modes of the posterior. Lakshminarayanan et al. (2017) showed that an ensemble of MLP models, referred to as deep ensembles, improved the calibration of uncertainty estimates, as measured by the entropy of the predictive distribution. Deep ensembles have demonstrated greater robustness to data shifts compared to other widely used UQ methods (Ovadia et al., 2019). One major advantage of MLP ensembles is their easy implementation. They can, however, be computationally expensive, as they involve generating multiple models. In the research included in this work, MLP ensembles are also referred to as Ensembles or ensemble models.

Monte-Carlo-Dropout

In the context of training Bayesian NNs, Monte Carlo-Dropout (MC-Dropout) can be regarded as an approximation to Bayesian inference (Gal and Ghahramani, 2016). In Monte Carlo-Dropout (MC-Dropout), parameter samples are retrieved by passing the input data through the NN multiple times. Stochasticity is introduced by applying dropout during inference. During each forward pass through the NN, the output of a subset of randomly selected

neurons is set to zero. This approach resembles the application of dropout during model training, serving also as a regularization technique in NNs (Srivastava et al., 2014). Subsequently, the samples are averaged to obtain an uncertainty estimate for a test instance, as described earlier in this chapter. Since MC-Dropout requires training only one model, this calibration method is efficient in terms of both computational cost and time.

2.7 UQ using First-Order Risk Minimization

In ML, optimization problems are usually addressed as first-order risk minimization, implying optimization of the parameters describing the distribution over the prediction. The first-order risk corresponds to the expected loss between the prediction and the true outcome as defined in Equation 2.11.

In binary classification, the likelihood is modeled as a Bernoulli distribution $X \sim Ber(p)$. The negative logarithm of the Bernoulli distribution corresponds to the BCE loss, which is usually minimized during NN training using gradient descent. After training, the model outputs the most likely parameter p for a given test instance. In the case of the Bernoulli distribution, p can be directly used for UQ, as it also corresponds to the probability that the test instance is active.

Contrarily, in regression problems, the likelihood is modeled as a Gaussian normal distribution $X \sim N(\mu, \sigma)$. The mean μ corresponds to the predicted value of the test instance, while σ gives information about the uncertainty connected to the prediction. The MSE is a common choice for optimizing NNs, which corresponds to minimizing the log of the Gaussian likelihood with the variance assumed to be constant. However, to obtain uncertainty estimates, the model needs to output the model variance, as well. To do so, MVE is used in this work for regression models to obtain uncertainty estimates using first-order risk minimization.

Mean-Variance Estimation

The mean–variance estimation (MVE) approach was originally proposed by Nix and Weigend (1994). Choi et al. (2018) suggests that the uncertainty estimates obtained with MVE are related to the heteroscedastic aleatoric uncertainty, as it captures the noise inherent in data. As such, a NNs can be defined as the mean-variance-estimators which output two values: one representing the predicted value μ_n and the other representing the model's estimate of the aleatoric uncertainty $\sigma^2_{\text{al},n}$. The predicted variance is processed through a

Softplus function to ensure non-negativity and stability. This model is trained using the NLL, which, in the case of a presumed Gaussian distribution, becomes the following,

$$\begin{aligned}\mathcal{L}^{\mathrm{NLL}} &= \frac{1}{N}\sum_{n=1}^{N} -\log\varphi(y_n|\boldsymbol{x}_n,\boldsymbol{\theta}) \\ &= \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\log(\sigma^2_{\mathrm{al},n}) + \frac{(y_n-\mu_n)^2}{2\sigma^2_{\mathrm{al},n}} + \text{constant},\end{aligned} \tag{2.60}$$

where $\varphi(y_n|\boldsymbol{x}_n,\boldsymbol{\theta})$ is the probability density function for the Gaussian distribution with parameters μ_n and $\sigma^2_{\mathrm{al},n}$ estimated by the model with parameters $\boldsymbol{\theta}$.

2.8 UQ using Second-Order Risk Minimization

Approaches that use second-order risk minimization output a second-order moment in addition to the prediction, which gives insight into the distribution over the parameters of the posterior predictive distribution. For example, in binary classification problems, the second-order hypothesis is modeled as a Beta distribution, while in regression problems, parameters of a normal distribution for the mean of the first-order prediction are retrieved. Additionally, the second-order moments of the variance of the first-order distribution are modeled as a gamma distribution. In this thesis, evidential deep learning is chosen as a second-order risk minimization approach and applied to regression tasks.

Evidential Deep Learning

In this work, evidential learning is applied to regression NN as proposed by Amini et al. (2020). This approach leverages the principles of evidential reasoning, where the network outputs parameters of a distribution that characterizes both aleatoric and epistemic uncertainty. In evidential deep learning, the NN outputs four parameters $\gamma_n \in \mathbb{R}, \nu_n > 0, \alpha_n > 1, \beta_n > 0$, which together model the prediction $\mu_n \sim \mathcal{N}(\gamma_n, \sigma_n^2\nu_n^{-1})$ and variance $\sigma_n^2 \sim \Gamma^{-1}(\alpha_n, \beta_n)$. In the implementation, the outputs for ν_n, α_n, and β_n are all passed through a Softplus function, and then 1 is added to α_n. Through this hierarchical model, the parameters express the epistemic uncertainty and aleatoric uncertainty estimates as,

$$\sigma^2_{\mathrm{ep},n} = \frac{\beta_n}{\nu_n(\alpha_n-1)}, \quad \sigma^2_{\mathrm{al},n} = \frac{\beta_n}{\alpha_n-1}. \tag{2.61}$$

The Evidential model is trained using the evidential loss, which optimizes the predicted parameters $\gamma_n, \nu_n, \alpha_n, \beta_n$ to balance their fit to the data and simultaneous calibration of uncertainty estimates. The loss is defined as,

$$\mathcal{L}^{\text{EL}} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\log\left(\frac{\pi}{\nu_n}\right) + \log\left(\frac{\Gamma(\alpha_n)}{\Gamma(\alpha_n+\frac{1}{2})}\right) - \alpha_n \log\left(2\beta_n(1+\nu_n)\right) \quad (2.62)$$

$$+\left(\alpha_n+\frac{1}{2}\right)\log\left((y_n-\gamma_n)^2\nu_n + 2\beta_n(1+\nu_n)\right) \quad (2.63)$$

$$+\lambda|y_n-\gamma_n|\cdot(2\nu_n+\alpha_n), \quad (2.64)$$

where Γ is the Gamma distribution and λ is a HP determining the level of regularization. The first part of the evidential loss in Equation 2.62 corresponds to the log-likelihood of the predicted mean and variance of the normal-inverse Gamma distribution. The second part in Equation 2.63 optimizes the predictions toward the true target values y_n while also accounting for the uncertainty in the predictions. Finally, the last part in Equation 2.64 provides the regularization that penalizes the model for being overly confident.

2.9 Post hoc Calibration

One of the most straightforward techniques is to calibrate probability-like scores via post hoc calibration. Thus, post hoc calibration methods refer to approaches that correct probabilities by fitting a calibrator function to the outputs of a ML model after the training process has been completed. Hence, these methods do not provide uncertainty estimation of the parameters, instead, they provide a first-order UQ of the prediction (Bengs et al., 2023). The calibrating step usually requires a separate dataset, which is referred to as a calibration dataset. Since calibration is performed post hoc, it can be applied to point estimates produced by a wide range of approaches, including models trained with MLE or to more advanced methods.

The application of post hoc calibration methods is usually quite fast and straightforward. Furthermore, their capacity to improve uncertainty calibration without reducing accuracy has established them as a widely adopted approach. However, the effectiveness of these approaches is dependent on the quality of the calibration set, making them vulnerable to distribution shifts between the validation and the test data (Ovadia et al., 2019).

Several parametric and non-parametric methods can be found in the literature that take the post hoc approach to calibrate uncertainty estimates Wang (2025).

In this work, two post hoc probability calibration techniques for classification models are used, including the parametric approach Platt-Scaling (PS) (Platt, 1999) and the non-parametric Venn-ABERS (VA) predictors (Vovk and Petej, 2014). The following chapters give insight into the theoretical background and the application of these two methods.

Platt Scaling

Platt-Scaling (PS) (Platt, 1999) is a parametric post hoc calibration method that has been widely used for calibrating probabilities (Guo et al., 2017; Wang, 2025; Mervin et al., 2020). For the new instance x with prediction $\hat{y}$, PS fits a logistic regression model to the logits of the predictions of a classifier:

$$\mathbb{P}(y|x) = \frac{1}{1 + \exp\left(a\hat{y}_l + b\right)} \tag{2.65}$$

where a, b are the parameters of the logistic regression and $\hat{y}_l$ correspond to the logits to the prediction. The logistic regression is fitted by minimizing the BCE loss using the validation data, with the aim of correcting over- or underconfident model predictions. As such, the calibrated predictions will be optimal on the validation set with respect to the KL-based calibration error (Popordanoska et al., 2024).

Venn-ABERS predictors

The use of Venn-ABERS (VA) predictors involves training isotonic regression functions using the calibration dataset together with the test (Vovk and Petej, 2014) . The procedure involves two training runs: in the first, the test instance is assigned the label 0, in the second, it is labeled 1. The functions represent the hypothesis that the test instance is active versus inactive. As such, the probabilities obtained from the isotonic regression functions correspond to a lower p_0 and an upper bound p_1 on the estimated probability. Finally, these bounds are condensed to a point estimate, as proposed by Toccaceli et al. (2016):

$$\mathbb{P}(y|x) = \frac{p_1}{1 - p_0 + p_1} \tag{2.66}$$

2.10 UQ in Drug Discovery–the State of the Art

In drug discovery, the reliability of the uncertainty estimates is critical when developing new therapeutic agents. As many factors that contribute to poor probability calibration cannot be mitigated in real-world scenarios, it is crucial to identify methods that can handle these challenges to achieve reliable uncertainty estimation. The following sections summarize prior research that has applied and compared the approaches used in this work within the context of drug discovery.

Application of UQ in Drug Discovery. Several papers exist that apply uncertainty estimation to ML models for various drug discovery tasks. These primarily include the application of ensemble models, such as MC-Dropout and deep ensembles, as well as post-hoc calibration approaches. In contrast, a gap exists in the literature concerning Bayesian methods, as their application to drug discovery tasks has rarely been reported.

First, ensemble-based methods, including MC-Dropout and deep ensembles, were successfully applied to tasks that play a role in the development of new therapeutic agents. For example, Kim et al. (2021) and Ryu et al. (2019) use MC-Dropout in graph NN for molecular property prediction, to detect unreliable data points associated with high uncertainty, or to detect noisy data points. Furthermore, Hung and Gini (2021) successfully apply MC-Dropout in mutagenicity prediction tasks. Jiang et al. (2024) propose an ensemble of diverse graph NN models for molecular property prediction, while Rodríguez-Pérez et al. (2022) showed that high ensemble variance can be used to identify predictions associated with higher errors in microsomal stability predictions. Moreover, Heid et al. (2023b) report deep ensembles to be a reliable approach to achieve better estimates of the model variance, and subsequently, of the total uncertainty.

Second, evidential learning has been used to support the drug discovery process by estimating the uncertainties associated with the prediction of inhibitory activity against a target of interest (Wang et al., 2023a,b). Fan et al. (2024) apply evidential deep learning in a binary classification task by replacing the Softmax function with a normalizing flow, which alleviates overconfident uncertainty estimates in molecular property prediction.

Last, post hoc calibration approaches have been used in molecular prediction tasks to correct the uncertainties of ML predictions. Yang and Li (2023) propose a post hoc calibration method, in which part of the model weights are retrained to calibrate the aleatoric uncertainties of an atom-based UQ method based on deep

ensembling and MVE. Furthermore, Zeng et al. (2025) use temperature scaling, which is a simpler variant of PS, to improve the uncertainties in protein-DNA binding site prediction using a multiscale graph convolutional NN combined with a convolutional NN to extract structural and sequence information.

Benchmarking UQ Methods for Drug Discovery. So far, there is no widespread agreement in the literature on which methods are most successful for achieving calibrated probabilities. Hirschfeld et al. (2020) compared common UQ methods for molecular property prediction, including deep ensembles, MVE, and distance-based methods. The authors compared the quality of their uncertainty estimates using seven regression datasets and concluded that none of the methods outperformed the others systematically. A similar conclusion was drawn in another paper by Yin et al. (2023), which compared the performance of deep ensembles, MC-Dropout, MVE, and the evidential deep learning approaches using two molecular prediction datasets. Furthermore, the authors recommend selecting the UQ method based on the downstream use, as different approaches performed well in different tasks. For example, performance was highly dependent on whether the model was used for out-of-distribution prediction.

When comparing MC-Dropout to deep ensembles, the deep ensemble approach was shown to outperform MC-Dropout for deep NN models with different feature representations across a range of prediction tasks in terms of predictive performance and uncertainty estimation (Dutschmann et al., 2023). Fan et al. (2023) use a noise-free dataset of rescored binding affinities to report that the uncertainty estimations produced by MC-Dropout do not always coincide with distance-based uncertainty ranking. Furthermore, the authors reported a significant impact of the chosen compound representation on the reliability of the uncertainties obtained by MC-Dropout.

Regarding post hoc calibration techniques, PS and its simpler version, Temperature scaling, have been demonstrated to improve UQ in bioactivity prediction tasks (Renz et al., 2019). This finding was supported by the paper of Mervin et al. (2020), who also concluded that post hoc probability calibration methods, such as VA predictors, improve the calibration of probabilistic outputs from classification ML models. Furthermore, post hoc calibration approaches have also been shown to be effective when applied to regression models (Dheur and Taieb, 2023). Wang et al. (2021) demonstrated that combining different models, including MVE, ensembling, and distance-based methods, into a weighted average enhances the ability of QSAR regression models to rank errors. Merging this ranked average approach with a post hoc calibration method using NLL leads to well-calibrated predictions, even under domain shift.

Chapter 3

Comparing UQ Methods for Drug-Target Interaction Predictions using Real-World Pharmaceutical Data and Temporal Splits

This chapter was adapted from the following publications:

Section 3.2: Friesacher et al. (2025b)
Friesacher, H.R.[1], Svensson, E., Winiwarter, S., Mervin, L., Arany, A., Engkvist, O., 2025. Temporal distribution shift in real- world pharmaceutical data: Implications for uncertainty quantification in qsar models. Artificial Intelligence in the Life Sciences 8, 100132

Section 3.3: Friesacher et al. (2024)
Friesacher, H.R.[1], Svensson, E., Arany, A., Mervin, L., Engkvist, O., 2024. Temporal evaluation of probability calibration with experimental errors, in: International Workshop on AI in Drug Discovery, Springer. pp. 13–20.

Section 3.4: Svensson et al. (2025)
Svensson, E., Friesacher, H.R.[2], Winiwarter, S., Mervin, L., Arany, A., Engkvist, O., 2025. Enhancing uncertainty quantification in drug discovery with censored regression labels. Artificial Intelligence in the Life Sciences, 100128

[1]F.H.R. was responsible for conceptualization, data curation, methodology, software, investigation, formal analysis, visualization, writing (original draft)
[2]F.H.R. was responsible for software, methodology, data curation, conceptualization, reviewing, and editing of the draft

3.1 Introduction

Estimating uncertainties associated with predictions from QSAR models can accelerate the drug discovery process by identifying promising experiments and enabling efficient resource allocation. Prior research has primarily been conducted on publicly available datasets, using random or scaffold-based splits for evaluation. These splitting strategies have been shown to either overestimate the performance of ML models, in the case of the random split, or underestimate it when using scaffold-based splits (Hirschfeld et al., 2020; Yin et al., 2023; Svensson et al., 2024b). Hirschfeld et al. (2020) highlight the need for a more realistic evaluation, such as a temporal data split, to gain insights into the real implications and nuances between the approaches. Earlier work on internal pharmaceutical assay-based data from Merck compares different splitting strategies and concludes that temporal splitting best approximates the true predictive performance (Sheridan, 2013).

Usually, public benchmarks do not allow proper temporal evaluation as they lack relevant information and sufficient replications for reliable statistics (Yin et al., 2023). If a temporal splitting strategy is feasible, the temporal evaluation of molecular property prediction can be misleading (Lenselink et al., 2017), because the timestamps of data points in public datasets typically refer to when the compound was added to the public domain, rather than when the experiment was performed. Instead, modeling the real evolution of experiments in a pharmaceutical company is what makes a temporal evaluation truly useful (Fluetsch et al., 2024).

This chapter summarizes the research on UQ using real-world pharmaceutical data under temporal splits. The main contributions of this chapter are:

- We analyze the evolution of real-world pharmaceutical data and the resulting temporal distribution shifts in the data [Section 3.2]
- We conduct a large-scale calibration study of common UQ methods for classification [Section 3.2] and regression models [Section 3.4]
- We investigate whether the inclusion of data uncertainty in the form of probabilistic labels improves uncertainty estimation [Section 3.3]
- We adapt the methodological framework to enable the incorporation of censored regression labels and examine their impact on the quality of the uncertainty estimates [Section 3.4]

3.1.1 Real-World Pharmaceutical Dataset

This section gives an overview of the included assays as well as their categorizations. Furthermore, the temporal splitting strategy is described, which was used to simulate the evolution of real-world pharmaceutical data. For more details regarding the data, including dataset generation and feature representation, we refer to Chapters 3.2.1 for the classification analysis and 3.4.1 for the regression analysis.

The research described in this chapter uses internal data from 15 biological assays to gain insight into the properties of real-world pharmaceutical data and subsequently trains ML models for each assay separately. The included assays are diverse and represent different optimization problems typically addressed during the drug discovery workflow to enhance the pharmacokinetic and pharmacodynamic properties of a drug candidate.

The assays were assigned to two categories, target-based (TB) and absorption, distribution, metabolism, excretion and toxicity (ADME-T), and subsequently labeled based on category affiliation and assay description or assay size. The labels of the ADME-T assays were created using "A" as a prefix to indicate the affiliation to the ADME-T category (e.g., "A-logD" for the lipophilicity assay). The TB assays were ordered by size and numbered consecutively (TB-1 for the smallest to TB-7 for the largest TB assay). Table 3.1 provides an overview of the assays used in the work.

TB Assays. The TB category includes project-specific assays from activity screens to identify active substances on a specific target of interest. Active substances are compounds that modulate the function of a protein, for example, by inhibiting or activating the target. This work includes seven TB assays, with assay sizes ranging from 5,082 to 14,605 measured compounds as shown in Table 3.1. As opposed to the ADME-T assays, further specifics regarding these biological assays cannot be disclosed due to proprietary constraints.

ADME-T Assays. Assays in the ADME-T category typically assess the pharmacokinetic properties and toxicity profile of a drug candidate. These properties are connected to the absorption, distribution, metabolism, and excretion (ADME) of a compound, while the toxicity screens identify compounds that hit unintended targets. The ADME-T category comprises assays that assess the general features of a compound, which are typically relevant for its success in the drug discovery and development pipeline (DiMasi, 2001; Van De Waterbeemd and Gifford, 2003). The assays are usually comparatively large and include data from various projects. In our study, eight ADME-T assays were used, including

Table 3.1: **Overview of the real-world pharmacological data.** A general overview of the assays used in this chapter, including their short name, the modeled end-point, the assay unit, and the assay size. Due to proprietary constraints, detailed descriptions can only be provided for ADME-Tassays. The "Goal" column in the table indicates the desired performance of a drug candidate on this assay (↑ for high, ↓ for low values). While LogD targets can be project-specific, a target value of 2 is generally assumed.

Short name	Assay description	Assay End-point	Assay Unit	Assay Goal	Assay size
Target-based					
TB-1	N/A	pIC50	µM	↑	5,082
TB-2	N/A	pIC50	µM	↑	5,237
TB-3	N/A	pIC50	µM	↑	10,465
TB-4	N/A	pIC50	µM	↑	10,624
TB-5	N/A	pEC50	µM	↑	12,612
TB-6	N/A	pIC50	µM	↑	13,093
TB-7	N/A	pIC50	µM	↑	14,605
ADME-T					
A-CYPa	CYP3A4	pIC50	µM	↓	12,875
A-CYPb	CYP2C9 (a)	pIC50	µM	↓	12,876
A-CYPc	CYP2C9 (b)	pIC50	µM	↓	14,062
A-PM	Permeability	Log Perm.	1e-6cm/s	↑	16,511
A-SOL	Solubility	Log Solub.	µM	↑	47,607
A-hERG	Toxicity	pIC50	µM	↓	67,687
A-logD	Lipophilicity	LogD	-	≈ 2	88,114
A-MS	Metabolic Stability	µl/min/1e6	Log Cl_{int}	↓	92,161

four large assays with measurement numbers between 47,607 and 92,161 and four smaller assays comprising 12,875 to 16,511 data points. Table 3.1 lists important properties of the assays as well as the assigned assay labels.

As opposed to the TB assays, the ADME-T assays included in this study are widely used in the drug discovery process, which allows the disclosure of more detailed descriptions of the assays:

- The Cytochrome P450 (CYP) assays measure the inhibition of one of two CYP isoforms, CYP3A4 ("A-CYPa") and CYP2C9 ("A-CYPb" and "A-CYPc"). These isoforms play an essential role in drug metabolism and the detection of drug-drug interactions (Deodhar et al., 2020; Ioannides, 1996; Furge and Guengerich, 2006). Two distinct assay types are available, exploring different interactions with the CYP isoforms. The CYP2C9 (a) and CYP3A4 assays measure drug molecule disappearance using liquid chromatography-mass spectrometry, while the CYP2C9 (b) assay measures CYP inhibition using a fluorescent substrate. In both assays, weaker interactions with the CYP protein are usually favorable to avoid rapid decomposition of the drug molecule and drug-drug interactions.

- The permeability assay ("A-PM") evaluates the flux of a compound across a Caco-2 cell, reflecting its potential in vivo absorption, which is measured in 1e-6 cm/s (Shah et al., 2006). High velocities are favorable, indicating a compound's ability to cross biological membranes.

- The solubility assay ("A-SOL") assesses the maximum concentration of a compound in an aqueous solution at pH 7.4. A Dimethyl sulfoxide (DMSO) stock solution is used, and the organic solvent is evaporated to obtain a solid sample. Compounds with high solubility are preferred to allow sufficient dissolution in biological fluids (Di et al., 2012).

- The human ether-a-go-go-related gene (hERG) assay ("A-hERG") provides vital insight into a compound's toxicity profile by measuring its inhibiting effects on the hERG potassium channel. Inhibition of hERG is correlated to severe cardiac side effects by prolonging the QT interval (Keating and Sanguinetti, 1996). Therefore, inhibiting interactions with hERG is usually undesirable.

- The lipophilicity of a compound is obtained in the "A-logD" assay by measuring the logarithm of the distribution coefficient between octanol and aqueous phase at pH 7.4. Lipophilicity is crucial since it significantly affects drug absorption, metabolism, and safety. A logD greater than 3 has previously been identified as a trigger for safety concerns (Waring, 2010; Chen et al., 2013).

- Finally, the metabolic stability assay ("A-MS") measures how fast a compound is metabolized in rat hepatocytes. The in vivo hepatic clearance is measured in µl/min/million cells. In general, low values for hepatic clearance are desirable, as they imply slower decomposition and, therefore, higher bioavailability of the drug molecule (Masimirembwa et al., 2001; Di et al., 2003).

Temporal Split. To simulate the realistic assay-based modeling of pharmaceutical projects, we split the data of each assay into five folds based on the date of the experiment. Figure 3.1 illustrates the temporal splitting strategy. For each assay, the data points were ordered according to when they were recorded. Subsequently, the data were assigned to five temporal folds, with each fold representing a specific time span in the history of the assay. These folds were then used to set up three experimental settings, using one, two, or three folds for training the QSAR models as shown in Figure 3.1. In each case, the first subsequent fold was used for validation, including model selection, and calibration where applicable.

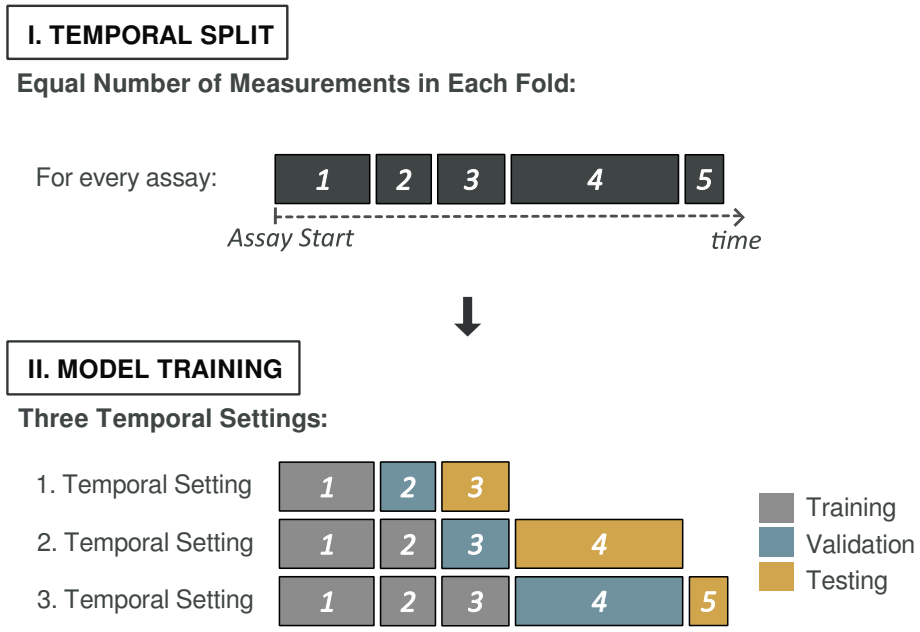

Figure 3.1: **Overview of the temporal split and the temporal settings used in model training.** I. The data was ordered according to the recording date and assigned to 5 folds used in model training and evaluation. The folds were equally sized, so that they included roughly the same number of data samples. Since the number of newly added compounds varies over time, the time spans covered by the folds differ, which is reflected in the varying sizes of the rectangles representing each fold. II. Three temporal settings were considered, with one, two, or three folds in the training set. In each setting, the first subsequent fold is used for validation, and the second subsequent fold is used for testing.

3.2 UQ under Temporal Distribution Shift

This paper makes the following contributions:

- We use a temporal splitting strategy on real-world pharmaceutical data to simulate the evolution of drug discovery data in a pharmaceutical company and to investigate the distribution shifts in label and descriptor space over time.
- In the context of these temporal distribution shifts, we compare the performance of popular UQ methods applied to QSAR models trained on real-world pharmaceutical data.
- We report how the performance of the UQ methods changes over time by comparing models trained at different time points of the assay.
- We assess how the temporal distribution shifts detected in the data affect the ability of post hoc calibration methods to correct uncertainty estimates.

Dundar et al. (2007) reported an intrinsic assumption of many training algorithms that datasets are independent and identically distributed (iid). While this assumption is foundational for traditional ML models, it imposes significant constraints and oversimplifies the complexities of realistic scenarios (Cao, 2022). In the context of probability calibration, deviations from the iid setting have been shown to impair the performance of common uncertainty estimation methods previously reported to improve model calibration under iid conditions (Ovadia et al., 2019; Koh et al., 2021). Consequently, the simplified problems may fail to accurately represent or address the challenges inherent in real-world datasets.

To enable a realistic evaluation of UQ methods for drug-target interaction predictions, a temporal splitting strategy was applied to real-world pharmaceutical data. To our knowledge, only a few studies have addressed the performance of uncertainty estimation approaches under temporal shifts. In some of these works, temporal splitting approaches are applied to ChEMBL (Zdrazil et al., 2024) data, using the publication date as a reference (Wang et al., 2023a). As the date of publication does not correspond to the date when the experiment was conducted, it remains questionable how accurately this information can reflect the timeline in a pharmaceutical company. Other studies (Rodríguez-Pérez et al., 2022; Stoyanova et al., 2023) used internal data from pharmaceutical

companies with the necessary information to perform proper temporal splits. Rodríguez-Pérez et al. (2022) studied the performance of multitask graph NNs for uncertainty estimation, focusing on intrinsic clearance data. Another recent work compares the uncertainty estimation of various regression models for pharmacokinetic property prediction of potential drug molecules (Stoyanova et al., 2023). However, both of these studies use training data from a fixed time span for all experiments and, therefore, do not address model performance over time. Furthermore, they do not address shifts in the data caused by the temporal splitting strategy. Svensson et al. (2025) recently published an extensive temporal study comparing uncertainty estimation methods trained on drug-target interaction data with and without censored data (Chapter 3.4). While this study provides a comprehensive guide on handling uncertainty estimation methods for regression tasks, a comparable large-scale study that applies temporal splitting strategies to different biological assays has yet to be published for classification approaches.

In this work, we aim to address these gaps by assessing the performance of different uncertainty estimation approaches using single-task classification models over time and under assay-specific distribution shifts in the data. A general description of the data as well as the temporal splitting strategy can be found in Section 3.1.1. The first part of this study investigates the evolution of the data and the resulting distribution shifts in label and descriptor space. The second part compares common UQ approaches that employ real-world temporal splits to evaluate model performance in a more realistic setting.

3.2.1 Methods

The following section is structured into three parts to examine the material and methods used in this study. The first section describes the binary classification of compounds and the resulting assay data. The second part addresses the modeling approaches, including the baseline estimators and the more sophisticated uncertainty estimation approaches. The last section provides insight into the experiments and metrics used to compare the uncertainty estimation approaches.

Data

In this study, 7 TB assays and 8 ADME-T assays were used for modeling single-task classification models. A general description of the assays can be found in Section 3.1.1. The assays exhibited different sizes, modeled endpoints, and ratios of compounds belonging to the preferred class (PC). The total size of the assays is illustrated in the upper left panel of Figure 3.2.

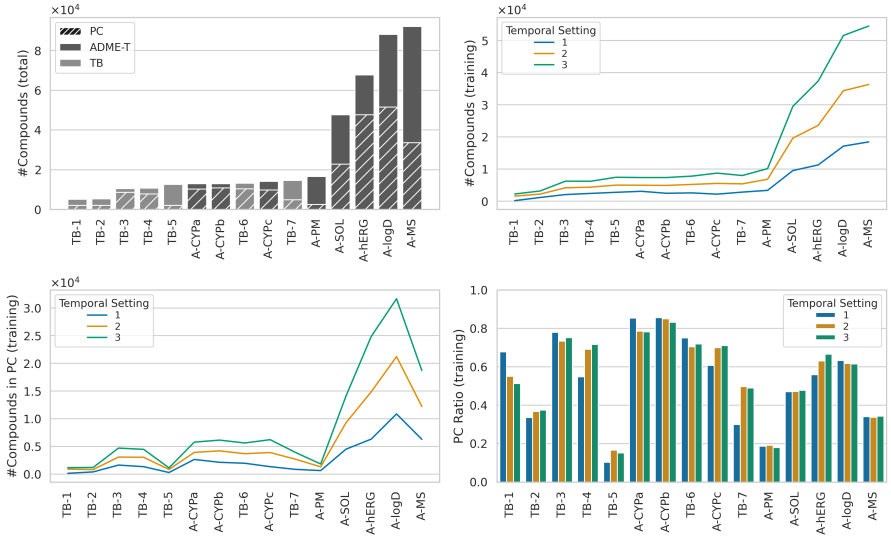

Figure 3.2: **Classification models: Data overview.** The upper left panel plots the total size of the individual assays ordered according to assay size. The striped areas in the bar indicate the number of compounds belonging to the preferred class (PC) in each assay. The upper right panel shows the amount of training data in each temporal setting across all assays, with 1, 2, or 3 temporal folds used for training. In the bottom panels, the total number (left) and ratio (right) of training set compounds belonging to the PC are displayed for each temporal setting and assay. Compound counts are shown in units of 10^4.

Considering all assays and settings, 45 separate training datasets were used throughout this work. The high number of training datasets results from the three different temporal settings included in this study, using one, two, or three folds for training as shown in Figure 3.1. For experiments in which the results of all three settings are examined, the assays are labeled as "*Assay Abbreviation [Temporal Setting]*". Naturally, the training dataset sizes and number of training compounds belonging to the PC vary across the temporal settings as shown in the upper right and lower left panels of Figure 3.2. Furthermore, the bottom right plot in Figure 3.2 demonstrates that the ratios of PC compounds in the training sets differ between temporal settings.

Binary Classification of Compounds. The measured values were converted to a logarithmic scale, and a suitable threshold was determined for each assay individually. Assay-specific thresholds were defined to determine if a compound

belongs to the PC as shown in Table 3.2. All TB assays obtain a compound's inhibiting potency by measuring the IC50, except TB-5, in which the EC50 was used. The IC50 value measures the compound concentration needed to inhibit a protein's activity by half, while the EC50 value indicates the compound's concentration that triggers half of the maximum possible effect. Subsequently, these values were converted to pIC50/pEC50 by taking the negative logarithm of the measurements converted to molar. The PCs in the TB assays comprise compounds with a pIC50/pEC50 value above 6, indicating that a substance achieves the desired effect when its IC50/EC50 value is below 1 μM. In addition, four ADME-T assays, including the three CYP and the hERG assay, contain pIC50 values. Since these assays aim to detect interaction with off-targets, lower pIC50 thresholds of 5 were selected to decrease the risk of false negatives. The PC includes compounds with pIC50 values below this threshold. For A-logD, A-SOL, A-PM, and A-MS, individual thresholds were chosen as shown in Table 3.2 to assign compounds with desirable properties to the PC.

Molecular Representations. Extended-connectivity fingerprints (ECFP) (Rogers and Hahn, 2010) were generated from simplified molecular input line entry system (SMILES) representations of the compounds and used as model input. ECFP are one of the most widely used fingerprint types for molecular property prediction. While more advanced molecular representations and model architectures exist, such as graph NNs for molecular graphs or language models for SMILES representations, ECFPs have been demonstrated to perform comparably to these more sophisticated approaches (Jiang et al., 2021; Mayr et al., 2018). Furthermore, since our study aims to gain insight into uncertainty estimation in QSAR models rather than finding the best approach or comparing molecular representations, we opted for the simple and popular ECFP representation. The RDKit package (Landrum, 2006) was used to generate ECFPs of length 4096 from the SMILES (Weininger, 1988) of the compound structures. Due to additional computational constraints, we concentrated on RF and MLP models as suitable choices for examining UQ in a temporal context.

Model Generation

A Python package is publicly available at `https://github.com/MolecularAI/uq4dd`, which contains the code used for model generation and evaluation inspired by the design pattern proposed by Hartog et al. (2024). Figure 3.3 provides an overview of the models compared in this study.

Table 3.2: **Classification models: Data overview.** Details of the assays used in this study, including assay size and modeled endpoint. Descriptions of the ADME-T assays are shown. The ratio of compounds belonging to the PC is reported for each assay. The last column indicates the corresponding threshold T used to assign compounds to classes and if the PC lie above or below T (PC $< / > T$).

Abbreviation	Assay Description	Assay Size	PC Ratio	Threshold T (PC $< / > T$)
Target-Based				
TB-1	NA	5,082	0.38	> 6
TB-2	NA	5,237	0.39	> 6
TB-3	NA	10,465	0.82	> 6
TB-4	NA	10,624	0.73	> 6
TB-5	NA	12,612	0.16	> 6
TB-6	NA	13,093	0.79	> 6
TB-7	NA	14,605	0.33	> 6
ADME-T				
A-CYPa	CYP3A4	12,875	0.79	< 5
A-CYPb	CYP2C9 (I)	12,876	0.84	< 5
A-CYPc	CYP2C9 (II)	14,062	0.30	< 5
A-PM	Permeability	16,511	0.15	> 1
A-SOL	Solubility	47,607	0.48	> 2
A-hERG	Toxicity	67,687	0.70	< 5
A-logD	Lipophilicity	88,114	0.58	> 3
A-MS	Metabolic Stability	92,161	0.36	< 1

Model Tuning. The HP tuning for the two baseline estimators, RFs and MLPs, was performed using an exhaustive grid search. The exact parameter space search is described in Table A.1 of the Appendix. The BCE loss was calculated to compare the model performance on a validation set. Since assay data often exhibit distributional changes over time, the temporal splitting approach described in Figure 3.1 was also used to obtain validation sets that reflect this progression. This allowed the selection of HPs that perform best under these realistic conditions.

Baseline Models. All architectures used in this work stem from a RF or a MLP. Both approaches are commonly used in research addressing uncertainty estimation in QSAR modeling (Mervin et al., 2021b; Dutschmann et al., 2023; Mervin et al., 2020). The RF models were generated using scikit-learn

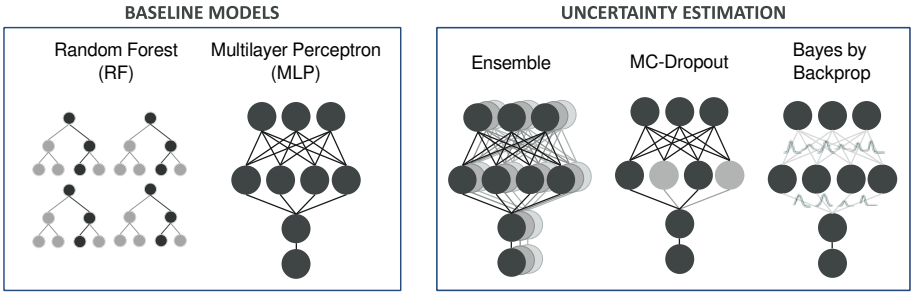

Figure 3.3: **Overview of the classification models.** The architectures of the baseline models and train-time UQ methods compared in this study are shown. All models were trained in a single-task manner. The HPs of the baselines, RF and MLP, were tuned in an extensive grid search. The baseline MLP was used as the basis for the three UQ methods, Ensembling, MC-Dropout, and a BBB.

(Pedregosa et al., 2011). During HP tuning, the maximum depth of the trees and the required number of estimators of each assay and temporal setting were individually tuned using the validation BCE loss. Probability-like outputs were generated from the ratio of decision trees in the RF that classified a test instance as active. The MLP models were trained using PyTorch (Paszke et al., 2019) with the BCE loss function. Section 2.4 gives a theoretical introduction to MLP architecture and training. The network architecture was optimized for the number of hidden units, the number of hidden layers, and the dropout rate. Additionally, the learning rate and scaling factor of a ReduceOnPlateu learning rate scheduler were also optimized. The models were trained using early stopping on the validation loss. Adam was used as an optimization algorithm (Kingma, 2014) to train the NNs. Probability-like scores were obtained by applying a sigmoid function to the output of the MLP.

The baseline models were further modulated to generate more sophisticated uncertainty estimation methods. Train-time UQ methods were trained using the MLP as the foundation. Furthermore, post hoc probability calibration methods were applied to selected models. A detailed description of the train-time uncertainty estimation and post hoc calibration methods can be found below.

Train-time Uncertainty Quantification. In this work, train-time UQ approaches refer to models that estimate uncertainty during training by accounting for model variance. We compare two ensemble-based techniques that retrieve

samples from the posterior distribution of the parameters without directly applying the Bayesian framework: MLP ensembles, which we refer to as Ensembles and are generated from 25 individually trained base estimators, and MC-Dropout models (Gal and Ghahramani, 2016), which included aggregating 400 forward passes. For a theoretical discussion of these two methods, we refer to Section 2.6. Uncertainty estimates are generated by aggregating the predictions of N base estimators $f_i(x)$ to obtain a point estimate $\hat{y}$:

$$\hat{y} = \frac{1}{N} \sum_{i}^{N} f_i(x) \tag{3.1}$$

We also include one full Bayesian NN trained with the BBB approach (Blundell et al., 2015). The BBB approach is a variational approximation approach that aims to optimize a simple surrogate function to approximate the posterior distribution of the model parameters. Section 2.5.2 provides theoretical insight into this method. Uncertainty estimates are generated by obtaining predictions from 400 parameter samples, which are subsequently averaged.

Post hoc probability calibration. Post hoc calibration methods apply a post-processing step to the scores of the classifier. Thus, in contrast to train-time UQ approaches, they only apply a calibrating correction step to the probability-like outputs of a binary classifier rather than intrinsically estimating the uncertainty. Two post hoc probability calibration techniques were fitted to each model using the validation set. These approaches included PS (Platt, 1999) and VA predictors (Vovk and Petej, 2014). The suffixes "+PS" and "+VA" indicate models calibrated with PS or VA predictors, respectively. For instance, the calibrated Ensemble model was labeled "Ensemble+PS" or "Ensemble+VA". Section 2.9 provides the theoretical background for post hoc calibration models.

Experiments

The first part of this study focuses on the data characteristics resulting from the temporal splitting strategy. We studied the shift in label and descriptor space over time, mainly concentrating on the differences between TB and ADME-T assays. The shift in label space was assessed by comparing the PC ratios in each temporal fold. Shifts in the descriptor space were quantified using the maximum mean discrepancy (MMD) (Gretton et al., 2012), which provides a kernel-based estimate for the distance between the distributions of two datasets. MMD has been successfully applied to different tasks, including the detection of distribution shift (Rabanser et al., 2019; Ouyang and Key, 2021), adversarial

sample detection (Gao et al., 2021), and unsupervised domain adaptation (Yan et al., 2017; Zhu et al., 2020). Furthermore, MMD was shown to be applicable to a wide range of problems in molecular biology (Borgwardt et al., 2006). Lee et al. (2023) used MMD on graph representations of chemical structures to score molecules obtained from a generative model.

In this work, we compute the MMD to detect distribution shifts between the ECFP spaces of the training datasets $X := \{\boldsymbol{x_1}, ..., \boldsymbol{x_M}\}$ and the test datasets $Z := \{\boldsymbol{z_1}, ..., \boldsymbol{z_N}\}$ which are distributed according to $P(X)$ and $Q(Z)$.

$$MMD(P,Q) = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} k(\boldsymbol{x_i}, \boldsymbol{x_j}) + \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k(\boldsymbol{z_i}, \boldsymbol{z_j}) - \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} k(\boldsymbol{x_i}, \boldsymbol{z_j}). \tag{3.2}$$

When using the Tanimoto coefficient (Holliday et al., 2002) as kernel k, the MMD lies between 0 and 1, with 0 indicating no shared features and 1 indicating no differences between the compounds in the datasets. To investigate the overlap of chemical scaffolds in the training and test datasets, Murcko scaffolds of the chemical structures were retrieved using RDKit (Landrum, 2006). The proportion of test set scaffolds that overlapped with those in the training set was determined.

The second part of the study aims to determine how well different UQ methods estimate the probability that compounds have a certain desirable feature, such as being active on a TB assay or inactive on an ADME-T toxicity assay. To improve reproducibility, ten repetitions were generated for each method using random initialization. The reliability of the models was assessed by calculating various performance metrics on the test set predictions, as described below. The metrics were calculated for each repetition and subsequently aggregated to obtain the mean and SD for each score and model across all datasets. For each assay and temporal setting, a two-sided, independent t-test was used to assess whether the difference between the best model and any other model was statistically significant.

To compare model performance, the area under the receiver operating characteristic (ROC) curve (AUC) ([↑]) (Hanley and McNeil, 1982), the BCE loss ([↓]), and the ACE ([↓]) (Nixon et al., 2019) were calculated using the test set predictions. Note that the ACE is an improper score (Gneiting and Raftery, 2007; Bröcker, 2009), so a perfect calibration error of 0 does not

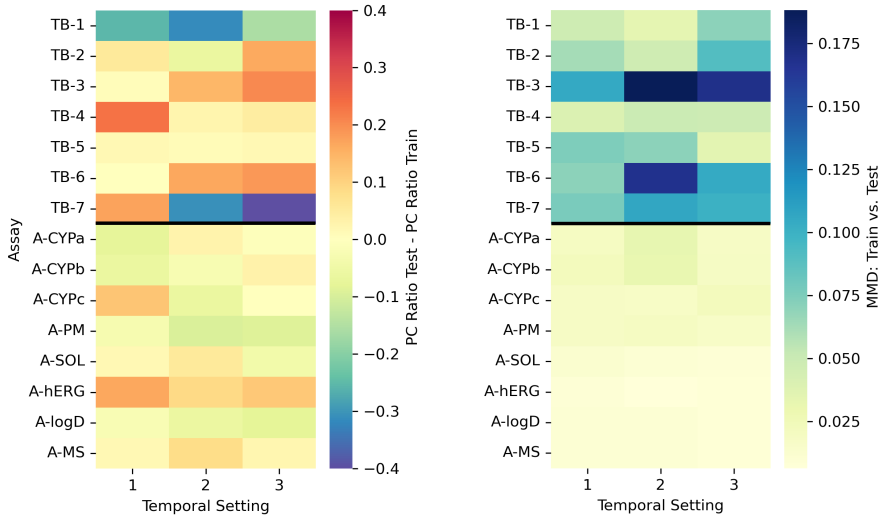

Figure 3.4: **Quantification of the distribution shifts between the training and test datasets over time.** The shift in label space and in the descriptor space is illustrated for each temporal setting, using the data of 1, 2, or 3 temporal folds for training. Results are shown for each assay. The left panel shows the shift in label space in terms of the difference in ratios of the PC between the training and test datasets. The right panel shows the MMD in the descriptor space between the training and test datasets for each temporal setting and assay, quantifying shifts in descriptor space.

automatically correspond to the best model. Thus, the ACE was always evaluated in combination with the BCE for a more comprehensive analysis. We refer to Chapter 2.2.2 for the background theory of proper scores and calibration errors.

3.2.2 Results and Discussion

We divide the results of this study into two consecutive parts to investigate the efficacy of UQ and probability calibration methods using real-world temporal data. The first part addresses the properties of the data under distribution shifts in the label and descriptor space. In the second section, we compare the probability calibration of various UQ methods and set the results in the context of the underlying distribution shift resulting from the temporal split.

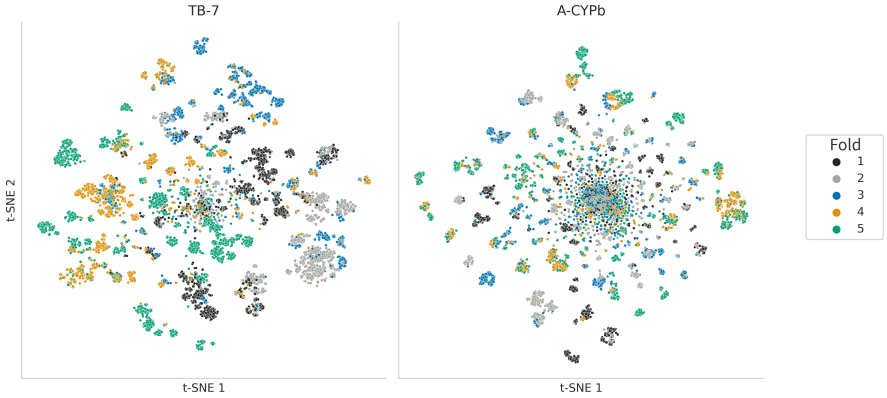

Figure 3.5: **T-SNE plots of the ECFP space.** T-SNE plots of the ECFP space are shown for one example of each assay category to illustrate how the explored chemical space changes over time. Compounds are colored according to the time span, e.g., the temporal fold, to which they were assigned. The t-SNE plots of the remaining TB and ADME-T assays are shown in Figure A.1 and A.2 in the Appendix.

Distribution Shifts over Time

Shift in Label Space. The ratios of the PC in different temporal folds of an assay were compared to evaluate shifts in the label space. The ratio of compounds belonging to the PC varies across temporal settings. The magnitude of this difference is dependent on the assay, as shown in the bottom right panel of Figure 3.2. Some assays exhibit larger differences, like TB-7, while for others, such as A-PM, the ratio is more stable over time. To evaluate the distribution shift in label space between training and test data, the difference in the ratios of compounds in the PC was calculated. The left panel in Figure 3.4 illustrates these differences between the training and the test set for each assay. The ratio of compounds belonging to the PC in each temporal fold is listed for all assays in Table A.2 of the Appendix. We assessed all three temporal settings, using one, two, or three folds as training data. The second consecutive fold after the training folds was considered the test set. We refer to Figure 3.1 for a more comprehensive explanation of the different temporal settings. The left heatmap of Figure 3.4 shows that TB assays evolve differently in label space over time than ADME-T assays. Generally, the differences in PC ratios between the training and test sets are smaller in ADME-T assays, while the more extreme values in TB assays indicate larger shifts in label space. Recall that the TB category includes project-based assays, which aim to find modulators for a

specific target of interest. Therefore, a plausible explanation for the larger shifts in label space could be that various chemical series are tested in search of promising compounds. These series may differ in their abilities to modulate a target, leading to changing ratios of preferred compounds over time. Some assays show an enrichment of the PC over time, as observed in TB-3 and TB-6. However, this pattern was not observed in all assays, and the TB-1 and TB-7 assays even show opposite tendencies. Two datasets exhibited high shifts in label and small shifts in descriptor space, which might indicate the presence of activity cliffs. These two assays included the second temporal setting of TB-1 and the first temporal setting of TB-4. Note that label shifts might also result from chemical series with activity values close to the classification threshold. This could lead to similar chemical compounds labeled differently, despite the absence of an actual activity cliff. A detailed discussion of the shifts in descriptor space can be found in the next paragraph.

Similar to the TB assays, the more stable ratios in the ADME-T assays can also be attributed to the nature of this assay category. These assays are not specific to individual projects and are used to evaluate the pharmacokinetic and toxicity profiles of compounds. Based on the results above, it is highly questionable whether the iid assumption for these assays remains valid over time, particularly in the TB category. This category includes some challenging assays, such as TB-3 and TB-7, which exhibit significant label shifts.

Shift in Descriptor Space. The shifts in ECFP space are visualized using two-dimensional t-SNE plots to reveal patterns and clusters in the dataset. The t-SNE plots for TB-7 and A-CYPb are shown in Figure 3.5, while the plots for the remaining assays can be found in Figure A.1 and A.2 in the Appendix. Figure 3.5 reveals a clear pattern in the TB-7 assay, which indicates a shift in the chemical space over the assay history. Furthermore, chemically similar compounds tend to be assigned to the temporal fold as indicated by the color purity in various clusters. In contrast, the t-SNE plot of the ADME-T assay does not show a clear pattern in clusters and color gradients. To quantify the shift in descriptor space, the MMD was calculated between the training and test sets (Figure 3.4). The observed tendencies are similar to the patterns reported for the label shifts.

To conclude, the TB assays exhibit larger shifts than the ADME-T assays in both label and descriptor space. These results can again be explained by the different characteristics of the two assay categories, resulting in distinct developments through the descriptor space over time. To find promising compounds in the TB assays, various chemical series are usually screened, containing chemically similar compounds. In more detail, the decision on which compounds will be

tested in a TB assay is usually made by an agent. The agent, often a chemist, bases this decision on the information obtained from previous experiments and chooses the structures of compounds that will be tested next. As a result, temporal variations are observed in those assays, driven by the interventions of that agent, which lead to modifications in the chemical structures.

Interestingly, when comparing the ratio of scaffolds in the test set shared with the training set, no apparent differences in the scaffold overlap between the TB and ADME-T assays could be detected (Table A.3 in the Appendix). This means that the difference in distribution shifts between the ADME-T and the TB assays is not a result of the scaffold overlap between the training and test sets. In conclusion, the shifts in descriptor space are more pronounced in TB assays, while those in ADME-T assays are comparatively small. Similar to the shifts in label space, the iid assumption might not be appropriate, particularly in the TB assays.

Probability Calibration Study

We assessed the performance of various uncertainty estimation methods in three experiments, focusing on the probability calibration of the models. Throughout this part of the study, the model performance is assessed separately for TB and ADME-T assays. The first experiment compares the baseline methods and the UQ approaches limited to the third temporal setting, in which three folds were used as training datasets. The second section investigates the change in model performance over time by comparing the three temporal settings. Moreover, the results will be linked to the assay-specific conclusions about the label and data shift drawn in Section 3.2.2. The third experiment concentrates on the potential of post hoc probability calibration methods in the context of distribution shifts between the calibration and test sets. The results of the majority of models are presented. The numerical results of all methods are listed in Table A.1.4 in the Appendix.

Comparison of Uncertainty Estimation Methods. We compared the predictive performance of RF, MLP, Ensemble, MC-Dropout, and BBB models in terms of AUC, BCE, and ACE. For a straightforward comparison, we limit the reported results to the third temporal setting, in which three folds were used for model training. The data assigned to the last fold was used as a test set. The AUC values of the models are listed in Table 3.3. The AUC results for the TB assays show that the MLPs, as well as the non-Bayesian uncertainty estimation methods, outperform the RFs and BBBs on most datasets. In more detail, the Ensemble model is always among the best approaches. The MLP and the

Table 3.3: **Summary of AUC results for the third temporal setting.** AUC results for the baselines (RF and MLP) and train-time UQ models, Ensembles, MC-Dropout, and BBB) are reported. The models were trained with compounds from three temporal folds. The mean and SD of 10 model repetitions are shown. The best-performing approach and those not significantly different from it, as determined by a two-sided t-test, are highlighted in bold.

	RF	MLP	Ensemble	MC-Dropout	BBB
Target-Based					
TB-1	0.437 ± 0.025	**0.586 ± 0.065**	**0.614 ± 0.006**	**0.592 ± 0.061**	0.502 ± 0.024
TB-2	0.778 ± 0.018	**0.793 ± 0.01**	**0.795 ± 0.002**	**0.791 ± 0.01**	0.289 ± 0.026
TB-3	0.708 ± 0.023	**0.765 ± 0.009**	**0.768 ± 0.001**	0.761 ± 0.009	0.73 ± 0.002
TB-4	0.909 ± 0.027	0.95 ± 0.007	**0.956 ± 0.0**	0.952 ± 0.003	**0.957 ± 0.001**
TB-5	0.676 ± 0.028	**0.896 ± 0.008**	**0.9 ± 0.001**	**0.897 ± 0.008**	0.78 ± 0.171
TB-6	0.641 ± 0.06	**0.768 ± 0.007**	**0.771 ± 0.001**	**0.768 ± 0.007**	0.701 ± 0.007
TB-7	0.533 ± 0.086	**0.71 ± 0.026**	**0.718 ± 0.004**	**0.718 ± 0.03**	0.479 ± 0.014
ADME-T					
A-CYPa	**0.675 ± 0.015**	0.625 ± 0.013	0.627 ± 0.002	0.625 ± 0.013	0.622 ± 0.016
A-CYPb	0.581 ± 0.012	0.644 ± 0.005	0.648 ± 0.001	0.643 ± 0.006	**0.661 ± 0.001**
A-CYPc	0.631 ± 0.024	0.714 ± 0.003	0.714 ± 0.0	0.715 ± 0.003	**0.734 ± 0.003**
A-PM	0.613 ± 0.031	**0.769 ± 0.004**	**0.77 ± 0.001**	0.765 ± 0.004	**0.784 ± 0.023**
A-SOL	0.692 ± 0.008	0.78 ± 0.011	**0.792 ± 0.002**	0.779 ± 0.013	0.511 ± 0.011
A-hERG	0.632 ± 0.009	0.729 ± 0.005	**0.735 ± 0.001**	0.729 ± 0.005	0.652 ± 0.003
A-logD	0.641 ± 0.015	0.833 ± 0.003	**0.839 ± 0.002**	0.834 ± 0.003	0.828 ± 0.004
A-MS	0.694 ± 0.006	0.71 ± 0.01	0.716 ± 0.003	0.713 ± 0.006	**0.745 ± 0.007**

MC-Dropout models retrieve results that are not statistically different from the best model in 6 and 5 out of 7 datasets. The BBB is the best model for assay TB-4, while the RF approach is consistently outperformed. In contrast, the results for the ADME-T assays show that either the Ensembles, or the BBB models, or both outperform the other approaches in 7 out of 8 assays. The remaining assay is A-CYPa, for which the RF approach achieves the best result.

The model calibration is analyzed using the ACE and the BCE scores. The results are illustrated in Figure 3.6. The analysis of the BCE and ACE values reveals trends similar to those observed in the AUC scores. The results of the models trained on TB assays show that the Ensemble approach is among the best-performing approaches for all datasets, except for TB-1, for which MC-Dropout performs best in terms of ACE. The baseline MLP matches the performance of Ensemble in 4 out of 7 times in terms of BCE and in 5 out of 7 assays in terms of ACE. Furthermore, the models perform overall worse on TB-1 and TB-2 in terms of BCE. A reason for this result could be the small dataset size of these two assays, which might lead to overfitting and, therefore, poorer calibration of the models. Furthermore, both assays exhibit comparatively large shifts in label and descriptor space, as shown in Figure 3.4, which introduces additional difficulties in generalizing well over time. The results of the models

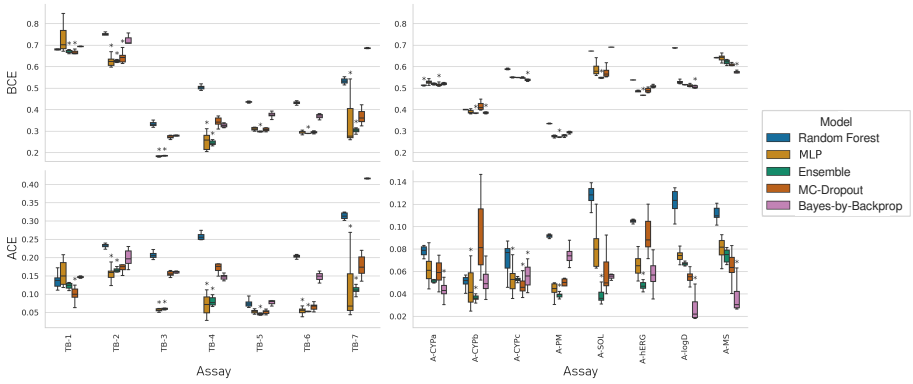

Figure 3.6: **Summary of BCE and ACE scores for the third temporal setting.** The first column shows the results for TB assays, while the second one reports the performance of models trained on ADME-T assays. BCE scores are plotted in the first row, and ACE scores in the second row. Results for the baselines (RF and MLP) and train-time UQ models (Ensemble, MC-Dropout, and BBB) are reported. The models were trained with compounds from three temporal folds. The results of 10 model repetitions were aggregated. For each assay, the best-performing approach and those not significantly different from it, as determined by a two-sided t-test, are marked with an asterisk.

trained on the ADME-T data demonstrate the superiority of the Ensemble and BBB methods. More specifically, in all assays, either Ensemble or the BBB performs best with regard to BCE and ACE. A-CYPa is the only exception for which RF and MC-Dropout perform best. The Ensemble and BBB models achieve the best results across both metrics in the same number of assays, namely in 4 out of 8 cases.

It is surprising that the more sophisticated uncertainty estimation methods improve the probability calibration of the baselines trained on ADME-T assays but fail to do so for models trained on TB category datasets. The Ensemble and the BBB approaches account for epistemic uncertainty by including model variance in their predictions. The Ensemble approach retrieves model samples representing different local minima of the loss surface, while the BBB treats the NN weights as probability distributions. Koh et al. (2021) showed that the performance of a classifier can degrade significantly when there is a distribution shift between the source and target domains. Moreover, Garg et al. (2020) reported specifically for the presence of label shift that a classifier that is ideal for the source domain might no longer be ideal for the target domain. In general, ensemble approaches have been shown to outperform other uncertainty

estimation approaches, including approximate Bayesian NNs, in terms of predictive accuracy and calibration without (Gustafsson et al., 2020) and under distribution shift in the descriptor space (Ovadia et al., 2019; Mehrtens et al., 2023). These conclusions are supported by the results in this study, which show that Ensemble performs better than other uncertainty estimation methods under distribution shift.

However, despite being the best uncertainty estimation approach, the Ensemble models rarely outperform the baseline MLP in the presence of distribution shift. An additional reason for the failure of the uncertainty estimation methods to generate better uncertainty estimates is that the baseline could be their inability to handle the shifts in label space well. This conclusion might be transferable to model calibration, thus explaining the difference between architectures trained on TB and ADME-T assays to produce better-calibrated probabilities. A large study that compared uncertainty estimation approaches used for regression models trained on the same dataset also reported that the Ensemble and BBB approaches outperform other common uncertainty estimation approaches for regression (Svensson et al., 2025). In contrast to the classification setting, the uncertainty estimates of baselines trained with TB assays could also be improved in the regression study. This observation could indicate that uncertainty estimation models for regression tasks are less sensitive to shifts in the label space than approaches for classification. A reason for this observation could be that the model can access the actual values of the measurements in the regression setting, which might attenuate the shift in the target space. For example, strongly inactive and weakly inactive compounds exhibit different target values for regression models, while this information is lost in classification tasks due to the application of binary classification thresholds.

Uncertainty Estimation over Time. We assessed the quality of the uncertainty estimates over time by comparing the model performance in all three temporal settings. The results are displayed in Figure 3.7. The plots show that the model performance in one time span, e.g., the first temporal fold, rarely allows conclusions about the performance of the same approach at another point in time. In 3 out of 7 TB assays, a single model is always among the models with the best BCE score in all three temporal settings. These approaches include MC-Dropout for TB-1, MLP for TB-6, and Ensemble for TB-7. Regarding the ACE scores, the Ensemble is consistently among the best methods for TB-2, and the MLP is among the best for TB-3. The BCE results of the ADME-T assays show consistent results over time for 4 assays, including the RFs for A-CYPa and the Ensemble approach for A-PM, A-SOL, and A-hERG. The ACE results of the ADME-T assays reveal a consistent model performance in 2 out of 8 assays, namely the MLPs for A-PM and the Ensemble for A-hERG. In general,

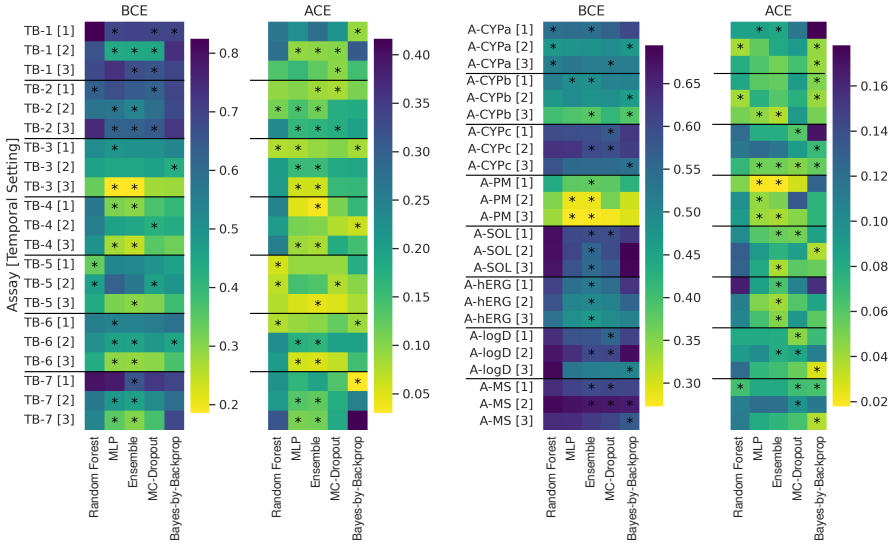

Figure 3.7: **Summary of BCE and ACE scores across all temporal settings.** The first two columns show the ACE and BCE scores for TB assays, while the last two report the performance of models trained on ADME-T assays. The temporal setting is indicated in brackets after the assay abbreviation. Results for the baselines (RF and MLP) and train-time UQ models (Ensemble, MC-Dropout, and BBB) are reported. The models were trained with compounds from three temporal folds. Averages over 10 model repetitions are shown. For each assay, the best-performing approach and those not significantly different from it, as determined by a two-sided t-test, are marked with an asterisk.

model performance in terms of ACE is slightly less stable over time than in terms of BCE. Interestingly, Svensson et al. (2025) applied regression modeling techniques to the same data and reported more consistent model performance for the ADME-T assays. More details about this study can be found in Chapter 3.4.

Plotting the model performance on all temporal settings confirms the conclusions drawn in earlier sections. The MLP and Ensemble methods obtain the best results in terms of both BCE and ACE for the TB assays. Both are among the best-performing approaches in at least one temporal setting in all assays, except TB-4, where MLP is not among the best models in any setting. Interestingly, regarding the BCE results, the MLP is as often among the best models as the Ensembles, namely for 13 out of the 21 settings. The ACE scores reveal a similar result. MLP is among the best methods for 11, and Ensemble for 13

out of 21 settings. All other approaches trained on the TB assays are much less often among the best-performing models.

As described earlier, TB-1[2] and TB-4[1] exhibited large shifts in label space and small shifts in descriptor space, which might indicate the presence of activity cliffs. Interestingly, the performance of models trained on these datasets did not differ considerably from the results of other TB assays. Thus, if the label shift was large, model performance was not affected significantly by the presence of shifts in descriptor space. A reason for this result could be the overall large shifts observed in TB assays, which makes the prediction tasks generally harder, diminishing the impact of activity cliffs on model performance. Furthermore, the combination of high label and low descriptor space shifts could also be an artifact of chemical series exhibiting activity values close to the classification threshold rather than actual activity cliffs.

The results for the ADME-T assays also support the conclusions from the previous experiment. Concerning the ACE scores, the Ensemble and BBBs outperform all other models, with Ensemble being among the best methods for 13 and BBB for 11 out of 24 settings. The MLPs and the MC-Dropouts are among the best-calibrated methods in 5 and 7 settings, respectively. The Ensembles outperform the other approaches in terms of BCE, obtaining significant results in 13 settings. The MC-Dropout models are among the best methods in 8, and BBB in 7 out of 24 settings.

In conclusion, the Ensemble approach generates the best-performing models for most ADME-T assays. Considering the high computational resources required to train these Ensembles, another good choice are BBBs. This approach is much more time- and resource-effective and generates well-calibrated models for many ADME-T datasets. Note that a fixed variance was chosen for the Gaussian prior and that tuning this HP could even improve the performance of the BBB models. Given that Ensemble models always lead to more underconfident predictions (Rahaman and Thiery, 2021), the good performance of these methods indicates overfitting of the baselines. The Ensembles and BBB models counteract this behavior for the ADME-T assays but not for datasets from the TB category. The ACE values for TB assays are higher than for the ADME-T assays, as illustrated in Figure 3.6, indicating worse model calibration. This result implies room for improvement in probability calibration and eliminates good baseline model calibration as a reason for the inability of Ensemble and BBB models to improve the ACE of the MLP. An alternative reason for the failure of Ensembles and BBB could be the shift in label space, which is considerably larger for TB assays.

Based on these results, it is questionable whether the costly generation of Ensembles for improved uncertainty estimation is justified for datasets with

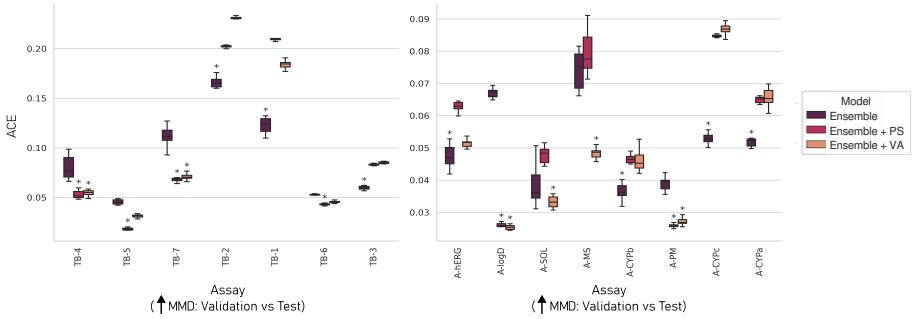

Figure 3.8: **Summary of ACE scores of post hoc probability calibration approaches using the third temporal setting.** The left panel shows the ACE scores of models trained on TB assays, while the right panel reports the ACE performance of ADME-T models. The assays in each panel are ordered according to increasing distribution shifts in descriptor space between the calibration and the training set. The distribution shift is determined by the MMD between the two datasets using the Tanimoto coefficient on the ECFP space. Results for the Ensembles, the Platt-scaled Ensembles (Ensemble+PS), and the Ensembles calibrated with a VA predictor (Ensemble+VA) are reported. The models were trained with compounds from temporal folds. The results of 10 model repetitions were aggregated. For each assay, the best-performing approach and those not significantly different from it, as determined by a two-sided t-test, are marked with an asterisk.

large distribution shifts, such as the TB data. As discussed earlier, this shift might be difficult to handle for the UQ methods, resulting in the inability of UQ methods to improve the probability calibration.

Post hoc Probability Calibration. The effects of two post hoc calibration methods, PS and VA predictors, were assessed. Furthermore, the distribution shift between calibration and test sets was taken into account by obtaining the MMD between the two datasets. The MMD between the calibration and test datasets is reported in Table A.4 in the Appendix. For the sake of clarity, we only include the results of the third temporal setting for the Ensemble approach, which was reported to be one of the best-performing models in previous sections of this study. Only the ACE is reported since the post hoc calibration step includes the application of monotonously increasing functions, which cannot correct non-monotonous distortions and, therefore, does not change the ranking of the predictions. Hence, these approaches only affect the probability calibration while the AUC scores of the models remain constant.

The results of all calibrated models across all temporal settings and assays can be found in Section A.1.4 in the Appendix.

Figure 3.8 shows the performance of the Platt-scaled Ensembles (Ensemble+PS) and the Ensembles calibrated with VA predictors (Ensemble+VA). The results are shown separately for TB and ADME-T assays, and the assays are ordered according to increasing MMD values. In general, post hoc scaling leads to better results in 4 out of 7 TB assays and in 4 out of 8 ADME-T assays. In 4 out of these 8 cases, both Ensemble+PS and Ensemble+VA are the best models, while in 4 cases, either Ensemble+PS or Ensemble+VA performs best. Both panels in Figure 3.8 show that with increasing MMD between calibration and test set, the calibrating abilities of the post hoc calibration methods decrease. A stronger trend can be detected for the TB assays, for which the shifts are larger than for the ADME-T assays. Nevertheless, the pattern is also visible in the right panel plotting the results for ADME-T, which show no improvements after post hoc calibration for assays with large MMD, like A-CYPc and A-CYPa. The reported trends can also be seen for other approaches, albeit less clearly for some models.

An intuitive explanation for this pattern is that post hoc probability approaches perform better if the calibration set is similar to the test set, and worse if the calibration and test sets are different. Ovadia et al. (2019) showed that post hoc calibration approaches improve model calibration in the iid setting. However, their calibrating abilities degrade as the data shift increases, so that they are ultimately outperformed by train-time UQ methods in the presence of large shifts. These findings are supported by the post hoc calibration results in this study, which report good calibrating properties of the post hoc calibration methods in the case of small shifts, while for larger shifts, they are outperformed by the train-time versions of the methods.

3.3 Probability Calibration with Experimental Errors

This extended abstract makes the following contributions:

- We investigate whether the inclusion of data uncertainty in the form of probabilistic labels improves uncertainty estimation with classification RF models.
- Furthermore, the performance and calibration of the RF and probabilistic random forest (PRF) models with and without post hoc calibration were compared using two different calibration approaches.
- Finally, we use a temporal splitting approach to investigate how the temporal evolution of the test set affects model calibration.

There are various sources of uncertainty in ML as described in Section 2.1.2. A common classification found in literature is the distinction between aleatoric uncertainty, which originates from uncertainty in the data, and epistemic sources, which denote uncertainty originating from deficiencies in the model (Chapter 2.1.2) Experimental errors are a major source of aleatoric uncertainty and can substantially contribute to the overall uncertainty in model predictions. Several studies have demonstrated the presence of experimental variability in public and proprietary datasets (Kramer et al., 2012, 2016; Kalliokoski et al., 2013), affecting the reproducibility of results. Such variability imposes inherent limitations on the reliability of experimental data and on the performance of predictive models (Heid et al., 2023b; Brown et al., 2009).

In binary classification with hard labels, information about how close the measurement of the compound is to the chosen threshold is omitted. Thus, compounds with measured values close to the threshold are treated identically to those that lie far from it, despite the different uncertainties associated with their labels. Furthermore, experimental variability can significantly affect the model performance. Since the decision thresholds for the classification of compounds are often chosen arbitrarily, the assignment of labels might be random, especially in the case of high experimental errors. Hence, UQ models might benefit from labels that take into account uncertainty in the data. For example, Mervin et al. (2021b) reported that using probabilistic instead of hard labels improves the uncertainty estimates of compounds with observed bioactivity values close to the threshold.

Probabilistic random forest (PRF) models (Reis et al., 2018) represent a modification of the established RF algorithm that explicitly accounts for experimental errors and takes into account the distance between a measurement and the threshold. Mervin et al. (2021b) recommend using PRFs instead of RFs when experimental uncertainty in the data is high. Furthermore, the author demonstrated that PRFs provide the greatest improvement over RFs near the midpoint of the probability scale, corresponding to marginal cases close to the binary classification threshold.

In this work, we evaluate the performance of PRFs compared to traditional RF models in terms of probability calibration. For the PRF setup, we construct probabilistic labels as proposed by Mervin et al. (2021b), using the assay-specific experimental error, measured by the variability in control compound measurements, and the assay-specific thresholds. Furthermore, we use temporal splits on proprietary data as described in Chapter 3.1.1 to simulate the evolution of pharmaceutical data with the aim of creating a realistic setup for model evaluation.

3.3.1 Methods

Data

We evaluate single-task classification models on data from ten pharmaceutical assays. The data included a subset of the assays described in Chapter 3.1.1, namely all 7 TB assays and the CYP ADME-T assays. For each assay, the compounds were assigned to two activity classes. Given that the date of each measurement is available, a real temporal split was performed on the data as described in Chapter 3.1.1. The temporal splitting strategy retrieved five temporal folds, corresponding to different time spans of the assay.

Generation of Hard Labels. A detailed description of the binary classification procedure can be found in Section 3.2.1. Hard labels were generated using a pIC50/pEC50 threshold of 6 for assigning active or inactive labels based on the result. This specific threshold was chosen because the models will be deployed in the early stages of the drug discovery pipeline, in which the desired binding affinity of drug candidates is in the micromolar range (10^{-6} molar concentration) corresponding to a pIC50/pEC50 of 6. Note that for the CYP assays, the pIC50 threshold was also set to 6, unlike in earlier described work (Section 3.2), where a lower threshold was used.

Generation of Probabilistic Labels. Probabilistic labels were obtained by applying the thresholds used for hard label generation, and the assay-specific measurement error, corresponding to the SD of the control compound measurements. In detail, a normal distribution $X \sim \mathcal{N}(\mu, \sigma^2)$ was generated, where μ corresponded to the chosen threshold and σ^2 to the SD of the control compound of the respective assay. In this step, the control compound corresponded to the compound with the most measurements. Subsequently, the CDF of these assay-specific distributions was used to obtain the probabilistic label (Mervin et al., 2021b). Figure 3.9 shows the SD of the control compound as well as the available number of measurements to calculate the SD for every assay.

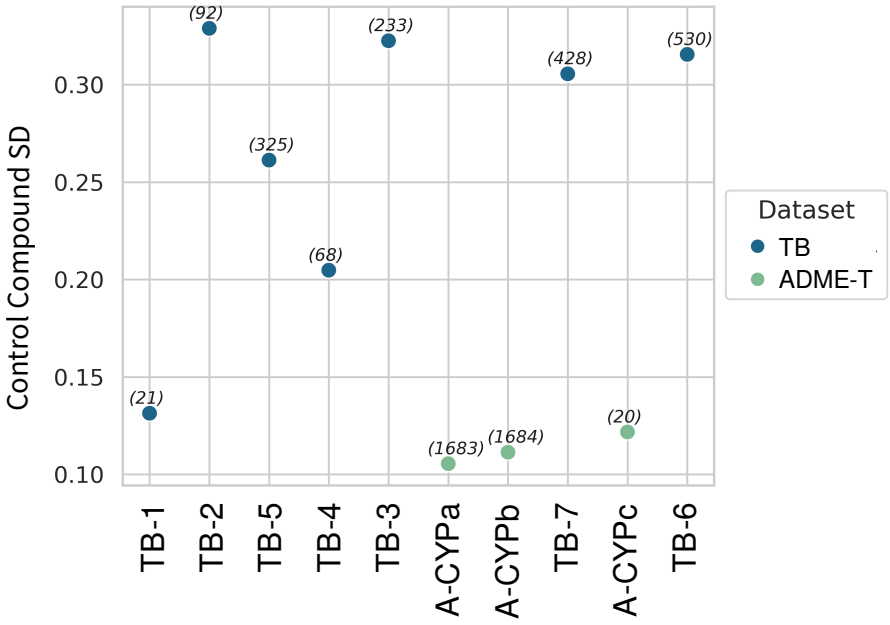

Figure 3.9: **SD of the control compounds.** The SD and the number of measurements (in brackets) of the control compound for every assay are shown.

Model Generation

Standardized SMILES were obtained using the method described in the MELLODDY-TUNER (Heyndrickx et al., 2023) package, and ECFPs of size 1024 and radius 2 were generated with RDKit (Landrum, 2006). RF models were generated using scikit-learn. The maximum depth of the trees and the required number of estimators were tuned using a validation dataset as described in Section 3.2.1. Probability-like outputs were generated by taking the ratio of decision trees in an RF that voted for a specific test instance to be active. Furthermore, PRFs (Reis et al., 2018) were generated using probabilistic labels as ground truth. A detailed description of the PRF training procedure can be found in Mervin et al. (2021b).

Post hoc probability calibration techniques fit a calibration model to the raw scores of a classifier using a separate calibration dataset. In this study, we use the validation dataset for this step, corresponding to the next temporal fold after the training folds.(Figure 3.1). Two uncertainty calibration approaches were used, namely PS (Platt, 1999) and VA predictors (Vovk and Petej, 2014). A detailed methodological background of the post hoc calibration approaches can be found in Section 2.9.

3.3.2 Results

Incorporation of Aleatoric Uncertainty using Measurement Errors

Table 3.4 summarizes the BCE loss and the ACE (Nixon et al., 2019) for five model repeats of all model types trained on two example datasets, namely the A-CYPc and TB-5 assays. The first three folds were used for model training, while the last fold was used for testing, corresponding to the third temporal setting as illustrated in Figure 3.1. Using probabilistic labels instead of hard labels improves the calibration error and the BCE loss of the RF and RF+PS models trained on TB-5 assay data. Models for the A-CYPc assay do not show any improvements when incorporating aleatoric error. This result could be explained by the difference in SDs shown in Figure 3.9, which are used for generating the probabilistic labels. Given that the measurement error of the A-CYPc assay is smaller compared to the TB-5 assay, the normal distribution used for generating the probabilistic labels is narrower, resulting in probabilistic labels that are more similar to the hard labels, thus leading to similar results of RF and PRF models.

Table 3.4: **Overview of RF model performance based on two example assays.** Averages over five model repeats are shown. The best results for each metric are marked in bold, while not significantly worse scores are indicated in italics.

	A-CYPc		**TB-5**	
Method	BCE (↓)	ACE (↓)	BCE (↓)	ACE (↓)
Hard Labels				
RF	*0.187 ± 0.005*	0.032 ± 0.001	0.312 ± 0.009	0.192 ± 0.008
RF+PS	**0.182 ± 0.004**	**0.029 ± 0.002**	0.235 ± 0.007	0.117 ± 0.006
RF+VA	*0.183 ± 0.002*	0.037 ± 0.001	**0.211 ± 0.007**	**0.089 ± 0.006**
Probabilistic Labels				
PRF	*0.181 ± 0.002*	*0.032 ± 0.002*	0.307 ± 0.01	0.187 ± 0.008
PRF+PS	**0.181 ± 0.001**	**0.03 ± 0.001**	0.229 ± 0.005	0.112 ± 0.004
PRF+VA	0.185 ± 0.002	0.038 ± 0.002	**0.212 ± 0.006**	**0.092 ± 0.006**

The post hoc calibration methods improve the BCE loss and ACE scores of TB-5 models, with RF+VA performing best in terms of both metrics, with a BCE and ACE of 0.211 ± 0.007 and 0.089 ± 0.006, respectively. The results for the A-CYPc assay show that in terms of ACE the RF+PS model performs slightly better than the uncalibrated RF model, while the PRF models did not improve after calibration.

In general, the control compounds of the ADME-T assays exhibit smaller SDs than those of the TB assays, as illustrated in Figure 3.9. The results of the assays omitted from Table 3.4 reveal that using probabilistic labels generally leads to better BCE scores for TB assays. In contrast, such clear improvements can not be observed for the ADME-T assays. This could be a result of the differences mentioned above in SDs of the control compounds between the assay categories. However, there are also exceptions from this trend, such as the TB-1 assay, which does not show improvements when including probabilistic labels, despite the large SD of its control compound. Hence, we conclude that it is required to look at the model performance on the individual assay to find the best calibration method for that specific dataset. For all assays, the same model performs best in terms of BCE scores when comparing models trained with hard labels versus probabilistic labels. This is also true in terms of the ACE results, except for the TB-1 assay, for which the RF model performs best for hard labels and the VA-calibrated model is best for probabilistic labels. However, the difference between PRF and PRF+VA is not significant. A more elaborate study is required to understand the effect of probabilistic labels on probability calibration in detail, which will be the object of our future research but is outside the scope of this abstract.

Probability Calibration across evolving Test Sets

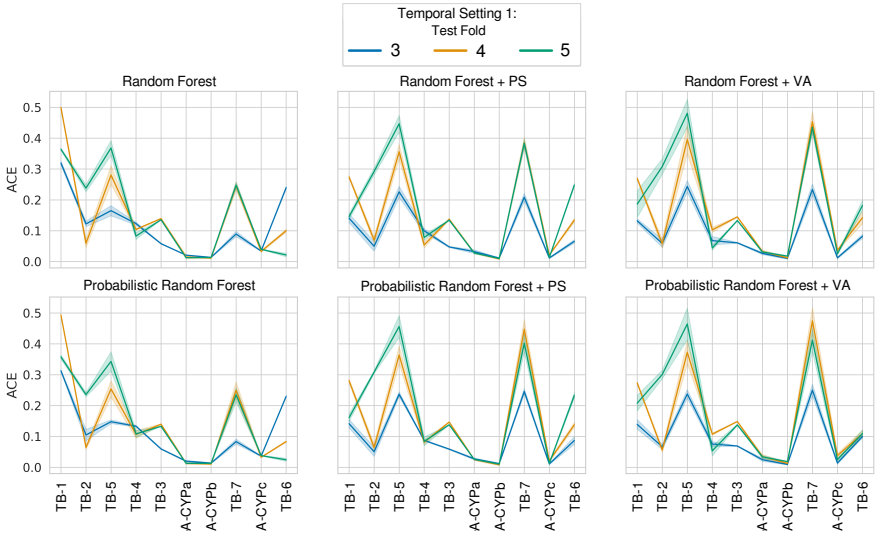

Figure 3.10: **Model calibration over time.** ACE is shown for five model repeats across all assays. The models were trained on one training fold. Test set 3 is closest in time to the training set, whereas test set 5 is furthest away.

Figure 3.10 shows the performance of five model repeats of different RF models across all ten assays and test sets in terms of ACE. In this experiment, only the first fold was used for model training, corresponding to the first temporal setting as illustrated in Figure 3.1. The three remaining temporal folds (Fold 3, Fold 4, and Fold 5) were used separately to evaluate the model performance on evolving test data. Test set 3 corresponds to temporal fold 3, which is closest in time to the training fold, while test set 5 represents the temporal fold furthest away. Fold 2 was used for model validation and HP tuning.

The ACE for test set 3 is the smallest across all models for the majority of assays, as shown in Figure 3.10, indicating that the models are better calibrated for compounds measured closer in time to the training fold. This pattern can also be observed in some assays when comparing test sets 4 and 5, however, the tendency is not as clearly visible as for test set 3. One of the reasons for the observed behavior could be a distribution shift in training and test data that increases as we progress in time, which is supported by a paper by Ovadia et al. (2019), in which an increasing distribution shift was reported to impair probability calibration.

3.4 Enhancing UQ with Censored Regression Labels

This paper makes the following contributions:

- We derive extended versions of ensemble-based models, Bayesian models, and Gaussian mean-variance estimators capable of learning from additional partial information available in censored regression labels.
- Similarly, we adapt available evaluation methods to compare models trained with and without the additional censored labels.
- Furthermore, we provide a large-scale comparison between the resulting censored regression models in a comprehensive temporal evaluation using internal pharmaceutical assay-based data.
- Finally, we showcase how the resulting model predictions can be used in practical applications to streamline and aid the drug discovery process in a case study.

Computational methods for QSAR modeling often suffer from limited data and sparse experimental observations. Additional information often exists in the form of censored labels that provide thresholds rather than precise values of observations. Censored labels arise when the experiment's measurement range is exceeded, such that the exact value cannot be recorded. For instance, a fixed range of compound concentrations is typically used to test the compound's effect on biochemical processes in an assay. If no response is observed within this range, the experiment may only indicate that the response lies above or below the tested concentrations rather than providing a specific value. This results in a censored label, where the true value is known only to exceed or fall below a certain threshold. While censored labels can be easily included in classification tasks by categorizing observations as active or inactive, integrating them into regression models that predict continuous values is far less trivial. Due to this challenge, censored data has not yet been properly utilized in regression tasks within drug discovery, despite its potential to enhance model accuracy and UQ.

The problem of censored regression has been widely studied in other fields, particularly in survival analysis, where Hollander et al. (1987) demonstrated that the information contained in data decreases after censorship. Despite the reduced information in censored labels, Hüttel et al. (2024) further showed that these labels are crucial in Bayesian active learning due to their contribution to

increased entropy and mutual information between the predicted and true data distributions. Additionally, censored labels have enhanced quantile regression in survival analysis, as demonstrated by Pearce et al. (2022). In the context of drug discovery, limited work has explored ways to incorporate censored labels for regression tasks. Arany et al. (2022) proposed an adaptation of the MSE to account for censored labels by using a one-sided squared loss. However, their approach did not address UQ. This approach was first explored for UQ in an earlier version of our work on a smaller subset of the datasets provided here (Svensson et al., 2024a). To the best of our knowledge, no prior work has yet adapted both the MSE and the more general Gaussian NLL for UQ in predictive modeling for drug discovery. Therefore, further exploration of how censored labels can be effectively integrated into regression models to improve prediction accuracy and uncertainty estimation in drug discovery is needed.

In this work, we address the challenge of effectively utilizing censored regression labels during QSAR modeling in drug discovery, a domain where data is often scarce, and UQ is critical for decision-making. While previous approaches have primarily focused on high-quality data, our work expands the ability of ML models to incorporate partial, yet valuable information provided in censored labels, thereby enhancing the reliability of the uncertainty estimates. To achieve this, we develop and extend existing methods, adapting them to the unique demands of drug discovery tasks. We also refine evaluation techniques to rigorously assess the impact of incorporating censored labels, ensuring that the benefits are demonstrated across relevant metrics.

3.4.1 Methods

The methodologies employed in our study are divided into two parts detailing the data and the modeling approaches. First, we describe the data used in our analysis. Following this, we outline the models and techniques implemented to leverage this data, focusing on how they are adapted to handle the unique challenges presented by censored regression labels.

Data

The analysis in this work was performed on data from 15 internal biological assays as described in Section 3.1.1. The data is categorized into two distinct groups: project-specific TB assays and ADME-T assays, which are cross-project assays related to side effects, such as Absorption, Distribution, Metabolism, Excretion, and Toxicity properties (Heyndrickx et al., 2023). This division allows us to explore the effects of censored labels on uncertainty estimation

using different assay types. The same dataset or parts of it were used in recent studies described in Chapters 3.2 (Friesacher et al., 2025b) and 3.3 (Friesacher et al., 2024), providing a relevant benchmark for our work.

During the data preparation, the measurements of all TB assays and the ADME-T assays measuring CYP and hERG inhibition were transformed to pIC50/pEC50 by taking the negative log base 10 of the measurements after transforming them to molar units. In addition, the measurements of the solubility, permeability, and metabolic stability assays were transformed to log10-scale. The assay measuring lipophilicity was not modified, as it already comprised results on the log scale. Duplicated measurements for molecular compounds in the data were then aggregated using the median of the result. The SD for this aggregation was also used in the study to indicate the experimental error. For each assay, a control compound was available that had been tested many times more than any other compound. The SD of the control compound was used as an indication of the general experimental error of the assay, also presented in Table 3.5. The control compounds were removed before modeling each assay.

Data censoring. Given that the determination of exact experimental results is often connected with additional experimental efforts, a significant proportion of the data is provided with censorship. We follow the definition of censored regression proposed by Hüttel et al. (2024). Formally, if we define the true experimental label $y_n^* \in \mathbb{R}$ of a molecular compound with features $\boldsymbol{x}_n \in \mathbb{R}^d$ of length d from an assay, a censored label provides only an observation of a threshold z_n below or above which the true result lies. In the case of right-censoring, this means that instead of observing y_n^*, we observe $y_n = \min(y_n^*, z_n)$, and for left-censoring, we instead observe $y_n = \max(y_n^*, z_n)$. As such, we introduce a mask m_n which is 1 if $y_n^* > z_n$ and -1 if $y_n^* < z_n$, otherwise $m_n = 0$. As a result, we get the dataset $\mathcal{D}_{\text{censored}} = \{\boldsymbol{x}_n, y_n, m_n\}_{n=1}^N$ for each assay with total number of labels N. In particular, note that the censoring threshold can vary for different labels, which is different from fixed-value censoring where the threshold is fixed (Powell, 1986). Furthermore, we assume that the true distribution and the censoring distribution are conditionally independent given the molecular compound's features, i.e., that $y_n^* \perp z_n | \boldsymbol{x}_n$. Labels that are not censored are henceforth referred to as observed labels. An observed dataset was also prepared for each assay to evaluate if the performance improved by including the censored labels. These datasets contain only the observed labels defined by $\mathcal{D}_{\text{observed}} = \{\boldsymbol{x}_n, y_n, m_n\}_{\forall n: m_n = 0}$.

While prior work mentions the possibility of extending the definition of censored regression to include both right- and left-censoring at once (Arany et al., 2022;

Table 3.5: **Regression models: Data overview.** A detailed overview of the assays used for training regression models, including the proportion of the censored data, the assay size, and the experimental SD derived from a control compound.

Short name	Assay description	Assay size	Censoring <	Censoring >	Control SD
Target-based					
TB-1	N/A	5,082	32%	1%	0.13
TB-2	N/A	5,237	43%	0%	0.24
TB-3	N/A	10,465	0%	0%	0.36
TB-4	N/A	10,624	12%	2%	0.20
TB-5	N/A	12,612	35%	0%	0.32
TB-6	N/A	13,093	0%	0%	0.31
TB-7	N/A	14,605	25%	0%	0.32
ADME-T					
A-CYPa	CYP3A4	12,875	61%	0%	0.11
A-CYPb	CYP2C9 (a)	12,876	63%	0%	0.11
A-CYPc	CYP2C9 (b)	14,062	58%	0%	0.12
A-PM	Permeability	16,511	8%	0%	0.17
A-SOL	Solubility	47,607	5%	6%	0.21
A-hERG	Toxicity	67,687	42%	0%	0.19
A-logD	Lipophilicity	88,114	0%	8%	0.10
A-MS	Metabolic Stability	92,161	8%	6%	0.14

Hüttel et al., 2024), the majority of previous applications focus only on right-censoring. During the aggregation of duplicated measurements in the data preparation for this work, observed labels were always prioritized over censored labels. If no observed labels were available, the compounds were assigned the most common censored threshold among the potential duplicated measurements. Table 3.5 lists the percentage of censored labels for each assay used in this work divided between left- and right-censoring, according to the log-scale of the end-points. Note that while many assays include only left-censored labels, i.e., $y_n^* < z_n$, two assays have a more balanced amount of censored labels, namely the ADME-T assays for solubility and metabolic stability. There is also one of the included assays that only have right-censored labels, namely the ADME-T assay for lipophilicity. Finally, two of the TB assays, TB-3 and TB-6, do not have any censored labels. These two assays are useful in the model comparison to determine how all of the considered models compare when no censored labels are available.

Molecular Representations. All remaining molecular compounds were encoded with RDKit (version 2023.03.3)(Landrum, 2006) from SMILES strings (Weininger, 1988) to ECFP (Morgan, 1965) of size 1024 and radius 2. Other, more advanced ways to encode molecular compounds exist, such as the graph-based ChemProp model (Yang et al., 2019) and the pre-trained language-based CDDD model (Winter et al., 2019). Models based on the resulting embeddings from these NN encoders have been compared and shown improvements in prior work (Lenselink et al., 2017; Hirschfeld et al., 2020; Dutschmann et al., 2023). However, Dutschmann et al. (2023) showed that fingerprints perform best in combination with RF and are close second to CDDD in combination with NNs. Additional prior work has similarly shown that NNs perform better in combination with fingerprints compared to other ML methods (Mayr et al., 2018; Van Tilborg et al., 2022). While we employ the fingerprint representations in this study for simplicity and computational reasons, we strongly encourage considering state-of-the-art, learned representations before deploying our proposed methods in practical applications.

Temporal split. The data of each assay were split into five temporal folds corresponding to different time spans of the assay, as described in Section 3.1.1. By comparing the feature-space and label distributions between the three training folds and their respective validation and test sets, we identified various scenarios of in- versus out-of-distribution cases among the datasets. For the feature-space analysis, we created t-SNE projections of the molecular compounds in each dataset, shown in Figure A.4 in the Appendix. For the label-space analysis, we compared the distributions of all observed labels of each fold as shown in Figure A.3 of the Appendix. The results of this analysis coincide with the conclusions drawn in Section 3.2.2, where the data was used for classification modeling.

Generally, for the ADME-T assays, no clear distribution shift is present in either the feature-space or the label-space over time. On the contrary, for the TB assays, both the feature-space and the label-space shift over time. These observed trends are naturally expected due to the fundamental difference between the two kinds of assays. ADME-T assays include final drug candidates from all kinds of drug discovery projects that stem from a diverse chemical space. TB assays, on the other hand, are more focused on a given chemical space, which can shift over time as the focus of the project changes. This important distinction would not be possible without the temporal split proposed in this work, and the effects of it on model performance as well as calibration of predicted uncertainty are explored during the evaluation of the models.

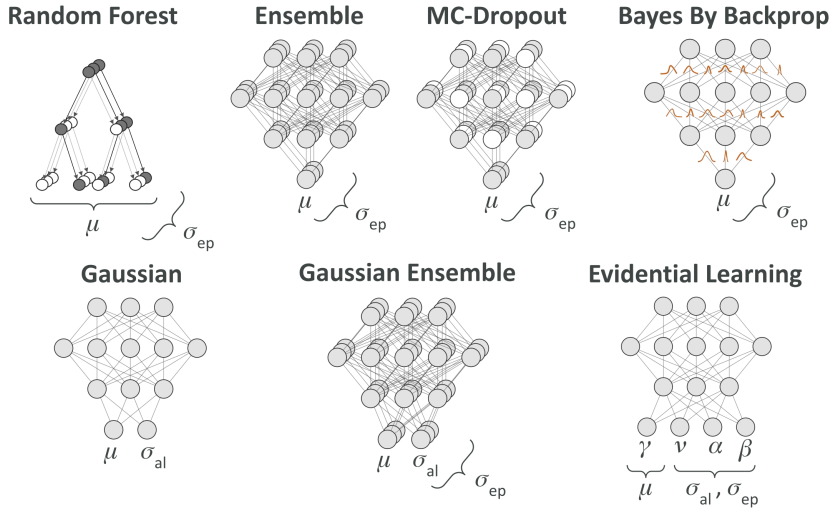

Figure 3.11: **Overview of regression models.** Illustrations of all models used in this study. The top row shows ensemble-based and Bayesian methods for which epistemic uncertainty can be obtained from the SD in sampled predictions. All models in the bottom row produce aleatoric estimates of uncertainty. Additionally, epistemic uncertainty can be derived from the Gaussian Ensemble and the Evidential model.

Model Generation

In this work, we propose five models that extend existing techniques to handle censored labels alongside two baseline models that only learn from observed labels without accounting for censorship. Figure 3.11 gives an overview of all models and how they can estimate epistemic and/or aleatoric uncertainty. Ensemble-based and Bayesian approaches, such as RFs (Sheridan, 2012), ensembles of NNs (Lakshminarayanan et al., 2017), MC-Dropout (Gal and Ghahramani, 2016), and the approximation of a Bayesian NN known as BBB (Blundell et al., 2015) are commonly used to estimate epistemic uncertainty. For aleatoric uncertainty, two primary methods are available: the Gaussian mean-variance estimator (Nix and Weigend, 1994) and evidential deep learning (Amini et al., 2020). By creating an ensemble of the Gaussian model, we can also derive estimates of epistemic uncertainty. Evidential deep learning, on the other hand, simultaneously models four parameters that can be used to directly estimate aleatoric and epistemic uncertainty. We extend all of these models, except the RF and Evidential model, to learn from both observed and

censored data, while the two remaining approaches are used as baselines. A detailed description of how the model selection is performed for each model is provided in Section A.2.2 of the Appendix. While the data used in this work is proprietary and cannot be disclosed, the full methodology is available on GitHub at `https://github.com/MolecularAI/uq4dd`. Our framework is inspired by the design pattern proposed by Hartog et al. (2024).

Ensemble-based approaches. In general, ensemble-based approaches are useful to model epistemic uncertainty by estimating the model variance. An ensemble is defined as a set of K base estimators $f(\boldsymbol{x}_n)$, for which the average of the individual base estimators' predictions is taken as the final prediction by the ensemble and the variance of the predictions as an estimate of the predictive uncertainty as follows,

$$\mu_n = \frac{1}{K}\sum_{k=1}^{K} f_k(\boldsymbol{x}_n), \quad \sigma^2_{\text{ep},n} = \frac{1}{K}\sum_{k=1}^{K}(f_k(\boldsymbol{x}_n) - \mu_n)^2. \tag{3.3}$$

We refer to Chapter 2.6 for more details on ensembles.

One of the simplest and most widely used traditional ML methods for ensemble-based UQ takes the decision trees in a RF as an ensemble (Sheridan, 2012). We use the implementation of decision tree regressors from Scikit-learn (Pedregosa et al., 2011), where each decision tree is trained independently using different sub-samples of the training data. The final prediction is obtained by averaging the predictions of all the trees and the variance constitutes the epistemic uncertainty estimate. Similarly, ensembles can be created by training multiple instances of the same NN architecture with different random weight initialization (Lakshminarayanan et al., 2017). We first consider the NN architecture with a single output, which means that each base estimator makes a prediction such that the final prediction by the ensemble is again the average of the individual predictions, and the variance estimates the epistemic uncertainty (Tetko et al., 2008). We use 50 randomly initialized and independently trained NNs to create this ensemble and hence refer to it as Ensemble. All NNs in this work are implemented and trained with PyTorch (Paszke et al., 2019). The theoretical background for NN training can be found in Chapter 2.4.2.

Furthermore, MC-Dropout is another technique that uses dropout in NNs to approximate Bayesian inference (Gal and Ghahramani, 2016). During MC-Dropout, multiple samples are drawn from the model using dropout during the inference phase (Chapter 2.6). Similar to the MLP ensembles, the variance between these predictions provides an estimate of the epistemic uncertainty. We use the same NN architecture from the Ensemble in our MC-Dropout model,

but train only one model and draw 500 samples from it during the inference step.

In a standard regression framework, the mentioned base estimators of these models are trained using the MSE loss. We propose to use the adapted CensoredMSE loss from Arany et al. (2022) to allow the models to learn also from additional censored labels where available. As such, we define a one-sided error of a given prediction μ_n and the true label y_n as follows,

$$\varepsilon_n = \begin{cases} \min\left(z_n - \mu_n, 0\right), & \text{if } m_n = -1, \\ y_n - \mu_n, & \text{if } m_n = 0, \\ \max\left(z_n - \mu_n, 0\right), & \text{if } m_n = 1, \end{cases} \tag{3.4}$$

and get the CensoredMSE as $\mathcal{L}^{\text{MSE}} = \frac{1}{N}\sum_{n=1}^{N} \varepsilon_n^2$. The interpretation of the proposed loss is that when the model correctly predicts a value above the threshold of a right-censored label or below the threshold of a left-censored label, the error is set to zero. This reflects a perfect prediction and does not result in any weight updates due to the zero gradient of a constant.

Bayes by Backprop. Blundell et al. (2015) proposed a Bayesian version of NNs, known as BBB, that approximates the posterior distribution of the network weights. In traditional NNs, the weights are deterministic, meaning that a single set of values is learned during training. However, in BBB, the weights are modeled as random variables with associated probability distributions, typically Gaussian. This allows the network to maintain a distribution over the weights, reflecting the uncertainty about the true values of these parameters given the observed data. The theoretical background of the BBB approach is provided in Chapter 2.5.2.

In our implementation, we convert the same NN used in the ensemble-based approaches into a Bayesian NN and use the CensoredMSE as the log-likelihood to allow the BBB model to learn from censored labels. The KL divergence used in the ELBO loss is unaffected by censored labels as it relates solely to the learned weight distribution with respect to the prior. During inference, we sample 500 sets of weights from the learned distribution, and the resulting ensemble of models generates a distribution of predictions. As for the ensemble-based approaches, the spread of these predictions provides an estimate of the epistemic uncertainty, which reflects the model's uncertainty due to limited data or model capacity.

Gaussian models. So far, the base estimators of all described models produce only a prediction of the regression label. Next, we introduce the mean-variance-estimator originally proposed by Nix and Weigend (1994). As shown in Chapter 2.7, using MVE for NNs results in two model outputs: one representing the predicted value and the other representing the model's estimate of the aleatoric uncertainty. The NLL used for model training is shown in Equation 2.60.

We refer to this model as Gaussian going forward, and additionally generate a Gaussian Ensemble from 5 independently trained Gaussian models. A similar approach was used by Busk et al. (2021), who suggests we get the aleatoric estimate from the expected value of all predicted variances and the epistemic estimate from the variance of the prediction. Furthermore, this decomposition of the sources of uncertainty is similar to how it is done for Gaussian Mixture Models (Hubschneider et al., 2019; Yang and Li, 2023). Hubschneider et al. (2019) argues that this decomposition results in epistemic uncertainty estimates that vanish with more data, thus being reducible, in contrast to the aleatoric part which does not, thus being irreducible. As such, these results also correspond to the definition of epistemic and aleatoric sources of uncertainty (Apostolakis, 1990; Kendall and Gal, 2017; Hüllermeier and Waegeman, 2021).

To adapt this learning objective to be able to handle additional censored labels where available, we take inspiration from the Tobit model (Tobin, 1958) used more recently in Bayesian active learning for survival analysis by Hüttel et al. (2024). The intuition behind this method is that for censored labels, we use the integral of the probability density function that is within the correct censored region. Thus, for a left-censored label with threshold z_n the objective is to maximize the integral of $\varphi(y_n|\boldsymbol{x}_n, \boldsymbol{\theta})$ below z_n, in other words to minimize the negative logarithm of the cumulative distribution function $\Phi(z_n|\boldsymbol{x}_n, \boldsymbol{\theta})$. On the contrary, for right-censored labels with threshold z_n the objective is to minimize the negative logarithm of the complement to the cumulative distribution function $1 - \Phi(z_n|\boldsymbol{x}_n, \boldsymbol{\theta})$. In summary, our proposed CensoredNLL is defined as,

$$\begin{aligned}\mathcal{L}^{\text{NLL}} = -\frac{1}{N}\sum_{n=1}^{N}(1 - |m_n|)\ \log\varphi(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}) \\ + |m_n|\ \log \begin{cases} \Phi(z_n|\boldsymbol{x}_n, \boldsymbol{\theta}), & \text{if } \ m_n = -1, \\ 1 - \Phi(z_n|\boldsymbol{x}_n, \boldsymbol{\theta}), & \text{if } \ m_n = 1. \end{cases}\end{aligned} \tag{3.5}$$

Evidential deep learning. Another probabilistic approach to uncertainty estimation is through evidential learning using a NN as proposed by (Amini et al., 2020). Chapter 2.8 introduces the theoretical background for this method. Evidential deep learning models output parameters of a distribution that characterizes both aleatoric and epistemic uncertainty. For model training, the evidential loss (Equation 2.61) is used.

While evidential learning has been used previously to estimate uncertainty in drug discovery-related tasks (Wang et al., 2023b; Yin et al., 2023) its effectiveness at disentangling aleatoric and epistemic uncertainty has recently been questioned (Juergens et al., 2024), therefore we decided not to provide an extension of this model for cases with censored labels. Instead, the Evidential model is used solely as a baseline to benchmark the performances of the other approaches with that of this commonly used model.

Evaluation

The focus of this work is to determine the performance of models on realistic pharmaceutical data. As such, all models are primarily evaluated on the full test sets containing observed and, if available, censored labels. The MSE loss is used to evaluate the performance of the predictions made by the models, in cases of censored labels we use the one-sided squared error from Equation 3.4. Other metrics are required to evaluate the accuracy and calibration of the predicted uncertainties. We consider two ways to evaluate predicted uncertainty: one that evaluates only the accuracy or calibration of the uncertainty, and another that evaluates predictive performance intertwined with how well-calibrated the predicted uncertainty is.

A detailed way to evaluate the predicted uncertainties by themselves is by comparing the confidence-based calibration curve to the identity function, which corresponds to perfect calibration (Hubschneider et al., 2019; Wang et al., 2021; Heid et al., 2023b; Yang and Li, 2023). The theoretical concept of calibration curves is explained in Section 2.2.3.

Furthermore, the Gaussian NLL (Zadrozny and Elkan, 2001) and the ENCE (Levi et al., 2022) are two global metrics that evaluate the intertwined predictive performance and calibration of uncertainties. Section 2.2.3 provides the theoretical background for both metrics. We use the extended version of the Gaussian NLL from the Tobit model that can handle censored labels as defined in Equation 3.5, and does not omit any constant terms. To calculate the ENCE metric, which is derived from the RMSE and the RMV, we again employ the one-sided squared error for the censored labels before taking the root of the mean.

3.4.2 Results and Discussion

We start our experiments by evaluating the effect of training models with the additional partial information available in the real pharmaceutical assay-based setup in the form of censored regression labels. This comparison is conducted as an ablation study, comparing the difference in performance between models trained with and without censored data for each assay. Next, we provide an in-depth comparison between the models on each temporal setting of every assay. The model comparison evaluates the predictive performance and calibration separately for aleatoric and epistemic uncertainty estimates. Finally, we provide practical illustrations of the resulting uncertainty estimates for the best-performing model in terms of the aleatoric and epistemic parts, respectively.

Ablation study

We compare the performance of each model trained with and without the censored data to determine the impact of training uncertainty-aware ML models with additional partial information in the form of censored regression labels for molecular property prediction. Each test set is kept the same, containing all available data, including the censored labels. However, the baseline models are trained only on the observed labels from the training datasets, i.e. $\mathcal{D}_{\text{observed}}$. Similarly, the validation set, used for HP optimization and early stopping, only contains observed labels for the baseline models. We first evaluate the differences in NLL as a measurement of the intertwined predictive accuracy and calibration of uncertainty estimates. We define the test score from the model trained without censored data as $\text{NLL}_{\text{observed}}$ and the test score from the model trained with censored data as $\text{NLL}_{\text{censored}}$. As NLL is minimized for more accurate predictions and better-calibrated uncertainty estimates, we define the evaluation metric as $\Delta\text{NLL} = \text{NLL}_{\text{observed}} - \text{NLL}_{\text{censored}}$ such that a positive difference indicates that the performance was enhanced for the model trained with censored data. Next, we do the same analysis in terms of the MSE to evaluate the effect of including the censored labels on the predictive accuracy independent of the uncertainty estimation.

We evaluate the difference in MSE and NLL for all assays across each temporal setting and test the statistical significance using a two-sided Mann-Whitney-Wilcoxon test (Mann and Whitney, 1947). Due to the vast number of statistical tests required, we apply Bonferroni correction to counteract the multiple comparisons problem (Dunn, 1961). As such, we use a significance level $0.05/m$ where m is the number of datasets times the number of models in the comparison. If the difference is positive and significant ($p < 0.05/m$) the

model trained with censored data is deemed best. Similarly, if the difference is negative and significant ($p < 0.05/m$) the model trained without censored data is deemed best. The results of the ablation study are summarized in Figure 3.12. In addition, the results on test sets containing only observed labels are provided in Figure A.5 of the Appendix. The magnitude of all underlying scores for Figure 3.12 and Figure A.5 are also supplied in Figure A.6 and A.7 in the Appendix.

Throughout the ablation study, the assays are ordered according to the percentage of censored labels available in the full dataset. Note that the two assays without any available censored labels, TB-3 and TB-6, have been excluded. Each star in Figure 3.12 and Figure A.5 indicates that the model trained with or without censored data was significantly better in one of the three temporal settings. As such, each bar can have a maximum of three stars. If a star is located above the bar, the model trained with censored data is the best, and vice versa if it is located below the bar. The dark gray stars mark the significance under Bonferroni correction, but the uncorrected significances are also shown as the light gray stars. The total number of significant cases under Bonferroni correction is summarized in the legend for each model. Note that the total number of datasets in this ablation study is 39, including the 13 assays and 3 temporal settings (Figure 3.1). The error bars illustrate the SD between the temporal settings.

The ΔNLL results in the bottom panel of Figure 3.12 show predominately enhanced performances when the models are trained with the censored regression labels. Particularly, the Ensemble and Gaussian Ensemble models show significant enhancements in a majority of the datasets when trained with censored labels. All other models are enhanced or comparable in all but a few datasets. By instead considering each assay at a time, we see that for all assays, apart from A-PM and A-SOL, more models are improved by the censored labels than hurt by them. The two exceptions are among the assays with the smallest percentages of censored labels available. In this case, the Bonferroni correction could be considered too restrictive given the much smaller number of comparisons in each analysis. In terms of the uncorrected significance, 8 out of the 13 assays see a strong majority of enhanced performances with the censored labels.

Furthermore, the magnitudes of the ΔNLL scores in Figure 3.12 generally become larger for assays with higher proportions of censored labels, with a more substantial shift occurring around 35%. A more detailed analysis of this trend for each model, respectively is seen in Figure A.7 of the Appendix. There, we see that for Ensemble and MC-Dropout the improvements grow for assays with $> 35\%$ censored labels. For these assays, the NLL is on average enhanced by 52% for the Ensemble model and 48% for the MC-Dropout. Similarly, the

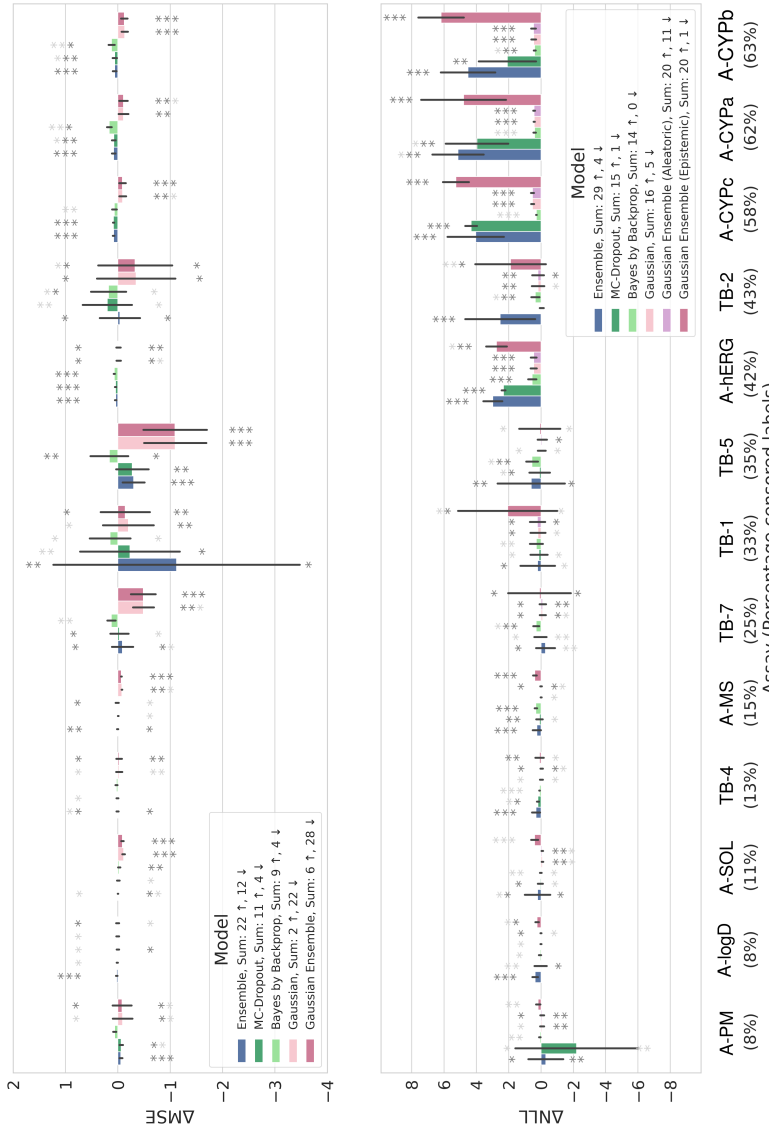

Figure 3.12: **Ablation study.** Difference between MSE (top) and NLL (bottom) for models trained with and without censored labels. The error bars aggregate the three temporal settings. For each temporal setting, a star indicates whether one of the models trained with or without censored labels was significantly better. As such, each bar can have a maximum of three stars. Stars above a bar indicate that the censored model was significantly better, with Bonferroni correction ($p < 0.05/m$) in dark gray and without correction ($p < 0.05$) in light gray. In contrast, the stars below a bar indicate the opposite. The legend summarizes the number of Bonferroni corrected significant differences per model.

Gaussian model and both uncertainty sources of the Gaussian Ensemble model also see substantial enhancements for assays with $> 35\%$ censored labels. For the Gaussian model and the aleatoric part of the Gaussian Ensemble, the NLL is enhanced by 34% on average for these assays. The NLL for the epistemic part of the Gaussian Ensemble for these assays on the other hand is enhanced by 56% on average. Finally, the BBB model is consistently improved already for assays with $> 13\%$ censored labels, although by a slightly lower average enhancement of 24%.

While the NLL provides an overview of the overall performance of these uncertainty estimation methods, it is also interesting to evaluate the effect of the censored labels on the predictive accuracy on its own. The ablation study in terms of ΔMSE is presented in the top panel of Figure 3.12. Note that the Gaussian Ensemble only occurs once in this result, as the different uncertainty estimates do not impact the MSE. There appears to be a greater difference between the effects of the censored labels on the two loss functions in terms of predictive accuracy compared to the NLL. Recall that the Ensemble, MC-Dropout, and BBB models are adapted using the one-sided squared loss, whereas the two Gaussian models use the Tobit loss with similarities to the NLL. Regarding the significant differences, the results for Ensemble, MC-Dropout, and BBB are similar in terms of ΔMSE as they were in terms of ΔNLL. The Ensemble model is enhanced for a majority of datasets, and the other two are enhanced or comparable in all but a few datasets. The improvements are however smaller in magnitude, as seen especially in Figure A.6 in the Appendix. Proportionally, the enhancements of the MSE are still, on average, around 20% for these models on the assays with $> 35\%$ censored labels. The combined results of these models in terms of MSE and NLL, indicate that the inclusion of censored labels during the training procedure mostly impacts the uncertainty estimation abilities of the models rather than predictive accuracy.

On the contrary, the predictive accuracy of the Gaussian models trained with the Tobit loss is mostly not improved by including the censored labels. The magnitude of the loss in predictive accuracy for the Gaussian models is largest for Target assays with $\geq 25\%$ censored labels included for the adapted version of the models, as seen in Figure A.6 in the Appendix. The proportional losses in MSE are comparable to the enhancements in NLL, consisting of around -30% on average for all assays. This is an unfortunate drawback of our proposed adaptation of the Gaussian models using the Tobit loss and should be considered when using the models in practice. However, as these ML approaches are most often used in applications for high-throughput screening in the early stages of the drug discovery pipeline, the ranking of molecular compounds is usually considered more critical than the exact prediction of a given property. As such, we believe the enhancement of the uncertainty estimates can outweigh the loss

in predictive accuracy. Furthermore, it is possible that a weighting scheme included in the Tobit loss functions could make the Gaussian models trained with censored labels also favorable in terms of predictive accuracy.

As a final step in this ablation study, we analyze the impact of training the models with and without the censored labels solely on the observed test labels. Similar to the analysis of the ΔMSE, this is useful to determine which impact the adapted training procedure has on censored versus observed test labels. While this analysis is interesting, we believe it is secondary to the analysis performed on the full set of test labels. The reason is that the full test set is the only one that simulates the real scenario in the drug discovery process. In reality, there would be no way of excluding the compounds that, in hindsight, would result in censored labels when tested in the lab. Figure A.5 in the Appendix shows the ΔMSE and ΔNLL on only observed test labels (i.e. excluding the censored labels). These scores are mostly not improved by including censored labels during training. The performance of the BBB model is mostly not significantly affected by the inclusion of censored labels during its training. The predictive accuracy of the Ensemble and MC-Dropout models is typically worse when censored labels are included, while the impact on the NLL is more varying. The performance of the Gaussian models is almost always negatively impacted by the inclusion of censored labels during training.

By considering the full datasets, including the censored labels, as the true underlying data distributions, we can conclude that the results on the observed labels only illustrate a crucial point. Namely, that the censored labels make up a critical part of the distributions, such that ML models trained with or without them do not transfer to the test set corresponding to the opposite case. Models trained without censored labels do not perform well on test sets with censored labels, and models trained on full datasets do not perform well on test sets without censored labels. We believe this is a natural consequence of the vast imbalance in the censored labels. As seen in Table 3.5, most assays only include censoring in one direction. Only two of the assays explored in this study have a more balanced occurrence of left- and right-censored labels, namely A-SOL and A-MS. However, the total proportion of censored labels for these assays is not large enough to significantly impact the difference in performance between the models trained with and without the censored labels as discussed above. While the reduced performance of the models trained with censored labels on observed test labels is unfortunate, we argue that it is so far an unavoidable side-effect of the optimal performance achieved on true test sets overall. In the future, other ways of incorporating censored training labels could be explored, such as using the absolute values of censored labels without the additional censored information.

All in all, this ablation study has shown that the inclusion of censored labels during training of uncertainty-aware methods can in some cases negatively impact the predictive accuracy as well as the performance on observed labels separately. However, it has also shown that including censored labels during ML is important to accurately estimate well-calibrated uncertainties in real pharmaceutical scenarios, where censoring occurs and cannot be avoided. The enhancements are especially prominent when the proportion of censored labels makes up roughly one-third or more of all available data for a given assay. Given the overall majority of improvements in the NLL seen with censored labels included during training on the full test sets, we recommend the community utilize this data in the future during QSAR modeling for regression tasks. Furthermore, we continue to compare these models to each other and the additional baselines in the following section.

Model comparison

We compare the proposed models trained with all available data, including censored labels, on each dataset by obtaining their predictive accuracy in terms of MSE, their local calibration in terms of the confidence-based calibration curves, and their overall performance in terms of NLL and ENCE . Where uncertainty estimates are evaluated, we separate the aleatoric and epistemic parts. For each dataset, the three temporal settings are evaluated separately, such that the results can be compared between different sizes of training sets, and as time evolves. For instance, it is relevant to know if conclusions drawn from the first temporal setting also hold throughout time, i.e. on the following temporal settings. In this analysis, the assays are first ordered by category and then according to the overall size of the dataset. For the scores, MSE, NLL, and ENCE, the model with the best average score over 10 repeated experiments is marked with a star. Additionally, any other performances not significantly different from the best model according to a one-sided Mann-Whitney-Wilcoxon test are also marked with stars (Mann and Whitney, 1947). Similar to the ablation study, we use Bonferroni correction to counteract the multiple comparisons problem (Dunn, 1961). Here, the significance level is $p < 0.05/m$, where m is the number of comparisons per dataset. The two TB assays without censored labels are included. Apart from the proposed extended models, which are trained with censored labels, two additional baselines are included: a RF model and a model trained with evidential deep learning. These models are trained without censored labels.



Figure 3.13: **Predictive accuracy.** Comparing the predictive accuracy of all models across all temporal settings in terms of MSE, aggregated over 10 experiments. For each dataset, the best model in terms of mean MSE is marked with a star together with any other models not statistically worse based on a one-sided Mann-Whitney-Wilcoxon test with Bonferroni correction. Apart from the RF and Evidential models, all other models are trained with censored labels where available.

Predictive accuracy. The predictive performances of the models are presented in Figure 3.13 in terms of MSE for each temporal setting. The different temporal settings are shown in Figure 3.1. An initial observation is that the MSE is generally lower for the models trained on the ADME-T assays. This result is expected given the diversity and size of the ADME-T assays, compared to the TB assays. More data means that the models can learn to generalize better and the lack of shifts in both the feature space and the label space provides an easier prediction task. For seven out of the eight ADME-T assays, the Ensemble model is the best or among the best for all temporal settings. In a couple of instances, the MC-Dropout model is comparable, and for the A-PM assay the BBB model significantly outperforms all other models. In the TB assays, there is a greater variation in the best-performing models. For three TB assays, the BBB model is consistently among the best across all three temporal settings, and in one TB assay, the Ensemble model is consistently among the best. However, for the remaining three assays TB-1, TB-2, and TB-7 the best-performing model changes drastically over time. This observation is of great importance, as it illustrates that, typically, the conclusions about the best-performing model drawn early in the development of an ADME-T assay are reliable and still hold

throughout the later development of the assay. On the other hand, as TB assays might significantly shift in chemical space over time the same is not always true for the model comparison on these kinds of assays. For TB assays, a new model comparison might be needed later on in the development of the assays to keep the best-performing model up-to-date.

For the most part, the RF and Evidential models, which are trained without censored labels, perform poorly compared to the best-performing models. This is also true for the Evidential model on the two assays that do not have any censored labels, meaning that neither of the other models has the advantage of being trained on more data. We interpret this as an indication that the poor performance of the Evidential model compared to the other models is not primarily related to the fact that the other models are trained with censored labels. The RF model, however, does reach comparable results to the BBB model for some of the temporal settings on the two assays, but it is never significantly better than all other models on any datasets. As such, it might be interesting to try to extend the RF model to also handle censored labels in the future. Note that this should be possible, similar to how the one-sided square loss was used for models originally trained using MSE. However, it would require lower-level changes to the implementations in Scikit-learn (Pedregosa et al., 2011) which we deem out of the scope of this study given that it does not significantly outperform any of the other proposed methods herein. We do not believe it would be possible to simply include the censored labels as they are during the training of the RF, given the non-fixed window of censoring per assay.

Considering all datasets, i.e., assays and their respective temporal settings, the Ensemble model performs best overall, by being among the best-performing models for 28 out of the 45 dataset instances. The only approach that matches the performance of the Ensemble model is the BBB model which is among the best for 19 dataset instances. From the remaining models, the MC-Dropout model is only among the best-performing models for 9 dataset instances, usually together with the Ensemble model. The Evidential model and the RF model are among the best, together with the BBB model, in 7 versus 5 cases each. In the second temporal setting, when the models for the TB-1 assay are trained with two folds, the two Gaussian models perform best together with the Ensemble model. However, this is the only instance where these models are among the best.

It is reasonable and compatible with previous conclusions, that the Ensemble model achieves the highest predictive accuracy, i.e. lowest MSE (Ovadia et al., 2019). The reason is that this model contains the highest number of individually trained models. As such, the consensus in the predictions by the base models can reach a higher accuracy in cases where the problem is solvable by the available

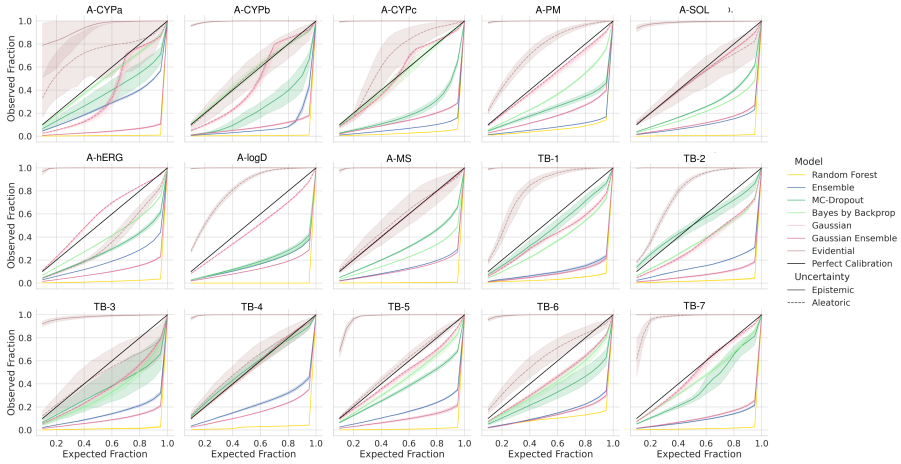

Figure 3.14: **Calibration curves.** Full calibration curves for all uncertainty estimates on the third temporal setting containing three folds in the training set, aggregated over 10 experiments. The black line in each panel illustrates what a perfectly calibrated model would look like. Apart from the RF and Evidential models, all other models are trained with censored labels.

data. In comparison to the Gaussian Ensemble, the regular Ensemble has a slight advantage in terms of solely predictive accuracy because it is only trained to optimize the MSE. The Gaussian model simultaneously needs to optimize the calibration of the predicted uncertainty and thus might need to sacrifice some predictive accuracy if they are misaligned. For the RF, the number of decision trees was optimized for each dataset in the model selection, as such, in some cases the ensemble contains more than 50 trees. However, as the RF is less complex than a NN it is still reasonable that the NN ensemble performs better. Out of the remaining models, the Evidential model relies on a single trained base model, but the Bayesian models instead learn an infinite set of model parameters.

Overall, our results show the Ensemble as the best-performing alternative considering only the predictive accuracy of the considered models. However, for practical applications, the computational cost of this model also needs to be taken into consideration. It is substantially more demanding to train 50 individual models for the Ensemble compared to the single model architecture used in the Bayesian approaches. As such, it is worth keeping in mind, that when BBB performs comparable to the Ensemble, it will also be significantly faster to train.

Calibration. Next, we evaluate the local calibration of the uncertainty estimates of each model in terms of the confidence-based calibration curves described earlier. The results are shown in Figure 3.14. We focus this analysis on the third and biggest temporal setting, in which 3 temporal folds are used for model training (Figure 3.1). The results from the two earlier settings are provided in Figure A.10 of the Appendix. For the most part, we see that all aleatoric estimates are much better calibrated than many of the epistemic estimates. This is an interesting observation as it holds for models that produce both epistemic and aleatoric uncertainty estimates, including the Gaussian Ensemble and the Evidential model, as well as the simple Gaussian model, which only produces aleatoric estimates. The observation is true for ADME-T and TB assays alike and thus irrespective of distribution shifts or dataset size.

Among the epistemic uncertainty estimates, a clear trend shows that the Evidential model is grossly under-confident, whereas many of the ensemble-based models are over-confident, especially the RF model. This means that the Evidential model predicts CIs that are far too wide and, on the contrary, that the ensemble-based models predict very narrow CIs. For the three ADME-T CYP assays, the BBB estimates of the epistemic uncertainty are well-calibrated. Similarly, for many of the TB assays, both the BBB model and the MC-Dropout model produce well-calibrated estimates of the epistemic uncertainty. These observations corroborate previously reported findings that ensembles tend to be under-calibrated in terms of the epistemic uncertainty compared to Bayesian approaches (Hubschneider et al., 2019).

As pointed out by Rasmussen et al. (2023), these confidence-based calibration curves can, in some cases, be misleading in determining whether predicted uncertainties accurately depict small versus large errors. Perfect calibration could, in principle, be achieved by a model that always predicts the same level of uncertainty. As such, Levi et al. (2022) propose to additionally measure the dispersion of the predicted uncertainties using the coefficient of variation, c_v, as follows

$$c_v = \frac{\sqrt{\frac{\sum_{n=1}^{N}(\sigma_n - \mu_\sigma)^2}{N-1}}}{\mu_\sigma}, \quad \text{where } \mu_\sigma = \frac{1}{N}\sum_{n=1}^{N}\sigma_n. \tag{3.6}$$

We provide these values for all models and datasets in Figure A.11 of the Appendix. According to these results, we deem that all of the best-performing models exhibit sufficient dispersion in their uncertainty estimates. There are some cases where the dispersion is very low for the RF model, but this is only the case on some of the very smallest training sets where it is understandable that the model struggled to learn regardless. Additionally, many previous studies on uncertainty estimation have shown that the calibration can typically be improved by recalibrating the uncertainties (Janet et al., 2019; Wang et al., 2021;

Rasmussen et al., 2023). It is, however, not straightforward how recalibration could be applied in the context of censored labels. Thus, we leave this to be explored in future work.

Overall performance. In the final step of our model comparison, we compare the models using the global scores NLL and ENCE which take both the predictive accuracy and the calibration of uncertainties into account. The results for NLL of the epistemic estimates are shown in Figure 3.15 while the results for NLL of the aleatoric estimates are provided in Figure A.8 of the Appendix. Additionally, the ENCE scores are provided in Figure A.9 of the Appendix, where similar conclusions can be observed as for the NLL scores. The NLL scores are highly varying for some of the poorly performing models, and therefore the plots have been cut to show the NLL below 2.5 in the case of the aleatoric models and 6 for the epistemic models. This was done for readability and does not hide any information about the best-performing models.

In terms of epistemic uncertainty, the BBB model is overwhelmingly best in terms of NLL, with 35 out of 45 best-performing scores. Only on the TB-4 assays, does the Ensemble model outperform BBB consistently throughout time. On 7 out of all 15 assays, the BBB model outperforms all other models entirely in all temporal settings. For an additional 4 assays, BBB is consistently among the best-performing models throughout time. This observation means that in terms of the NLL, the conclusions drawn early for an assay from both categories can oftentimes be trusted to hold throughout time. We believe that this robustness for the TB assays, which exhibit distribution shifts, speaks to the reliability of the epistemic uncertainty estimates in accounting for distribution shifts where present. As such, the best-performing model in terms of predictive accuracy might change over time, only due to differences in how well the models handle the distribution shifts. Still, the overall best-performing model is reliable for all assay types irrespective of distribution shifts.

Similarly to what was observed in the MSE, Figure 3.15 shows that the NLL scores are generally slightly lower for the ADME-T assays compared to the TB assays. TB-4 is again an outlier with quite low NLL scores for the best-performing models compared to the other TB assays. Recall from Equation 3.5 that the NLL includes a term with the squared error. Thus, it is likely that this trend arises from that source. In the ENCE scores presented in Figure A.9 of the Appendix, the same trend can be partially observed but it is not as prominent. Apart from this, the BBB model also outperforms all other models in all temporal settings for 8 assays in terms of ENCE. Contrary to the NLL, the Ensemble is never among the best-performing models in terms of ENCE, but the Evidential model is consistently among the best for 3 assays. Regardless

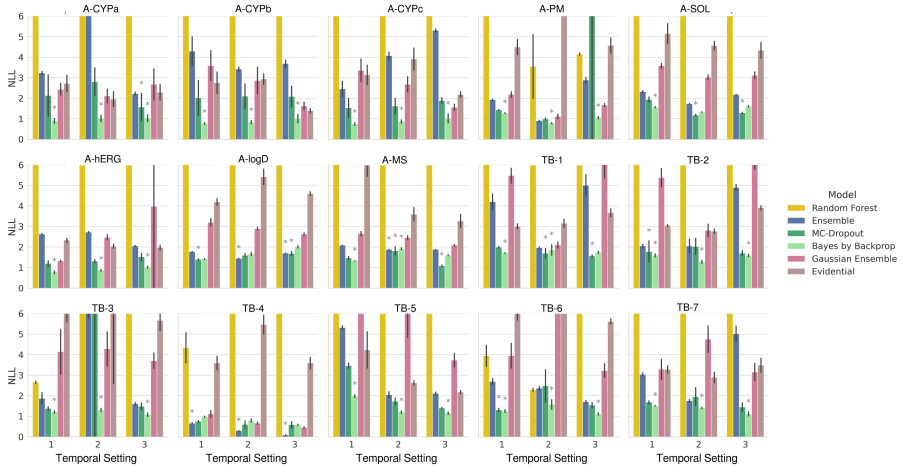

Figure 3.15: **Combined accuracy of uncertainty estimation and predictive performance.** Comparing the NLL of all epistemic uncertainty estimating models, aggregated over 10 experiments. For each dataset, the best model in terms of average NLL is marked with a star together with any other models not statistically worse based on a one-sided Mann-Whitney-Wilcoxon test with Bonferroni correction. Apart from the RF and Evidential models, all other models are trained with censored labels where available.

of which model is the best, the one that is best in the first temporal setting is, in most cases, also best in the subsequent temporal settings, also in terms of ENCE.

For the aleatoric uncertainty estimates, shown in Figure A.8 of the Appendix, the Evidential model is significantly outperformed by either or both of the Gaussian models in all but 3 of the 24 ADME-T dataset instances. Among the TB assays, there are more times that the Evidential model is comparable to or outperforms the Gaussian models. However, the TB-5 assay is the only one where the Evidential model is consistently among the best through time. On the other hand, the Gaussian Ensemble is the single best or among the best for the vast majority of datasets, namely in 39 out of 45. In the case of the NLL for the aleatoric uncertainty estimates, like for the MSE, the best-performing model is almost exclusively robust throughout time for the ADME-T assays, but not always robust for the TB assays. The same conclusions can be seen in terms of the ENCE score. Note also that the NLL and ENCE scores for the aleatoric uncertainty estimates are mostly lower than for the epistemic uncertainty estimates, similar to the results seen in the calibration curves.

Case study

Finally, we perform a case study on one example assay from each category to explore the practical implications and usefulness of the epistemic and aleatoric estimates, respectively. In the case study of aleatoric uncertainty, we compare the predicted uncertainty estimates with ground truth experimental error in the form of the SD between duplicated experiments. For this analysis, we cannot include censored labels due to the aggregation strategy described in earlier sections. Additionally, we exclude experiments repeated less than three times and those with no variation reported in the data. We chose the assays with the most available compounds with duplicated experiments, e.g. A-hERG and TB-6. For the epistemic case study, we instead pick TB-7 as a TB assay due to its particularly challenging distribution shift. Similarly to the aleatoric case study, we compare the results with the A-hERG assay, which has no visible distribution shift in either the label or feature space as seen in Figure A.3 and Figure A.4 of the Appendix, respectively.

This case study is solely possible due to the temporal split and the additional information available through the internal pharmaceutical data, which are generally not accessible or trustworthy in public datasets. Similar remarks were made by Sheridan (2013) and point to the necessity of openness from the pharmaceutical industry to learn from each other and push the field forward.

Epistemic uncertainty quantification. For the practical case study of the epistemic uncertainty estimates, we illustrate the epistemic uncertainty estimates averaged over the 10 repeated experiments of the BBB model on the A-hERG and the TB-7 assays. The BBB model is chosen as it is significantly the best model in terms of NLL across all temporal settings of both of these example assays, as shown in Figure 3.15. The case study compares the epistemic uncertainty between different regions of the t-SNE projections of each test set, as illustrated in the rightmost part of each panel in Figure 3.16. It also relates this to the full t-SNE projection, including the folds used for training and validation, as seen in the leftmost parts of the same figure. Note that this only accounts for distributional shifts in a linear approximation of the feature space. As the fundamental nature of the feature space is non-linear, this analysis cannot cover these non-linearities. Similarly, we cannot evaluate any parts of the epistemic uncertainty related to other factors than the distribution shift in this analysis. The other factors might include epistemic uncertainty that relates to the choice of model architecture or training procedures. We have to rely on the metrics used in the model comparison section above to account for these factors of epistemic uncertainty.

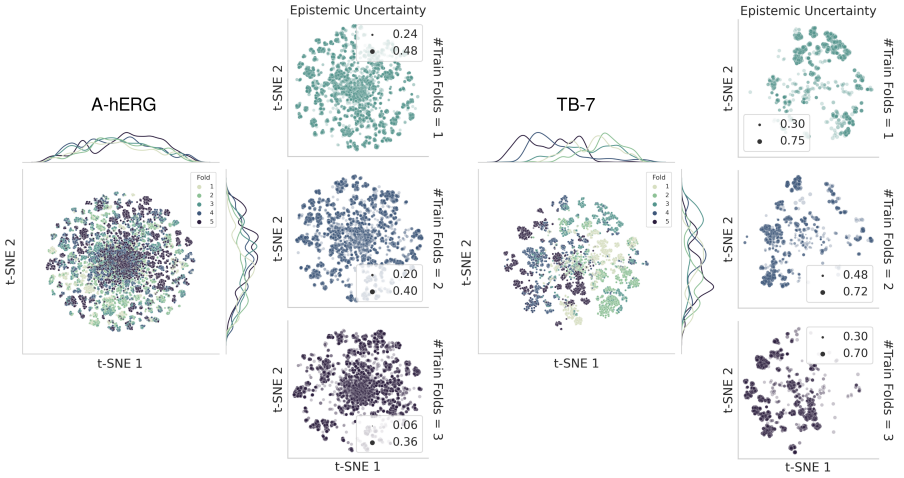

Figure 3.16: **Epistemic estimates by Bayes by Backprop.** A practical illustration of the epistemic uncertainty estimates by the BBB model on two assays: A-hERG without clear distribution shifts over time, and TB-7 with prominent distribution shifts over time. T-SNE plots for all temporal settings corresponding to 1,2, and 3 training folds are shown.

In the full t-SNE projections of each dataset containing all five folds, we see the trend described in the Methods section of this study (Section 3.4.1). The chemical space of the A-hERG assay is highly diverse, already in the first fold, and does not shift substantially over time. However, some clusters are formed in the outer edges of the chemical space, where the different folds are well separated. For the TB-7 assay, on the other hand, there is a clear shift in the chemical space from the right side of the plot representing the initial measurements to the left side showing the later folds. Similar trends are also present in the label distributions as seen in Figure A.3 of the Appendix. Furthermore, the label distribution for the TB-7 assays does not shift continuously over time. Instead, it shifts greatly toward higher pIC50 values in the second fold and then back toward lower values in the last two folds. The remaining three plots to the right in each panel of Figure 3.16 show the t-SNE projections of each test set, i.e., folds 3, 4, and 5, separately. Here, the size of the data points is determined by the averaged epistemic uncertainty predicted by the BBB model trained on the three temporal settings, respectively, i.e., with an increasing number of training folds. Note that the legends of these plots detail the respective minimum and maximum predicted uncertainties on the given test set.

For the A-hERG assay, only minor differences can be observed between compounds positioned in various regions of the feature space. This is expected given the lack of overall distribution shifts between the folds. Additionally, we evaluate the span of uncertainty estimates, according to the minimum and maximum values listed in the legend of each plot. The third temporal setting has the largest span of 0.3 compared to 0.24 and 0.2 for the first two settings. This indicates that the model is better at separating the compounds in the final temporal setting compared to the previous one, likely due to the increased size of the training data. Another effect of the increasing training data can be seen in the difference in the maximum predicted uncertainty between the folds. For the model trained on three folds, the maximum is 0.36, which is smaller than the values seen for the models trained on less data, which are 0.4 for the model trained on two folds and 0.48 for the model trained on one fold. This is a direct and expected behavior of the epistemic uncertainty, that more data results in more certain models i.e., lower predicted uncertainties. In the case of the TB assay, we note that the maximum estimated uncertainty on each test set is very high, between 0.7-0.75. This can be a result of the general distribution shift present between every fold or the overall small dataset. In both cases, overall high uncertainty is to be expected.

Based on the distribution shift present in the feature space, and considering that epistemic uncertainty should account for distributional shifts, we expect that distribution shifts are reflected in the predicted epistemic uncertainties. As such, our analysis provides empirical evidence to support this claim, and it illustrates that the uncertainty estimates cover additional sources of uncertainty related to the model itself, such as poor generalization in cases when training data is limited. It is important to understand all sources of uncertainty when basing future critical decisions on model predictions, such as in drug discovery.

Considering the cases addressed in this section of our study, we can deduce practical suggestions on how the identified sources can impact the continued drug discovery process. If poor generalization is determined, indicated by overall high uncertainty estimates and low predictive performance of a model, the model needs to be retrained with more data before deployment. Another alternative is to reconsider the choice of model, but our temporal study shows that the BBB model remains the best choice in the future, despite the addition of more data. When distribution shifts are identified, such as in the TB assay, further data exploration in chemical spaces with high uncertainty estimates is needed before deployment.

Further research is necessary to disentangle the sources of epistemic uncertainty, including distribution shifts and other model-related sources. One alternative approach would be to quantify the distribution shift using other means, either with distance-based approaches, such as the average Tanimoto or cosine

similarity (Sheridan et al., 2004) between an inference compound and compounds in the training set, or the interpretable method proposed by Kulinski and Inouye (2023). Additionally, more advanced pre-training procedures can be used, that are trained to incorporate distribution shifts more effectively (Bertolini et al., 2023). After the distribution shift has been independently quantified, the predicted epistemic uncertainty could be re-evaluated such that the remaining model uncertainty is disentangled from this information.

Aleatoric uncertainty quantification. To examine the aleatoric estimates by the best-performing model from the model comparison above, we need a way of quantifying the true experimental error. As such, we consider the variation between duplicated measurements of a single compound on the assay to provide this information. However, note that this information was not provided to the models during training because duplicated measurements were aggregated as part of the data preprocessing described in Section 3.4.1. This procedure of aggregation is common when applying ML to molecular property prediction, but an important question is whether the available methods for estimating aleatoric uncertainty can learn the experimental error from such preprocessed data.

One potential issue with the described estimates of the true experimental error is that the number of duplicated measurements varies greatly between compounds. Some compounds are tested many times, whereas many are tested only a handful of times or only once. The compounds for which there is only one measurement do not give any indication about the experimental error, and those where only a few duplicates are available might not be as reliable as those where the experiment was repeated many times. For this reason, we consider only compounds from the test sets that have more than 2 measurements and ones where the variation was greater than exactly zero.

After filtering for duplicated experiments in the available data, the A-hERG assay contains 712 compounds with duplicated measurements from the test sets of all three temporal settings combined. The TB-6 assay similarly contains 1712 duplicated experiments in total between the three test sets. In this analysis, we focus on the aleatoric estimates averaged over the 10 repeated experiments of the Gaussian Ensemble. In the previous experiments of this study, this model emerged as the significantly best or top-performing model across all temporal settings of the A-hERG and TB-6 assay. Figure 3.17 illustrates the relationship between the predicted aleatoric estimates and the available experimental errors, colored by the three test sets.

In general, we cannot necessarily expect the model to learn the correct scale of the experimental errors since the experimental error is not included in the training data. Instead, we are looking to evaluate if any upward trends can

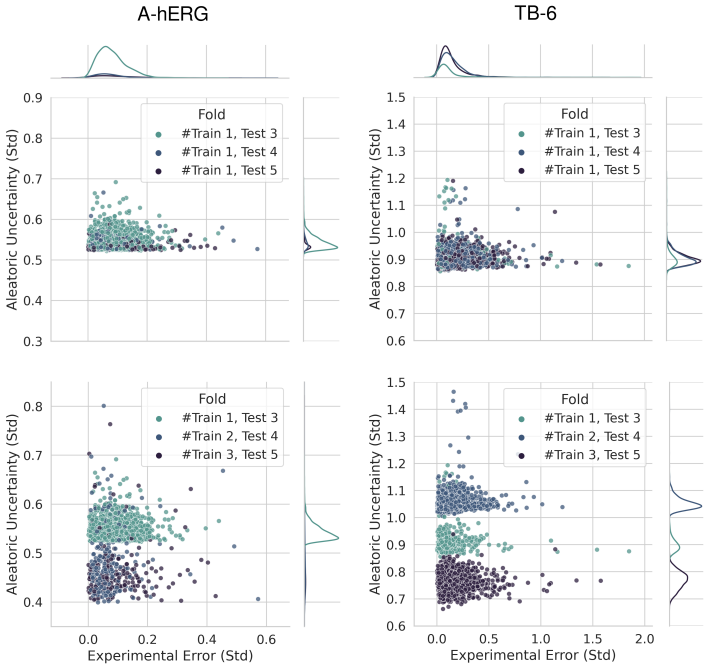

Figure 3.17: **Aleatoric estimates by Gaussian ensemble.** A practical illustration of the aleatoric uncertainty estimates by the Gaussian Ensemble on two example assays. Comparing the predicted aleatoric uncertainty to the experimental error. (Top row) Predictions by the model trained only on the first fold, corresponding to the first temporal setting. (Bottom row) Predictions by the respective models trained in all temporal settings.

be observed, such that compounds with larger experimental error also prompt greater predicted aleatoric uncertainty. However, no such trend can be observed for any of the models or test sets on the two assays. Given that the aleatoric estimates are still well-calibrated according to the observed errors, as seen in the calibration curves of Figure 3.14 and the overall scores shown in Figure A.8 and Figure A.9 of the Appendix, this could indicate that the learned aleatoric uncertainties reflect something other than the experimental error examined here. The model may be able to learn trends in the label space given similar compounds in the feature space. For instance, activity cliffs can exist where very similar compounds prompt significantly different responses in an assay (Maggiora, 2006). Such relationships could be explored in future research.

Additionally, we compare the scale of the predicted aleatoric estimates across different test sets and assay types. First, we see that the estimates are significantly lower on the A-hERG assay than on the TB-6 assay. This might relate to several factors that make the two assays distinctly different. It could simply be a result of the overall size of the datasets: recall from Table 3.1 that the TB-6 assay has only 13,093 labels compared to the 67,687 labels available for A-hERG. However, this trend also corresponds well with the observed SD of the control compounds on the two assays, which was 0.19 for the A-hERG assay and 0.31 for the TB-6 assay. These errors from the control compounds should be more reliable and thus give a better general estimate of the overall homoscedastic noise present in each assay.

Furthermore, an important discrepancy can be observed by comparing the distributions of the aleatoric estimates on each test set between the first row and the second row of Figure 3.17. In the first row, all predictions are made by the same model, the one trained on only the first fold. Here, the distributions of the aleatoric estimates are roughly the same for each test set of the two example assays, respectively. In the second row, the illustrated predictions of each test set are made by different models, trained on an increasing number of folds corresponding to the different temporal settings. Thus, these are the models used throughout all previous results of this study. Apart from these models being trained on different amounts of data, they are also optimized individually. Therefore, these models can have vastly different model architectures and be trained using different training procedures, such as learning rate.

Crucially, none of these factors should have any effect on the aleatoric uncertainty as it is often categorized in literature as irreducible (Apostolakis, 1990; Kendall and Gal, 2017; Hüllermeier and Waegeman, 2021). Despite this, we do see distinct differences between most of the distributions of aleatoric estimates by each of these different models in the bottom row of Figure 3.17. Specifically, we see some instances where increasing amounts of training data result in generally lower aleatoric estimates. However, there is also one case where the opposite is true, between test folds 3 and 4 of the TB-6 assay. We can only assume this relation to be because of the differences in model architectures. Therefore, our empirical results raise questions about whether the explored models' supposed estimates of aleatoric uncertainty are really fully disentangled from the epistemic uncertainty. Similarly, recent theoretical work on deriving suitable measures for aleatoric, epistemic, and total uncertainty has found that the aleatoric and epistemic parts do not necessarily have to add up to the total uncertainty (Sale et al., 2024). As such, the disentanglement of the sources of uncertainty should be considered an ongoing field of research that needs more work to fully determine how these estimates should be categorized and how they relate to the underlying noise in the data.

Chapter 4

UQ in Drug-Target Interaction Models Using Hamiltonian Monte Carlo Sampling

This chapter was adapted from the following publications:

Section 4.2: **Friesacher et al. (2025a)** Friesacher, H.R.[1], Engkvist, O., Mervin, L., Moreau, Y., Arany, A., 2025a. Achieving well-informed decision-making in drug discovery: a comprehensive calibration study using neural network-based structure-activity models. Journal of Cheminformatics 17, 29.

[1]F.H.R. was responsible for conceptualization, data curation, methodology, software, investigation, formal analysis, visualization, writing

4.1 Introduction

One way to quantify uncertainty in ML models is via the Bayesian approach, which obtains probability distributions instead of point estimates for model parameters as described in Chapter 2.5.1. However, the high dimensionality of NN models often makes it computationally infeasible to obtain an exact solution of the posterior distribution. Therefore, sampling approaches can be used to approximate the posterior. HMC sampling is considered the gold standard for posterior approximation (Izmailov et al., 2021), as it utilizes geometrical information from the posterior to retrieve samples efficiently (Betancourt, 2017). For instance, Schweighofer et al. (2023) use HMC to obtain a ground truth for uncertainty quantification with adversarial models. However, HMC sampling is computationally expensive, with long training times that make sampling from larger models challenging. Furthermore, the samplers are sensitive to HP settings, which need to be tuned with care to achieve converging chains that sample efficiently from the whole posterior.

In this work, we aim to quantify the uncertainty associated with a model prediction using HMC sampling. More specifically, we aim to either sample the whole NN to retrieve a ground truth for UQ in drug-target interaction predictions or to improve the UQ of conventional MLP by sampling only parts of the NN. Furthermore, different HP optimization strategies for NN-based drug-target interaction predictions are compared to identify approaches that yield well-calibrated models.

This chapter summarizes the research on UQ using public data and HMC sampling for improved probability calibration in classification QSAR models. The main contributions of this study are as follows:

- We use a computationally efficient Bayesian UQ method named HMC Bayesian last layer (HBLL), which obtains samples for the parameters of a Bayesian logistic regression fitted to the hidden layer of the baseline NN [Section 4.2].
- We evaluate the performance HBLL in comparison to other UQ methods and investigate the impact of different HP optimization metrics on the probability calibration [Section 4.2].
- We train a Bayesian NN using HMC applied to the full NN, which can be used as ground truth for UQ in QSAR models [Section 4.3].

4.1.1 Public Dataset

Table 4.1: **Overview of public assay data.** Details for the assay data extracted from the ChEMBL dataset are reported. Data from three assays of varying sizes and positive ratios were extracted.

ChEMBL - IDs	Target	#Results	Active Ratio	pIC50 Threshold
CHEMBL340	Cytochrome P450 3A4	7619	0.252658	5.5
CHEMBL1951	Monoamine oxidase A	2917	0.259170	5.5
CHEMBL240	hERG	9558	0.079828	6.5

Target-specific bioactivity data were extracted from the ChEMBL database (version: 29) (Mendez et al., 2018) including compound activities from three different targets, namely monoamine oxidase A (MAO-A), CYP3A4, and hERG. The targets were chosen to represent common optimization problems in the drug discovery process. CYP3A4 plays an important role in drug-drug interactions and the metabolism of compounds (Deodhar et al., 2020; Ioannides, 1996). Furthermore, inhibition of hERG can lead to severe cardiac side effects (Keating and Sanguinetti, 1996), and detecting interactions with these proteins is essential for evaluating the pharmacodynamic and toxicity profiles of drug candidates. MAO-A, on the other hand, was chosen as representative for a class of targets that are typically modulated to achieve a desired biochemical effect. MAO-A is modulated by therapeutic agents treating behavioral and neurological disorders. Table 4.1 summarizes the properties of the used target data.

Fold Generation via Clustering. The data was split into five different folds to enable cross-validation. In cross-validation, data is excluded from the training process, which is used to validate and evaluate the model after training. This allows the assessment of the model's generalizing abilities by evaluating its performance on unseen data. In our study, we used three folds for model training, one for validation, and one for testing. Figure 4.1 illustrates the generation and use of the dataset splits. The validation fold used for the model´s quality assessment during hyperparameter tuning and for early stopping was excluded from the training dataset.

To obtain the folds, we used the procedure of fold generation described in detail in Simm et al. (2018). In short, Tanimoto similarity computed from ECFP features was used to measure chemical similarity between compounds, which served as the basis for subsequent cluster assignments. Next, the entire clusters containing similar compounds were randomly assigned to folds. This procedure ensures that training and test datasets consist of compounds from divergent chemical space, mimicking the real-world scenario, in which the model is used

to predict bioactivities for chemically unfamiliar compounds. This prevents testing the model on compounds similar to those used in training, which would yield overoptimistic performance estimates.

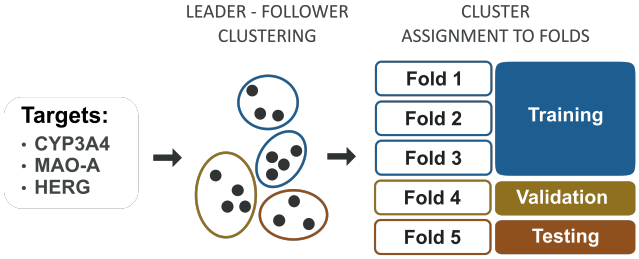

Figure 4.1: **Generation of the public dataset.** The chemical structures were extracted from ChEMBL, and subsequently filtered and clustered. The clusters were assigned to five folds, which were used to set up a training, validation, and test fold.

4.2 A Comprehensive Calibration Study via HP Tuning and HMC Bayesian Last Layer

This paper makes the following contributions:

- We provide a comprehensive probability calibration study using NNs for drug-target interaction predictions
- We propose using the computationally efficient Bayesian uncertainty estimation method HBLL, which generates HMC trajectories to obtain samples for the parameters of a Bayesian logistic regression model fitted to the hidden layer of the baseline NN.
- We compare different metrics, including accuracy and calibration scores, used for model HP tuning to investigate which model selection strategy achieves well-calibrated models.

Several papers examine the quality of uncertainty estimates in models used across various drug discovery applications. A comprehensive overview can be found in Section 2.10. While all of these papers provide significant contributions to identify approaches that obtain high-quality uncertainty estimates, there is, to our knowledge, no study investigating the impact of different HP optimization

metrics on the calibration properties of bioactivity prediction models. In the first part of this study, we aim to close this gap by contributing an analysis of how to train binary classification models when striving for good uncertainty estimates.

In the second part, various UQ approaches are evaluated based on the quality of their uncertainty estimates. We propose a limited computational complexity Bayesian approach, which allows the retrieval of samples from the posterior distribution of the last layer weights. Partial Bayesian NNs apply Bayesian approximation solely to the final layer of a NN, and have already been introduced in the literature (Fiedler and Lucia, 2023; Brunzema et al., 2024; Ober and Rasmussen, 2019; Snoek et al., 2015; Watson et al., 2021). Harrison et al. (2024) recently showed that variational Bayesian last-layer models enhance the calibration and accuracy of baseline models in classification and regression tasks. Note that all of these works use variational approximation approaches, while we apply HMC to retrieve samples from the posterior of the last layer weights. Consistent with the terminology used in the existing literature (Fiedler and Lucia, 2023; Brunzema et al., 2024; Watson et al., 2021; Harrison et al., 2024), we refer to the method as HMC Bayesian last layer (HBLL). We compare the proposed HBLL approach to the uncalibrated baseline model, two common UQ, and one probability calibration method. Finally, we investigate if combining the post hoc calibration approach PS with other UQ methods benefits model calibration.

4.2.1 Methods

Data

Section 4.1.1 provides an overview of the data used in this work. Bioactivities were converted to pIC50, and target-specific thresholds were chosen for assigning bioactivity labels as shown in Table 4.1. For MAO-A and CYP3A4, a pIC50 value of 5.5 was chosen as a threshold, which resulted in active ratios of approximately 25% for both targets. To investigate if our conclusions were also valid for smaller active ratios, we chose a stricter pIC50 threshold of 6.5 for the remaining target hERG, leading to an active ratio of 7% in this dataset. ECFPs (size = 32k, radius = 3), were obtained using RDKit Version 2022.03.2 (Landrum, 2006) which were used as model inputs. More advanced methods are available for molecular representations and model architectures, including graph NNs for handling molecular graphs and language models for processing SMILES representations. However, since the aim of this method is to identify modeling approaches that produce high-quality uncertainty estimates rather

than find the overall best model, we chose the simple ECFP representation. Furthermore, the combination of ECFP with MLPs, as used in this study, has been shown to outperform other deep learning approaches (Van Tilborg et al., 2022; Mayr et al., 2018).

Single-Task Model Generation

Models were implemented and trained using PyTorch Version 2.1.0 (Paszke et al., 2019). The open-source hamiltorch package (Cobb et al., 2023) was used to generate the HBLL models using HMC samplers (Cobb and Jalaian, 2021).

Baseline Model. In this study, we adopt the modeling framework SparseChem, proposed by Arany et al. (2022), which has been used by multiple industry stakeholders (Heyndrickx et al., 2023). MLPs were generated and used as the baseline for binary classification tasks. The mathematical background for training and predicting with MLPs is outlined in Section 2.4. Subsequently, the baseline models were extended to compare different uncertainty estimation and probability calibration approaches. Figure 4.2 illustrates the architecture of the baseline MLPs and the HP optimization workflow. The baseline MLP is comprised of two layers, with a ReLU function and a dropout layer in between. Probability-like scores were obtained by applying a sigmoid function. The size of the hidden layer and the dropout rate were tuned in a grid search as described in the next section. Deeper MLPs with 2 to 4 additional layers were tested to evaluate whether they could enhance the performance. The results of this experiment show no significant improvements in using deeper MLPs, as shown in Section B.1.2 in the Appendix. With regard to the HBLL model, the performance of HMC samplers in the last layer should be comparable independently of the depth of the network. Sommer et al. (2024) demonstrated that the mixing of HMC samplers improved in deeper layers of the network, while it was harder in the first and last. Considering these findings, we opted for the shallow MLP structure, as described above.

Model Tuning. The baseline models were trained using the BCE loss. A validation dataset was used for early stopping during training and to optimize the HPs of the models. An exhaustive grid search in parameter space was performed to tune the size of the hidden layer, and the dropout rate of the model, as well as the learning rate and the weight decay used during model training, as shown in Figure 4.2. The HP space that was considered during HP tuning is reported in Table B.7 in the Appendix. For each HP setting, the HP metric was averaged over ten model repeats, which were initialized randomly, to

ensure repeatability. BCE loss, ACE, accuracy, and AUC were used for model selection to assess the impact on probability calibration.

Train-time Uncertainty Quantification. Approaches that estimate uncertainty during training are referred to as train-time uncertainty quantification methods in this work. More specifically, we define train-time uncertainty estimation approaches as methods that do not need a separate calibration set for correcting the predicted probabilities. The main idea of these techniques is to provide uncertainty of the model parameters by treating the model parameters as random variables with associated probability distributions. Two ensemble-based approaches are included in this study: MLP ensembles, which are referred to as Ensembles throughout this work, and MC-Dropout models. The theoretical background for ensemble-based methods can be found in Chapter 2.6. Furthermore, we include a partial Bayesian NN in our study, which generates HMC trajectories to obtain samples for the parameters of a Bayesian logistic regression fitted to the hidden layer of the baseline neural network. This approach is referred to as HBLL throughout this work. HMC is a MCMC method, which allows drawing samples directly from the posterior distribution of the parameters. Chapter 2.5.3 provides the theoretical background for sampling with HMC. In all train-time uncertainty quantification methods, we obtain a point estimate $\hat{y}$ by averaging over the predictions $f_i(x)$ of N base estimators (Ensembles), forward passes (MC-Dropout), or sampled models (HBLL):

$$\hat{y} = \frac{1}{N} \sum_{i}^{N} f_i(x) \tag{4.1}$$

Post hoc Probability Calibration. Post hoc calibration methods refer to approaches that correct probabilities by using a calibration dataset. These methods do not provide uncertainty estimation of the parameters. However, they provide a first-order uncertainty estimation of the binary prediction (Bengs et al., 2023). In this work, PS is used to calibrate the outputs from the baseline and train-time uncertainty quantification approaches. Chapter 2.9 provides an introduction to post hoc calibration with PS.

Model Evaluation. All performance metrics were calculated from predictions on a test dataset for each model and HP tuning strategy or probability calibration method. AUC (↑) scores were obtained to assess the model´s ability to correctly classify samples. In addition, we assessed whether the generated models were capable of producing calibrated probability predictions. We used the BCE loss

(↓), the BS (↓), and two types of calibration errors, including the ECE (↓) and ACE (↓), to measure the probability calibration of the models.

In the context of probability calibration, it is important to distinguish between proper and improper scores. While a good proper score implies an overall good model, this is not necessarily the case with improper scores. The definition of proper scores can be found in Section 2.2.2. Proper scores include the BCE loss and the BS. In contrast, calibration errors are improper scores that do not consider the refinement but measure only the calibration error instead. The ECE is commonly used in literature to assess if a model is calibrated. In addition, we also used the ACE, which has some desirable properties making it more robust towards skewed distributions of the predictions, as described below. The theoretical background of both calibration errors is shown in Chapter 2.2.2.

It is important to highlight that, because these calibration errors are improper scoring rules (Gneiting and Raftery, 2007), predictions with zero calibration error are not necessarily good predictions. A model that always predicts the overall ratio of class instances in the dataset will be perfectly calibrated, despite its poor accuracy. Nevertheless, these scores are useful in assessing and comparing the probability calibration of models, which aligns with the goals of our paper. Consequently, we will employ these scores alongside proper scoring rules to evaluate both the calibration and overall performance of the models.

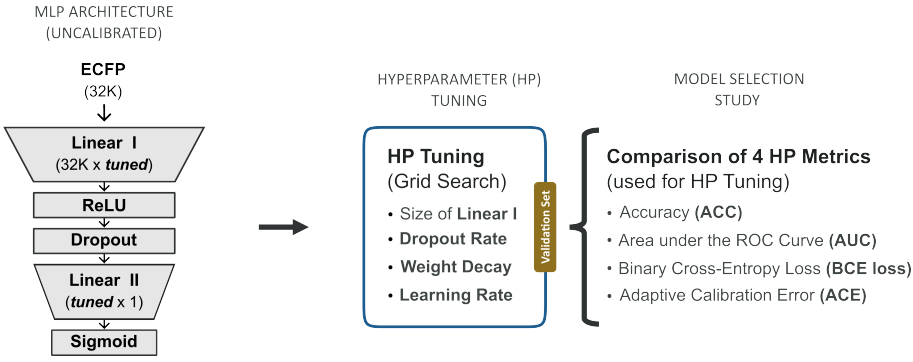

Figure 4.2: **Overview of the architecture of the MLP baseline model and the HP tuning workflow.** The size of the hidden layer and the dropout rate, as well as the weight decay and learning rate used during training, were tuned in a grid search using a validation dataset. Four different HP optimization metrics (HP metrics) were used, and the performances of the respective models were compared in a model selection study.

Experiments

For the sake of repeatability and estimation of the SD of the predictions, we generated ten model repeats with random seeds for each model type and averaged the resulting repeats. Note that, for computational reasons, only five repeats were generated for Ensemble models, resulting in 250 training sessions per target. The statistical significance of the best results was tested in each experiment by performing a two-sided t-test with a threshold of $p = 0.05$. Paired t-tests were used when comparing the baseline model with modifications of this specific model or between these modifications, including MLP plus PS (MLP+PS), MC-Dropout, HBLL, and HBLL plus PS (HBLL+PS). In all other cases, an unpaired t-test was used.

Model Selection Study. Since it is assumed that model overfitting affects probability calibration, we assessed the impact of different HP optimization metrics (HP metrics) on model calibration. To do so, we compared the calibration errors of models with HPs either maximizing accuracy or the AUC value or minimizing the BCE loss or the ACE. We assessed how the AUC, the calibration errors, and the BS were affected by varying HP metrics. Furthermore, we compared if the results of this analysis vary across targets.

Model Calibration Study. We studied the ability of three uncertainty estimation techniques and one post hoc calibration approach to generate predictions with smaller calibration errors than the baseline, indicating better uncertainty estimates. Figure 4.3 gives an overview of the methods assessed in this experiment.

In all cases, we build on an uncalibrated MLP baseline model, which we also refer to as the baseline model in this work. For the post hoc calibration method, PS, the validation dataset was used to fit a logistic regression to the generated scores. In the following sections, we will refer to this model as MLP+PS, Moreover, we assessed two ensemble-based uncertainty estimation approaches: MLP ensemble models, which are referred to as Ensembles, and MC-Dropout. For the generation of Ensemble models, 50 base estimators were trained with random initialization, whereas for the MC-Dropout approach, 100 predictions were generated using dropout during the forward passes. In both approaches, the predictions were averaged to obtain a prediction for a test instance. Finally, we used our proposed method HBLL by removing the last layer of the baseline MLP and replacing it with a Bayesian logistic regression model. The time

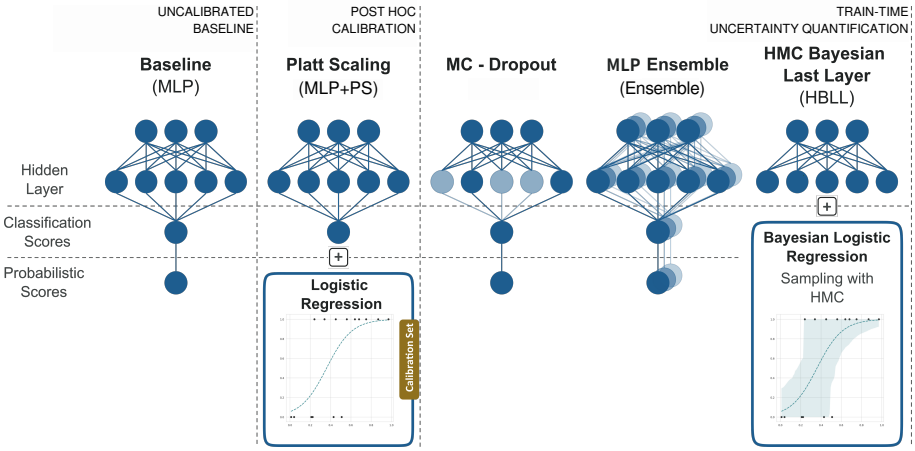

Figure 4.3: **Overview of model architectures assessed in the model calibration study.** The baseline model (MLP) was compared to the post hoc calibration method PS (MLP+PS) and the Bayesian approaches MC-Dropout and Ensembles. Furthermore, the proposed Bayesian approach HBLL was included in the analysis. The models were trained on the training dataset. For the post hoc calibration approach, the validation dataset was used to fit the logistic regression model.

required to train the individual models is reported in Section B.1.4 of the Appendix.

The parameters for the logistic regression model were sampled from their true posterior distribution using HMC. To generate the HBLL models, a vanilla HMC approach with one sampling chain was chosen with 500 burn-in and 1000 samples. The trajectory length and integration timestep were chosen based on preliminary runs and fixed to $L = 1200$, $\epsilon = 0.01$. No additional adaptation method, such as a NUTS, was used. The inverse mass matrix was set to the Hessian corresponding to the MAP parameter estimate of the logistic regression model. Note that, similarly to Ensemble and MC-Dropout, the training of the HBLL model is carried out on the training set and does not use a calibration set. The validation set was used to tune the precision of the Gaussian prior of the model parameters. The parameter space considered during tuning the precision is reported in Table B.8 in the Appendix. The selection of this single scalar parameter has an analogous effect as regularization and positions the method between the two main types of methods discussed so far. Again, calibration errors, BS, and AUC scores were used to compare the performance across model architectures and targets.

As a second step, post hoc calibration and UQ methods were combined to assess whether the model's probability calibration would benefit from first quantifying the uncertainty inherent in the predictions and subsequently calibrating the uncertainty estimates. The architecture of the combined models is illustrated in Figure 4.4. We applied PS to the Ensemble and HBLL model, by fitting a sigmoid function to the logit scores of the predictions, which resulted in the PS UQ models Ensemble+PS and HBLL+PS. Since PS does not affect the AUC scores of the predictions, only calibration errors and BS were calculated to compare the models' performance with their PS counterparts.

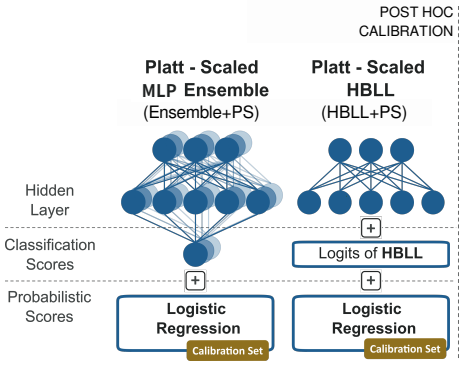

Figure 4.4: **Architecture of the uncertainty estimation approaches combined with PS.** The architectures of Ensemble+PS and HBLL+PS are shown. For generating PS UQ methods, a sigmoid was fit to the logits of the Ensemble and HBLL model. An additional calibration dataset was used for the calibration step.

4.2.2 Results and Discussion

In the following sections, we evaluate several hyperparameter tuning strategies to assess their ability to produce calibrated models. Furthermore, we evaluate different performance metrics, such as the calibration errors, to assess the ability of various probability calibration and uncertainty estimation approaches to generate high-quality uncertainty estimates. These experiments will identify practices that can generate well-calibrated ML models.

Model Selection Study

Calibration of Model Selection Strategies across Targets. We investigated the calibration of baseline models with four different HP settings, each of which was selected to optimize accuracy, AUC, BCE loss, and ACE. Figure 4.5 illustrates the results of the model selection study across all targets and HP metrics. Figure 4.5 shows comparable patterns in the ECE and ACE results across targets and hyperparameter metrics, allowing us to analyze them collectively as calibration errors throughout this section. Note that the ACE was slightly smaller than the ECE for all models, as detected in the numerical results in Section B.1.1 in the Appendix. A reason for this observation could be the high variance of the mean predictions in less populated bins, contributing considerably more to the ECE than other bins and leading to an overestimation of the calibration error. In summary, models tuned using the BCE loss, and ACE performed better in terms of calibration error than those tuned on the accuracy or AUC values. More specifically, the test set performance of the individual models on the CYP3A4 dataset shows that the model optimizing the ACE and BCE on a validation set results in the best calibration errors, as illustrated in the first row of Figure 4.5. In contrast, optimization based on AUC values leads to the highest calibration errors. The overall trends detected in the analysis of the CYP3A4 dataset could also be observed with the other two targets. In general, optimizing the HPs with regard to BCE loss or ACE resulted in models that exhibited smaller calibration errors than optimization based on accuracy or AUC. In more detail, HP optimization using the ACE was favorable for the MAO-A dataset, while for the hERG dataset, optimization with the BCE loss led to the significantly smallest calibration errors.

Since both calibration errors are improper scores and can give good results for inaccurate models, it is advisable to also consider proper scores when analyzing model calibration. The bottom row of Figure 4.5 shows the results of different HP metrics across the three datasets in terms of the proper score BS and the AUC. The BS summarizes the performance of both the calibration of the model and its ability to rank predictions correctly, while the AUC scores solely measure the ranking abilities of the models. Again, the models optimized based on BCE loss and ACE yielded the best BS and AUC values. For targets CYP3A4 and hERG, choosing the BCE loss as HP metric lead to the significantly best BS and AUC scores. Model trained on the MAO-A target, on the other hand, achieved the best BS and AUC results when optimizing for ACE.

Therefore, the BSs, and the AUC values confirm the outcomes obtained in the calibration error analysis, favoring HP selection strategies based on BCE loss or ACE rather than accuracy or AUC. This observation could be explained by reduced model overfitting when choosing ACE or BCE loss as HP metric.

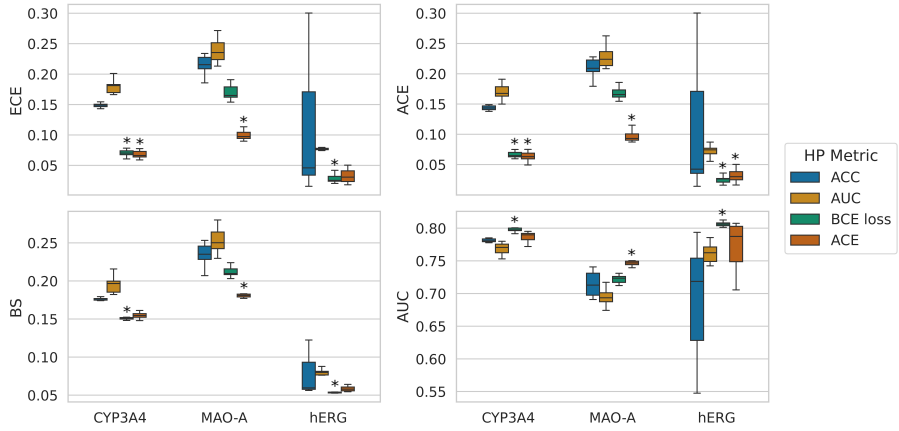

Figure 4.5: **Results of the model selection study for all targets.** The performance of the models is evaluated on the test set. Calibration errors (upper row), as well as BSs and AUC values (bottom row), are shown across targets for all HP tuning strategies. Results are averaged over ten model repetitions, except for the Ensemble models, for which five model repeats were computed. For each performance metric, an asterisk marks the best model and any others that are statistically indistinguishable from it. Statistical significance was determined in a two-sided t-test ($p < 0.05$).

This assumption is supported by the results of an additional experiment, which investigated the model performance on the training set compared to the test set to demonstrate the overfitting behavior of the models. The study results are displayed in Table B.6 in the Appendix. The experiment showed that the models with HPs optimizing the AUC or the accuracy perform best on the training set in terms of AUC, BS, and ACE across all targets. However, when looking at the test performance, the models with HPs optimizing BCE and ACE obtained the best results. The good performance on the training set and comparatively poorer results on the test set imply a more pronounced overfitting of the models with HPs optimizing the accuracy or AUC. The reduced overfitting of models tuned using the BCE loss or ACE could be the result of the regularizing effect of the calibration term which contributes to these scores, as shown in Chapter 2.3. Since accuracy and AUC scores do not account for probability calibration, models tuned to optimize these scores on the validation datasets perform worse on the test set.

ACE vs AUC across Model Selection Strategies. Figure 4.6 illustrates the performance of the model selection strategies in terms of AUC versus ACE scores. This allows for identifying the most suitable HP metric for each situation, enabling the user to select a model that delivers optimal test performance for a specific aspect (calibration, ranking abilities, or any custom-weighted combination thereof). Not that, while there is, in general, a trade-off between calibration and AUC, there is a clear Pareto dominance observed for MAO-A and hERG. Figure 4.6 shows that ACE as a metric clearly dominates every other HP metric in the case of the target MAO-A, while for hERG using the BCE loss leads to the best-performing model. In the case of CYP3A4, a slight trade-off can be detected since the BCE loss is better in terms of AUC and ACE. However, the differences in ACE between the model tuned with these two HP metrics are not statistically significant.

These results identify BCE loss as the best choice for HP optimization with regard to BCE loss or AUC. Surprisingly, models tuned to optimize AUC value on a validation dataset also showed worse AUC performance on the test set with HPs optimizing BCE loss or ACE. This observation could result from increased overfitting of the models tuned with metrics that do not consider the calibration error, as reported in the previous section. Due to these results, we will focus on models with HPs that minimize BCE loss in the subsequent sections of this paper. Furthermore, these models achieved the best BS in two targets and the second-best BS for the MAO-A dataset.

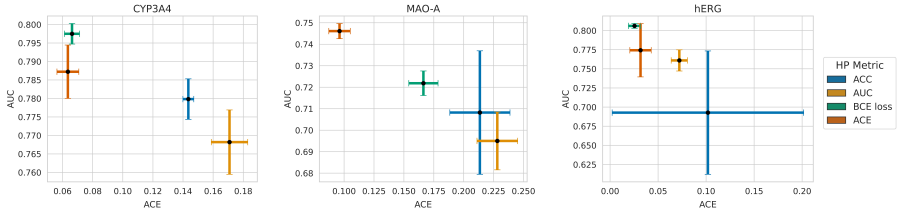

Figure 4.6: **ACE vs AUC across model selection strategies.** The ACE of models trained using different HP metrics is plotted against their AUC scores for each target. The performance of the models is evaluated on the test set. Points represent the average values, and error bars indicate the SD based on 10 model repetitions.

Model Calibration Study

We compared two popular uncertainty estimation methods and one common probability calibration approach, including the post hoc method PS and the train-time uncertainty estimation techniques Ensembles and MC-Dropout. For the sake of a clear and straightforward comparison, we only considered models with HPs selected to minimize the BCE loss in this probability calibration study. The results of models optimizing other HP metrics are listed in Section B.1.1 in the Appendix.

Calibration of Models across Targets. Figure 4.7 depicts the performance for every target across the model architectures. The MC-Dropout models performed much worse than the other architectures, and for the sake of a clearer comparison, these results are excluded from Figure 4.7. The numerical results of all models, including MC-Dropout, can be accessed in Section B.1.1 in the Appendix.

Both calibration errors considered in this study lead to similar results regarding the performance of different model architectures. Besides MC-Dropout, all models achieve lower calibration errors than the baseline model for CYP3A4 and MAO-A. However, only MLP+PS and HBLL achieve a statistically significant improvement. For hERG, only HBLL significantly improved the calibration errors compared to the baseline MLP. Similar to the calibration errors results, the analysis of the proper score BS showed that MLP+PS and HBLL significantly outperformed other approaches for CYP3A4 and MAO-A. For hERG, no significant difference could be determined between baseline MLP, Ensemble and HBLL. Interestingly, these three approaches outperformed Ensemble in terms of BS. The analysis of the AUC scores showed only minor differences below 0.03 between the model architectures across all targets, which were not statistically significant in the majority of cases.

The results of the models using other HP metrics also support the conclusion that all included methods retrieve similar AUC values, as shown in Section B.1.1 in the Appendix. An exception to this is MC-Dropout, which performed poorly in some cases. Based on these results, we can conclude that the analyzed methods improve the probability calibration by decreasing the calibration errors compared to the baseline model while having a limited impact on the accuracy of the ranking abilities. In this context, interesting findings were reported by Roth and Bajorath (2024), who investigated the relationship between accuracy and calibration in a variety of classification models. Similar to our conclusions, the authors reported that despite their large differences in the calibrating performance, the models overall produced stable and accurate predictions.

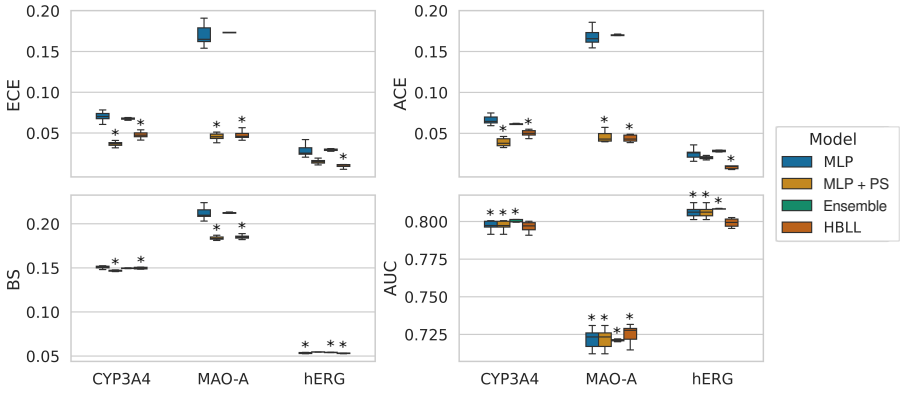

Figure 4.7: **Results of the model calibration study for all targets.** The performance of the models is evaluated on the test set. Calibration errors (upper row), as well as BSs and AUC values (bottom row), are shown across targets for all model architectures. Results are averaged over ten model repetitions, except for the Ensemble models, for which five model repeats were computed. For each performance metric, an asterisk marks the best model and any others that are statistically indistinguishable from it. Statistical significance was determined in a two-sided t-test ($p < 0.05$).

In conclusion, PS (MLP+PS) and HBLL are most powerful in terms of probability calibration as shown in Figure 4.7, with the latter resulting in the best calibration error for the MAO-A dataset and hERG. For the hERG dataset, HBLL performed significantly better than all other approaches. HBLL consistently outperformed all other train-time uncertainty estimation approaches across all performance metrics, except for AUC, where the Ensemble model achieved the best results in two targets. These results show that HBLL reaches the calibration error performance of state-of-the-art UQ and probability calibration methods and is only outperformed in one dataset by PS. Furthermore, common train-time UQ methods, such as Ensembles or MC-Dropout, do not reach the performance of HBLL or PS in terms of calibration errors and BS. A possible explanation for this might be that the rather simple MLP architecture leads to a less complex loss landscape and fewer local minima, resulting in similar base estimators of the Ensemble model. Small NNs with fewer neurons and fewer layers have comparably fewer parameters. With parameter count D, the loss landscape forms a D-dimensional hypersurface, which becomes more complex and has more local minima as D increases. Furthermore, Ensembles and MC-Dropout were reported to lead to less confident predictions, resulting in more uncertain predictions shifted towards 0.5 (Rahaman and Thiery, 2021).

Therefore, these methods do not necessarily improve probability calibration. As a consequence, we hypothesize that ensembling techniques are only beneficial if the base estimators are overconfident, and the predictions are pushed towards 0 or 1. However, they are ineffective for underconfident or well-calibrated models and can even impair the quality of the uncertainty estimates. Overparametrization of models has been identified as one of the factors contributing to poor model calibration (Wang, 2025). Given that the models used in this study were rather shallow and their calibration properties were prioritized during hyperparameter tuning, including the size of the hidden layer, the resulting base estimators might not suffer from overconfidence in the same severity as it is known for larger deep NNs.

ACE vs AUC across Models. We compared the AUC and ACE of the considered model architectures to determine which model is the best choice with regard to different evaluation metrics and to identify potential trade-offs between these metrics. Figure 4.8 illustrates AUC values plotted against ACE for all model architectures. Again, the MC-Dropout models were excluded from the plot due to their bad performance. The results for hERG reveal a trade-off between ACE and AUC values across the model architectures. In the plot, the results of the individual models form a Pareto front, with Ensemble performing best in terms of AUC and HBLL in terms of ACE. In this case, it is up to the model user to prioritize either model calibration or ranking abilities to determine a suitable model. The results for the other two models were noisier, and their patterns were not as clear. For CYP3A4, different models performed best with regard to different evaluation metrics. While MLP+PS was best in terms of ACE, Ensemble resulted in the best AUC scores. The analysis of the MAO-A models showed that one single model, namely HBLL, outperformed all others in terms of both metrics (Pareto dominant). However, it is important to note that the AUC differences between many models are not statistically significant, as mentioned in the previous section.

Post hoc Calibration of Uncertainty Quantification Methods. PS is a post hoc calibration method, which makes it versatile as it can be applied to any model after training. We combined PS with Ensemble and HBLL to assess if calibrating uncertainty estimates obtained from Ensemble modeling or HBLL enhances model calibration. The results for Ensemble+PS and HBLL+PS are shown in Figure 4.9. Since PS does not change the ranking of the predictions, the AUC is excluded from this analysis. Furthermore, since the ECE and ACE resulted in similar results in the previous sections, we only report the ACE in Figure 4.9 to evaluate the probability calibration of the models. In

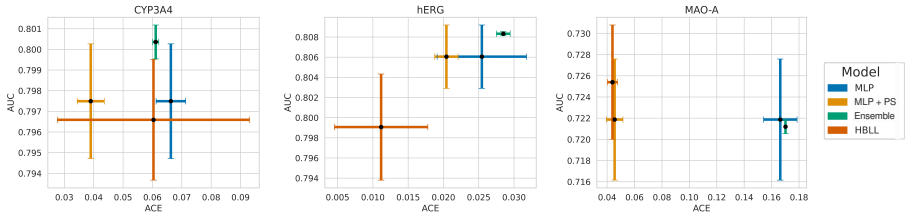

Figure 4.8: **ACE vs AUC across model architectures.** The ACE scores of different model architectures are plotted against their AUC scores for each target. The performance of the models is evaluated on the test set. Points represent the average values, and error bars indicate the SD based on 10 model repetitions

general, the PS models MLP+PS, Ensemble+PS, and HBLL+PS outperformed all train-time approaches across all targets. The only model that matched the performance of the PS approaches was the HBLL model, which was only significantly outperformed by its PS counterpart in terms of ACEs on the MAO-A dataset. The combined methods were reported to significantly improve calibration only in one of the three assays (MAO-A). In this case, the modified HBLL+PS achieved the lowest calibration errors while all other models performed significantly worse. In addition, HBLL+PS also generated the smallest BS, with no significant difference to the other PS models MLP+PS, Ensemble+PS, and the train-time uncertainty estimation approach HBLL. The results for target CYP3A4 show that, again, all PS models performed best, with Ensemble+PS resulting in the lowest calibration errors and BS. The only exception was the HBLL model, which consistently matched the performance of the calibrated models for CYP3A4. The HBLL model also resulted in the best performance of the hERG model, significantly outperforming PS models in terms of calibration errors.

Interestingly, PS of the Ensemble model led to improved calibration errors across all targets, while the difference between the HBLL and HBLL+PS models was much smaller. In some cases, PS of the HBLL models did not lead to any improvements at all. A possible explanation for this difference between the two UQ methods could be that tuning the single hyperparameter needed for HBLL model generation already has a calibrating effect. Hence, HBLL models are already well-calibrated and do not need an additional calibration step. As reported in previous sections of this paper, ensembling failed to produce predictions with lower calibration errors than the baseline in most cases. However, the calibration errors improved after applying the PS step.

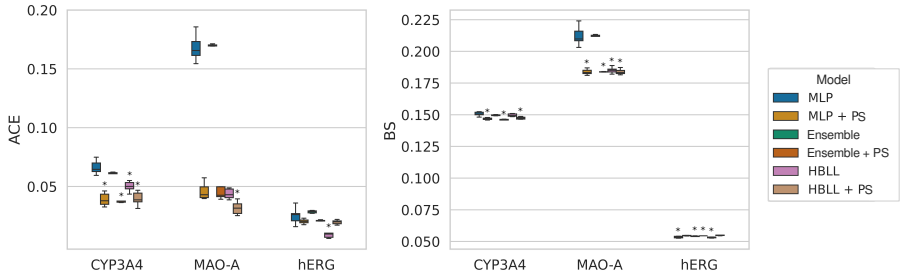

Figure 4.9: **Results of post hoc-calibrated UQ methods.** The performance of the models is evaluated on the test set. Calibration errors, BSs, and AUC values are shown across targets for selected uncertainty estimation methods and their PS counterparts. Results are averaged over ten model repetitions, except for the Ensemble models, for which five model repeats were computed. For each performance metric, the results of the best model are bold and underlined. All other bold results are statistically indistinguishable from the best result as reported in a two-sided t-test ($p < 0.05$).

4.3 Fully Bayesian NN using HMC Sampling for UQ in Drug-Target Interaction Predictions

This preliminary work aims to make the following contributions:

- We use HMC sampling to generate Bayesian NNs that model drug-target interactions of three biological targets. The use of HMC facilitates sampling from the true posterior distribution.
- We analyze several diagnostic metrics to examine the convergence and mixing abilities of the HMC chains. These metrics allow us to shed light on the sample generation process and to identify pathological behaviours of the chains.
- Finally, we compare the HMC-based Bayesian NN to other common UQ methods for drug-target interaction modeling.

In recent decades, Bayesian approaches have gained increasing attention in the ML community (Gal, 2016; Wilson and Izmailov, 2020), offering improved predictions and more informative uncertainty estimates than conventional non-Bayesian methods. Major efforts have been directed towards approximating the

Bayesian setting by either using a simple surrogate distribution that aims to mimic the posterior distribution, as in variational approaches (Chapter 2.5.2: e.g. Blundell et al. (2015)), or by representing only the posterior modes with ensemble models (Chapter 2.6: e.g. Lakshminarayanan et al. (2017)). Because the posterior distributions of NNs are multi-modal and very complex due to their high dimensionality, it is questionable if these approaches can accurately represent these posterior distributions.

The HMC approach generates a chain of samples from the posterior by using the Hamiltonian framework to construct geometry-informed trajectories. The HMC samples are guaranteed to converge to the true posterior under specific conditions (see Chapter 2.5.3), and the method has been reported to generate the most accurate posterior distributions compared to variational approaches or ensemble methods (Izmailov et al., 2021). Izmailov et al. (2021) generated HMC-based Bayesian versions of modern NN models and reported that the HMC-based Bayesian NN outperform conventionally trained models and MLP ensembles. Furthermore, the authors emphasize that although the HMC method might not be practical for everyday life applications due to the high computational cost, it provides important insights into the geometric properties of posterior distributions.

Several papers exist that take a HMC-based Bayesian NN approach to obtain a ground truth for uncertainty estimation (Schweighofer et al., 2023) or to improve uncertainty estimates (Vergara et al., 2018; Sheinkman and Wade, 2025; Benker et al., 2021). In addition, alternatives to the classic HMC sampling have been proposed to make the samplers more computationally efficient. For example, stochastic gradient HMC can reduce the computational effort, specifically when dealing with large datasets, as the gradient of the potential energy is approximated using mini-batches and additional noise and friction terms to correct for stochasticity.

So far, HMC has only been rarely applied to drug discovery-related ML problems. Gao et al. (2025) shows that a stochastic gradient HMC method outperforms ensemble-based methods to estimate the uncertainty in molecular property prediction tasks using graph NN. To date, a full-batch HMC approach for QSAR models is, to our knowledge, still missing from the literature.

In this preliminary work, we aim to close this gap by using full-batch HMC to generate Bayesian NN for QSAR models trained on three different tasks. This study has two primary purposes: first, the simulation will give insight into the behaviour of HMC samplers and how to construct them to efficiently navigate through posterior spaces. Second, the generated HMC models are intended to serve as ground truth to evaluate the performance of other UQ methods, like ensemble methods or variational approximation approaches.

4.3.1 Methods

Data

The details regarding the extraction and processing of the data used in this study can be found in Section 4.1.1. Target-specific bioactivity data was extracted and processed from ChEMBL (Mendez et al., 2018). This included drug-target interaction data for MAO-A, as well as CYP3A4 and hERG, which play important roles in the metabolism and toxicity profiling of a candidate compound. Assay-specific thresholds were chosen to create labels for binary classification of the compounds. More details about the assays and the thresholds are shown in Table 4.1. The data was split into three folds based on their chemical structure using a cluster-based approach.

Model Generation

Pytorch (Paszke et al., 2019) Version 2.1.0 was used to train the baseMLPs, the ensemble-based, and the BBB models. Sampling with HMC was carried out using the PyTorch-based package hamiltorch (Cobb et al., 2023).

Multilayer Perceptrons. The MLPs were taken as the basis for all other UQ methods evaluated in this study. Compared to the study presented in Section 4.2, a smaller ECFP size of 4096 was used to reduce the computational effort of the HMC sampling process. The MLPs in this study resembled the model architecture described earlier in this thesis (Section 4.2.1). Since unbounded activation functions, such as ReLU, have been reported to lead to poorly behaving chains (Sommer et al., 2024), Tanh was used to avoid issues during HMC sampling.

The HPs of the base MLPs were optimized using the BCE loss on a validation dataset. The number of layers, the size of the hidden layers, the weight decay, and the learning rate were tuned in a grid search. A table with the screened hyperparameter space, as well as the best parameter configuration corresponding to the smallest BCE for each target, can be found in Section B.2.1 of the Appendix. Since UQ with sampling approaches does not include the use of dropout during training, the dropout rate of the models was set to 0 to ensure a fair comparison.

HMC-Bayesian NNs. The theoretical foundation of HMC is provided in Section 2.5.3 of this thesis. The architecture of the base MLP for the corresponding targets was used as a template to train Bayesian NNs using HMC samplers. For each target, 4 chains with 10000 samples were generated.

In general, the quality of the samples is highly dependent on the behaviour of the chain (Section 2.5.3). Therefore, the HPs of the HMC sampler were carefully tuned based on the acceptance rate of the samplers. This included the step size ϵ and number of steps, which were chosen based on the acceptance rate and autocorrelation between samples. For CYP3A4 and hERG a step size of 0.0013 was selected, while for MAO-A the step size was 0.0005. For all targets, the 1000 integration steps were performed before retrieving a new sample. Furthermore, the weight decay of the MLP training was used as a basis for the prior precision τ. Because the MLP was trained using the mean loss, while the log-likelihood in the HMC sampling was computed using the sum of the losses, the weight decay was multiplied by the size of the training dataset to obtain the correct value of τ.

To estimate the variance of HMC performance, cross-validation using importance sampling was applied (Gelman et al., 2013). The concept of importance sampling is explained in detail in Section 2.5.3. In short, the importance sampling approach obtains a weight w_s that reweighs every parameter sample s. To quantify the performance variance of HMC models, Z sets of weights were generated to generate Z versions of the HMC model with differently weighted parameter samples. To do so, the training data D_{tr} was divided into Z equal-sized subsets $D_{tr} = \{D_1, \ldots, D_Z\}$. Subsequently, the importance weight of the sample s was calculated as follows:

$$w_s^z = \frac{\mathbb{P}(D_z|\theta_s)}{\sum_{z=1}^{Z} \mathbb{P}(D_z|\theta_s)}. \tag{4.2}$$

For each data subset D_z, the likelihood given parameters sample θ_s was calculated and normalized, as shown in Equation 4.2. This approach enabled the generation of Z test set predictions, which were subsequently used to get an estimate of the SD.

Experiments

Evaluation of HMC Samplers. Several diagnostic metrics were analyzed to assess the behavior of the chains. These metrics are explained in detail in Chapter 2.5.3. First, the autocorrelation plots were generated for each target individually to assess the mixing of the chains. Second, trace plots of the model

parameters were obtained to check the convergence of the chain and to identify the number of burn-in samples. After eliminating the burn-in samples, the $\hat{R}$ and ESS of the remaining samples were calculated in the last step to check the convergence of the chain as well as the effective number of uncorrelated samples in the chain. Due to the high negative autocorrelation of consecutive samples, thinning was applied for the samples retrieved for the hERG and CYP3A4 models, providing a lower bound of the ESS.

Comparison of UQ Approaches. Besides HMC-based Bayesian NN, three additional UQ methods were evaluated based on the tuned base MLPs. These methods included the variational approximation technique BBB (Blundell et al., 2015), as well as Bayesian-like sampling approaches, like MLP ensembles (Lakshminarayanan et al., 2017), and MC-Dropout (Gal and Ghahramani, 2016). The MLP ensembles were generated by averaging the predictions of 10 base estimators. For evaluation of MC-Dropout, models with dropout rates of 0.25 (MC-Dropout [25]), 0.5 (MC-Dropout [50]), and 0.75 (MC-Dropout [75]) were generated, and the predictions of 400 forward passes were averaged. We refer to Sections 2.5.2 and 2.6 for a theoretical explanation of these methods. For all models, ten repeats were used to report the mean performance and the standard deviation.

4.3.2 Results and Discussion

Evaluation of HMC Samplers

The convergence and mixing behavior of the chains was analyzed using trace plots and $\hat{R}$ values, as well as the autocorrelation between the samples, and the ESS. For each model, the diagnostics were analyzed separately for the weights and biases of each layer, which we will refer to as parameter subsets.

Convergence of the Chains. The convergence of the chains was assessed using trace plots that show the parameter values in the order drawn by the HMC sampler. Figure 4.10 shows the trace plot of the hERG model separately for each parameter subset. The traceplots for the remaining assays are shown in Figures B.1 and B.2 in the Appendix. The figures reveal that convergence is reached for the weights W^1 in the first layer for all targets. However, the HMC samplers fail to achieve convergence for parameter subsets close to the output, especially for the biases, as indicated by the unstable values of the parameters in the last layer (Figure 4.10). The NN modeling CYP3A4 and hERG consist of only two layers, so that convergence is only achieved for the weights W^1 in

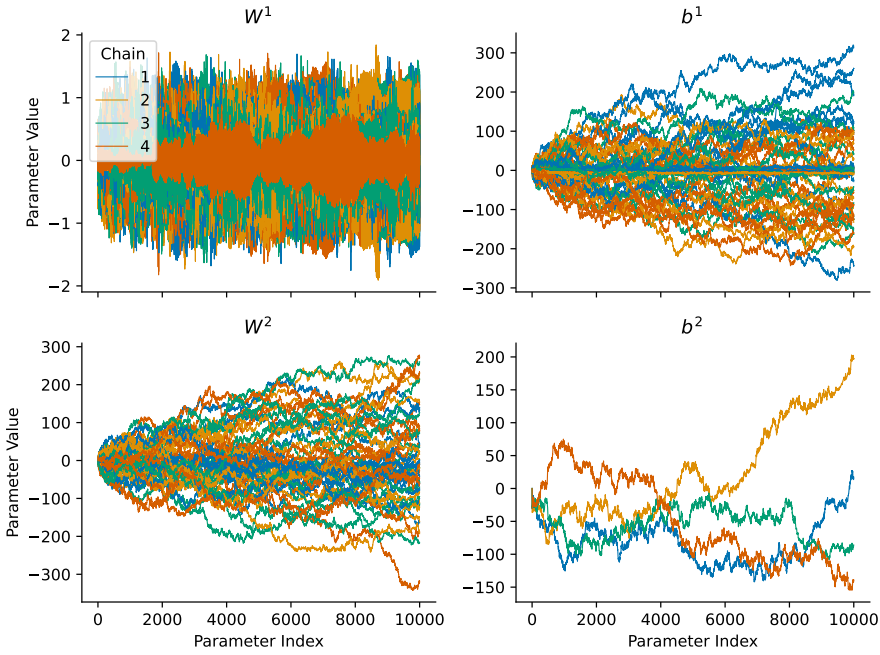

Figure 4.10: **Traceplot of the hERG assay.** The traceplot of the parameter samples for the hERG model is shown. The values for each parameter subset are shown in separate plots. The values are plotted for evolving HMC chains, indicated by increasing parameter indices. The chains responsible for the generated sample are marked by color.

the first layer. Contrarily, the MAO-A model, which consists of three layers, shows a somewhat improved convergence for the weights W^2 and biases b^2 in the second layer (Figure B.1). However, similarly to CYP3A4 and hERG, the parameters of the last layer (W^3 and b^3) are again not converged. The first 4000 samples were specified as burn-in and therefore eliminated from the chains.

To quantify the convergence of the HMC chains, the $\hat{R}$s of the chains without the burn-in samples were calculated using the between-chain and within-chain variance. Figure 4.11 shows the $\hat{R}$ values for each parameter subset across all assays. Recall that HMC samplers with $\hat{R}$ values below 1.01 are considered converged. As expected, the $\hat{R}$ values of parameter groups near the input of the NN confirm convergence of the chain, while for parameters closer to the output, $\hat{R}$ deteriorates. Thus, the analysis of $\hat{R}$ supports the conclusions of the trace plot analysis.

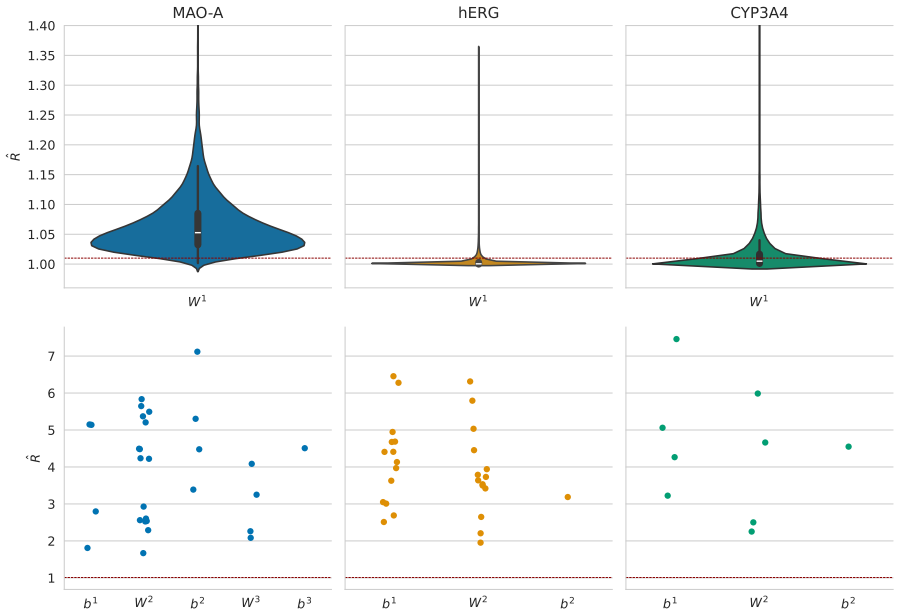

Figure 4.11: $\hat{R}$ **of the HMC samples across assays.** The $\hat{R}$ the individual parameter subsets are shown for each assay. A red line marks $\hat{R} = 1.01$, which is the threshold that indicates a convergence. The first row shows the $\hat{R}$ of the weights in the first layer (W^1) across assays. In the second row, the $\hat{R}$ values of the remaining parameter subsets are plotted.

Mixing of the Chains. Autocorrelation plots of the chains were generated to assess the mixing of the HMC chains. Figure 4.12 illustrates the autocorrelation of the first chain for each target. The autocorrelation of the other chains is plotted in Figure B.3 in the Appendix. For all targets, the weights of the first layer W^1 are mixing well, as implied by low autocorrelation at increasing lags. However, the autocorrelation of the remaining parameters is high, even at large lags. This result indicates that the chains are mixing slowly for these parameters. This is also reflected by the ESS for the parameter subsets illustrated in Figure 4.13. While the number of uncorrelated samples is large for parameters of the first layer weights W^1, the ESS for the other parameter subsets is in the low double- or three-digit range. The comparison of the ESS values of the individual targets shows that especially the chains of the MAO-A model retrieved a low number of uncorrelated samples, with some parameters only reaching an ESS of 15 or less.

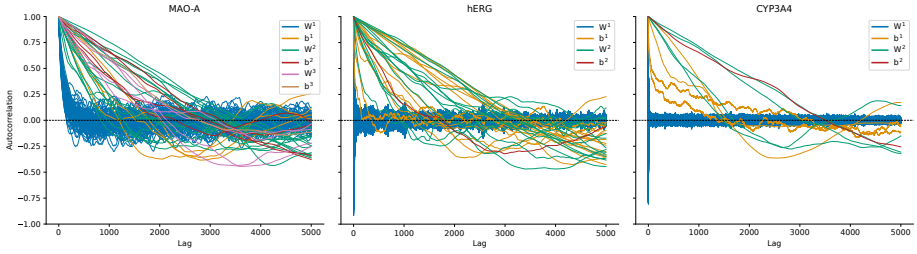

Figure 4.12: **Autocorrelation between HMC samples across assays.** The autocorrelation between the first 5000 HMC samples of the first chain is shown for each assay. Subsets of parameters are distinguished by color coding.

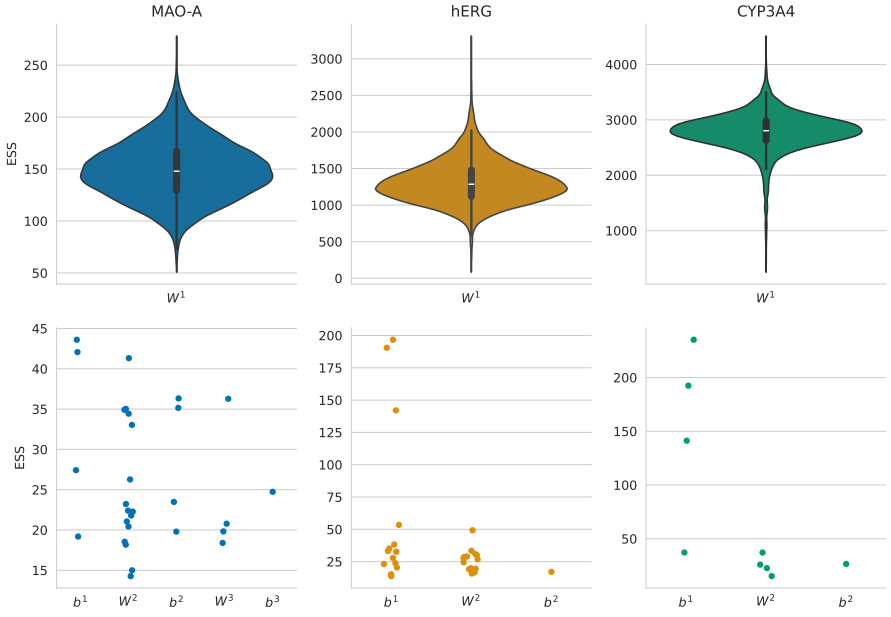

Figure 4.13: **Effective sample size across assays.** The ESS is shown for each assay. Subsets of parameters are distinguished by color coding. The first row shows the ESS of the weights in the first layer (W^1) across assays. In the second row, the ESS values of the remaining parameter subsets are plotted.

Comparison of HMC Sampling with other UQ methods

The HMC models were compared to other UQ approaches, including MC-Dropout with 25% (MC-Dropout [25]), 50% (MC-Dropout [50]), and 75% (MC-Dropout [75]) dropout rate, MLP ensembles (Ensemble), and BBB (Bayes by Backprop). Table 4.2 compares the performances of the individual models.

Table 4.2: **Comparison of model performance between the HMC model and other UQ methods.** The performance of the models is evaluated on the test set. The ACE, BSs, BCE, and AUC values are shown across targets. The mean and SD of 10 model repetitions is shown for the MLP, MC-Dropout, Ensemble, and Bayes by Backprop models. The variance of the HMC performance was estimated using importance sampling.

Target	Model	ACE	BS	BCE	AUC
CYP3A4	MLP	0.044±0.008	0.158±0.003	0.489±0.009	0.776±0.01
	MC-Dropout [25]	0.031±0.008	0.158±0.003	0.488±0.007	0.771±0.009
	MC-Dropout [50]	0.039±0.007	0.161±0.002	0.496±0.005	0.762±0.008
	MC-Dropout [75]	0.071±0.008	0.172±0.001	0.52±0.003	0.753±0.006
	Ensemble	0.042±0.006	0.158±0.002	0.487±0.006	0.779±0.007
	Bayes by Backprop	0.066±0.019	0.19±0.005	0.567±0.017	0.68±0.017
	HMC	0.0502±0.0001	0.1448±0.0001	0.4496±0.0001	0.8197±0.0001
MAO-A	MLP	0.122±0.025	0.212±0.018	0.639±0.061	0.67±0.044
	MC-Dropout [25]	0.067±0.025	0.191±0.009	0.565±0.021	0.708±0.027
	MC-Dropout [50]	0.074±0.007	0.201±0.006	0.589±0.015	0.667±0.026
	MC-Dropout [75]	0.083±0.014	0.217±0.001	0.627±0.004	0.51±0.048
	Ensemble	0.107±0.023	0.207±0.014	0.617±0.048	0.676±0.031
	Bayes by Backprop	0.118±0.036	0.216±0.015	0.644±0.042	0.657±0.035
	HMC	0.0611±0.0018	0.1792±0.0001	0.5307±0.0002	0.7512±0.0002
hERG	MLP	0.025±0.006	0.059±0.001	0.224±0.006	0.721±0.037
	MC-Dropout [25]	0.02±0.004	0.059±0.0	0.225±0.002	0.711±0.013
	MC-Dropout [50]	0.022±0.007	0.059±0.001	0.226±0.003	0.702±0.023
	MC-Dropout [75]	0.024±0.005	0.059±0.001	0.226±0.003	0.711±0.021
	Ensemble	0.026±0.004	0.058±0.001	0.221±0.006	0.74±0.032
	Bayes by Backprop	0.025±0.003	0.062±0.001	0.248±0.003	0.586±0.021
	HMC	0.0174±0.0004	0.0521 ± 0.0001	0.195 ± 0.0001	0.8209 ± 0.0001

The results show that in terms of AUC the HMC significantly outperforms all other models across assays, especially for the hERG assay, for which the difference in AUC between the HMC model and the second best performing Ensemble model is 0.08.

In terms of model calibration, the HMC model performs best in the case of the hERG and MAO-A models, indicated by low ACE scores. In the case of the CYP3A4 assay, the results are not as clear, as the HMC model only performs at the average level in terms of ACE. A reason for the different model performance between the assays could be the versatile nature of the CYP3A4 target. CYP

plays an important role in the metabolism of various drugs (Deodhar et al., 2020; Ioannides, 1996; Furge and Guengerich, 2006). It has highly promiscuous binding sites that can interact with structurally diverse compounds. Therefore, the decreased model performance of the CYP3A4 models could be due to a diverse dataset of active compounds, which makes the prediction task and the quantification of the predictive uncertainty harder.

For all models, the HMC approach also leads to the best BCE and BS scores. This is not surprising, given that the HMC approach performs best in terms of AUC and outperforms or matches the calibration performance of other UQ methods. Since the BCE and the BS are proper scoring rules that take into account calibration and accuracy of the model, the superior performance of HMC in terms of these scores was expected.

In conclusion, the HMC chains achieve convergence and good mixing for only parts of the model parameters, so that it remains questionable whether the chain is representing the posterior truthfully and if the HMC models can be considered ground truth for UQ. Nevertheless, the HMC models outperform other UQ approaches across assays. The better performance of HMC is reflected especially in improved AUC scores. This observation might indicate an underfitting of the models, which could be a result of the comparatively small ECFP and model size chosen in this study. This hypothesis is also supported by rather small improvements in calibration when using HMC sampling. The small differences between the calibration of HMC models and other UQ approaches might indicate that the models are already well-calibrated and are not overfitting due to their comparatively small size.

Chapter 5

Conclusion and Outlook

In this work, we explored UQ for ML approaches used for drug-target interaction prediction. More specifically, the key objective was to identify practices that yield reliable uncertainty estimates for bioactivity predictions, thereby supporting a more cost- and time-efficient drug discovery pipeline.

This chapter outlines the main contributions of this work in Section 5.1 and the general conclusions in Section 5.2. Section 5.3 concludes this thesis with a discussion of its limitations, remaining open questions, and potential avenues for future research.

5.1 Contributions

This thesis analyzes methods for UQ in NN-based QSAR models with the goal to obtain well-calibrated uncertainty estimates that reflect the true uncertainty associated with model predictions. Using internal and public datasets, we systematically evaluated various setups, including different UQ approaches and optimization strategies for model training, to achieve reliable uncertainty estimation for bioactivity predictions. Furthermore, we provided insight into how distribution shifts in the data can affect the probability calibration of common UQ methods.

5.1.1 Chapter 3: Comparing UQ Methods for Drug-Target Interaction Predictions using Real-World Pharmaceutical Data and Temporal Splits

The main focus of Chapter 3 is to compare and evaluate uncertainty estimation approaches for QSAR models in a practical, real-world context. A temporal splitting strategy was applied to internal data from a pharmaceutical company to simulate the evolution of drug-target interaction data. In pharmaceutical research, datasets are continually updated, requiring models to be retrained. It is therefore essential to assess whether a chosen uncertainty estimation strategy continues to deliver accurate uncertainty estimates as the data evolves or whether adjustments are necessary. Our findings offer valuable insights into uncertainty estimation and highlight the challenges posed by real-world applications. Furthermore, this study highlights the challenges introduced by real-world data, emphasizing the complexity of identifying effective strategies for uncertainty estimation in QSAR models.

Distribution Shifts in Real-World Pharmaceutical Data. The temporal splitting approach applied to the real-world pharmaceutical dataset allowed us to determine distribution shifts in the data over time. Sections 3.2 and 3.4 investigate these distribution shifts to shed light on the data evolution in descriptor and label space. The analysis showed that the distribution shifts over time strongly depended on the nature of the individual assays. The project-specific TB assays exhibited more pronounced distribution shifts, while the more abundantly used toxicity screens and assays assessing the pharmacodynamic properties (ADME-T assays) showed moderate shifts in descriptor space and little shift in label space over time. These results suggest that the iid assumption might not hold, especially for target-specific assays with larger shifts in label- and descriptor space. Note that real-world data is generally likely to exhibit distribution shifts over time, particularly in assay data, where chemical series are naturally present. Hence, we also expect the ADME-T data not to be completely iid. However, in the case of the ADME-T assays, the distributions between training and test are more similar than for TB assays.

Evaluation of UQ Approaches on Pharmaceutical Industry Data. The research included in Section 3.2 focused on UQ in classification tasks, while regression models were investigated in Section 3.4. In general, the Ensemble models and the BBB approach achieved well-calibrated models for both objectives. Given that training Ensembles require substantial computational

resources, BBB might be the best choice when striving for a fast method that yields well-calibrated estimates.

For many TB assays, where large distribution shifts over time where reported, the baseline classification MLP matched the performance of more sophisticated UQ methods used for binary classification. For regression models, on the other hand, Ensembles and BBB models improved model calibration as reported in Section 3.4 . Thus, these approaches might be able to generate better-calibrated results in a regression setting, when the actual bioactivity values can be accessed rather than less-informative binary labels.

Regarding the post hoc calibration of classification models (Section 3.2), PS and VA predictors were able to achieve better-calibrated results for some assays, while for others, the calibration stayed the same or even deteriorated after the post hoc calibration step. The calibrating capabilities were dependent on the distribution shift between the calibration and test sets, with declining performance when the shifts in the descriptor space increased.

In general, the uncertainty estimation approaches discussed in this study have been shown to improve model calibration on toy data or datasets that do not account for distribution shifts caused by data evolution over time. However, previous studies show that improved performance on iid data often fails to translate into better outcomes under distribution shift (Ovadia et al., 2019; Koh et al., 2021), which is also supported by the findings in this study.

Performance of UQ Approaches over Time. The analysis of UQ approaches over time revealed differences between classification (Chapter 3.2) and regression (Chapter 3.4) models.

For regression models, we showed that the best-performing models in terms of predictive accuracy and calibration are typically robust throughout time for ADME-T assays but not always for TB assays. As such, an evaluation of the best model for ADME-T assays can be trusted to hold without reevaluation, whereas TB assays may need to be reassessed occasionally.

Contrarily, the analysis of the performance of classification models over time revealed unstable model performance. For classification models, it is therefore difficult to draw conclusions from results from one point in the assay history to another. In conclusion, a reevaluation of classification model performance is required for all assays as soon as more recent data is added.

Disentangling Uncertainty Sources. The case study in Section 3.4 showed how the uncertainty estimates from the most effective models can be practically applied to inform and guide ongoing drug development efforts, offering valuable insights for risk management and decision-making. Herein, we found that the epistemic uncertainty estimates correlate well with theoretical assumptions, but that the aleatoric uncertainty estimates require further analysis to understand their relation to the underlying, inherent noise in the data.

Use of Censored Labels in Regression Tasks. The low-data challenge in QSAR modeling during drug discovery is typically accompanied by additional, often overlooked, partial information from censored labels. Despite their potential value, censored labels have not yet been fully utilized due to the lack of suitable methods to incorporate them during ML. Recognizing this gap, we have developed extensions to several established ML models, enabling them to effectively learn from censored labels while providing robust and reliable UQ. The results in Section 3.4 showed a particular advantage of including the censored labels during training in cases where a high enough proportion of the experimental labels are censored. Despite these enhancements, drawbacks such as potentially decreased predictive performance and overall performance on specifically observed labels should be considered before adopting our suggested approach. These side effects are likely a result of the commonly unbalanced nature of the censored labels and should be addressed in future work.

Including Aleatoric Uncertainty via Probabilistic Labels. In Section 3.3, we showed that using probabilistic labels in combination with probability calibration approaches can improve uncertainty estimation in RF models. In addition, we present a comprehensive analysis of how model calibration changes over time using temporal splits of internal data from a pharmaceutical company. Based on these preliminary results, further steps are needed to understand model calibration in a temporal setting. Furthermore, an extension to other model architectures might be useful to investigate if the same conclusions can be drawn for other model types.

5.1.2 Chapter 4: UQ in Drug-Target Interaction Models Using Hamiltonian Monte Carlo Sampling

The main focus of Chapter 4 is to use HMC sampling to obtain uncertainty estimates for QSAR model predictions. HMC is regarded as the gold standard for posterior approximation because it allows for efficient drawing of samples from the true posterior. Due to the high computational cost of HMC sampling, performing UQ with HMC remains challenging. In this work, we use a computationally efficient UQ approach with HMC using a partial Bayesian NN, in which only parts of the network are sampled. Furthermore, we present preliminary work on a fully Bayesian NN sampled with HMC for drug-target interaction prediction.

Improved UQ with HMC Bayesian Last Layer. In Section 4.2, we propose to use a computationally efficient Bayesian uncertainty estimation method named HBLL, which generates HMC trajectories to obtain samples for the parameters of a Bayesian logistic regression fitted to the hidden layer of the baseline NN. The HBLL approach is further compared to common probability calibration methods and train-time UQ techniques using public data. The results showed that HBLL was the only train-time UQ approach that improved probability calibration and could match or outperform other common uncertainty estimation approaches. Furthermore, the HBLL approach is a good compromise because of its reduced computational complexity compared to the full Bayesian treatment of the weights. Surprisingly, other train-time UQ methods failed to produce predictions with smaller calibration errors than the baseline, which might be a result of the previously reported inability of these approaches to calibrate non-overconfident models (Rahaman and Thiery, 2021).

In the last step of our analysis, we examined whether applying a post hoc calibrator to different uncertainty estimation methods improved model performance. To do so, we applied PS to the Ensemble and the HBLL model predictions. Interestingly, PS did not always improve model calibration. In general, calibration errors of the calibrated models were smaller when the train-time uncertainty estimation model did not improve probability calibration relative to the baseline MLP, which was often the case for the Ensemble models. On the other hand, the HBLL model already exhibited small calibration errors before the post hoc calibrating step. PS of these models produced smaller calibration errors only in one out of three targets, and it failed to retrieve significantly better BS for all three datasets.

Comparisons of Strategies for HP Tuning. In Section 4.2, we reported that the selection of the metric used for HP tuning substantially affects model performance, including model calibration. We observed that using metrics that took into account the probability calibration of the model not only resulted in smaller calibration errors but also in improved AUC scores.

Evaluation of HMC Samplers for full Bayesian NN. Section 4.3 provides a preliminary study of HMC sampling for generating full Bayesian NNs used for drug-target interaction modeling on public data. The analysis of the trace plots as well as the $\hat{R}$ scores revealed that while the chain converged well for the weights in the first layer, convergence was not reached for most other parameter subsets. This pattern was observed across all assays. A similar result was obtained for the mixing behavior of the chain. For all assays, the chain mixed well for the weights in the first layer, but failed to do so for the remaining parameter subsets.

Benchmarking HMC Sampling Against Established UQ Methods. Finally, the fully Bayesian HMC-based NN was compared with a traditional MLP model and other UQ methods, including MC-Dropout, MLP ensembles, and Bayes by Backprop. In terms of ranking abilities, the HMC approach retrieves better AUC values than all other models. The HMC models also achieve the best calibration performance, albeit with a smaller improvement compared to their gains in AUC. The comparatively small improvement in calibration relative to AUC may be due to the limited model size, which can lead to model underfitting.

5.2 General Conclusions

After focusing on different UQ methods in this work and applying them to various datasets, it can be concluded that there is no universal solution that will always yield the best uncertainty estimates. The best model is strongly dependent on:

- the dataset size
- distribution shifts in the data
- the architecture of the models
- the model objective as UQ methods perform differently in a classification and regression setting

Our results align with previously reported observations (Hirschfeld et al., 2020; Yin et al., 2023) that there is no one-size-fits-all solution, and the choice of UQ approach should be tailored to the downstream task for which the model will be used.

Furthermore, we report that Bayesian approaches can improve the uncertainty estimates of NN models. This includes variational approximation approaches, like BBB, which improved the calibration of uncertainty estimates obtained from models trained on real-world pharmaceutical data. In addition, sampling parts or the whole NN using HMC improved the UQ of QSAR models. We believe that increased focus on Bayesian approaches, for instance through the use of sampling-based methods, will improve our understanding of the posterior distributions of NNs and ultimately improve the probability calibration of models.

5.3 Future Research

In this final section, we explore aspects that remain unresolved in this work and that offer opportunities for future research. These new perspectives could provide deeper insight into UQ for drug discovery-related tasks and advance the development of calibrated uncertainty estimates for ML predictions.

UQ for Multi-task Models. In multi-task models (Caruana, 1997), related tasks are trained simultaneously, allowing the model to leverage information from one task to predict similar tasks. The multi-task approach also enables better predictions by increasing the size of the training data, leading to better generalization by learning from more diverse inputs. Furthermore, models can benefit from the multi-task setting when data is scarce.

Multi-task models have been applied in the drug discovery context (Simões et al., 2018; Dahl et al., 2014; Ramsundar et al., 2017), for example, to predict the bioactivity of a compound across multiple targets (Arany et al., 2022). Several studies examine UQ of multi-task ML models for molecular property prediction tasks (Stoyanova et al., 2023; Rodríguez-Pérez et al., 2022; Kim et al., 2021). However, a comprehensive study that compares the performance of ensemble-based and Bayesian UQ methods to QSAR models is, to our knowledge, still missing from the literature. Expanding our work to the multi-task setting would give important insights into whether the better generalization abilities of multi-task models also translate to better uncertainty estimates in QSAR modeling.

Including Target-related Information. While the data used in this thesis provides only chemical information about the compounds, information on the structure of the target protein could also be included. This is especially useful in multi-task models, which are able to simultaneously predict activities for multiple targets. The incorporation of target information allows the generation of meaningful predictions for targets with little (few-shot learning) or no (zero-shot learning) training data available (Schimunek et al., 2021; Svensson et al., 2024b).

Proteo-chemometric modeling (Lapinsh et al., 2001) provides an alternative to the traditional QSAR approach and enables the incorporation of target information. To do so, information about the target protein is encoded, for example, by using the amino acid sequences of the proteins of interest (Alley et al., 2019; Elnaggar et al., 2021). The protein embedding can be combined with the ligand embedding to subsequently model drug-target interactions (He et al., 2017). To our knowledge, the literature on UQ for proteo-chemometric modeling is scarce. Research in this direction would provide an extension to the previously described multi-task models and would offer the opportunity to extend UQ to few- and zero-shot learning in protein-target interaction predictions.

Hypernetworks for Posterior Estimation. Hypernetworks (Ha et al., 2017) are NN that output the parameters of a second NN. This method has recently been shown to improve generalization capabilities (Hospedales et al., 2021; Ji and Marttinen, 2023) and model calibration (Chauhan et al., 2024; Krueger et al., 2017) of NN. Hypernetworks have been applied to drug-target interaction prediction tasks (Nachmani and Wolf, 2020; Svensson et al., 2024b). Svensson et al. (2024b) show that hypernetworks for proteo-chemometric modeling match the performance of multiple benchmark methods and perform well on unseen protein targets.

Instead of predicting point estimates for NN, hypernetworks can be used to predict probability distributions over model parameters, resulting in a variational Bayesian setting (Krueger et al., 2017). Future extensions of this work include applying UQ to hypernetworks, a direction that is currently underexplored in the literature. The input of the hypernetwork can be either random noise, as modelled by Krueger et al. (2017), or additional information about the target, as applied by the HyperPCM approach by (Svensson et al., 2024b).

Future Directions for UQ with HMC. The preliminary work in Section 4.3 only provides a glimpse into generating efficient HMC samplers. Much more effort can be directed toward understanding how model architecture, model size, and the HP settings ot the samplers influence the behavior of HMC chains, with the goal of achieving well-mixing and convergent chains. An extension of this research may include the following topics:

- **Implementation of HMC in Pyro.** Pyro (Bingham et al., 2019) is a popular probabilistic programming language that allows to flexibly apply diverse methods for probabilistic modeling, including variational methods or MCMC sampling. Concerning HMC, it provides more sophisticated tools, such as an implementation of the No-U-Turn sampler (NUTS) (see next bullet point), that facilitate an efficient sampling process.
- **Sampling with NUTS.** Application of the No-U-Turn sampler (NUTS) (Hoffman et al., 2014) enables automatic adaptation of the step size to achieve an acceptance rate of around 0.8. Furthermore, the trajectory length, hence the number of integration steps L used before drawing a sample, is automatically tuned (Betancourt, 2017). This approach prevents the use of overly small values of L, which can lead to poorly mixing chains. Furthermore, it avoids inefficient chains that loop back to the starting point by stopping the trajectory as soon as the chain makes a U-turn. Thus, sampling with NUTS can achieve lower autocorrelation and better mixing behavior of the chains, which ultimately benefits UQ with HMC.
- **Adaptation of Mass Matrix.** The mass matrix can be used to decorrelate the posterior space by making dimension-specific adjustments to the momentum. Intuitively, the sampler will move faster in dimensions where the posterior is flat. Thus, tuning of the mass matrix can be helpful, especially in situations described in Section 4.3, when the sampler is not converging in some dimensions. The mass matrix can be either fixed at the beginning of the simulation or updated after every sampling step, which is referred to as Riemannian HMC (Girolami and Calderhead, 2011).

Expanding the Application of HBLL to larger NN. In Section 4.2, we show that the HBLL approach matches or outperforms other UQ approaches on three biological targets. At the same time, we acknowledge that the small size of the considered models is a limitation of our work since it is uncertain if our findings are also valid for more sophisticated chemical descriptors and larger NNs. HBLL generates partial Bayesian NN, by generating HMC samples only for the last layer of the model. Thus, HBLL is computationally efficient, and

allows HMC sampling to be applied easily to larger NN models. Furthermore, other computationally efficient HMC approaches for larger models exist (Cobb and Jalaian, 2021), which could be adapted to QSAR modeling and compared against the performance of HBLLs.

Customizing Proper Losses Through Differentiable Calibration Errors. As described in Section 2.2.2, proper scores can be used to assess model performance by taking into account both calibration and accuracy. Proper scores can be decomposed into a calibration and a refinement term, both of which contribute to the final value. Popordanoska et al. (2024) introduced a differentiable estimator of the calibration error and refinement, which add up to a proper loss. Adjusting the weights of the terms in the proper loss would allow experts to emphasize either calibration or accuracy, enabling task-specific adaptation of different aspects of model performance.

ML for Drug Discovery beyond Small Molecules. Traditionally, drug discovery has focused on developing small-molecule agents that modulate the activity of a biological target. Recently, other therapeutic approaches have been gaining importance, owing in part to the fact that many previously untargeted proteins are difficult to modulate with classical approaches (Yang et al., 2020). These emerging approaches include new drug modalities, such as antibodies, peptides, oligo- and polynucleotides, and drug conjugates, as well as new modes of action, including protein degradation, genome editing, and modulation of protein expression using RNA (Yang et al., 2020; Valeur and Jimonet, 2018). Due to the diversity of these new strategies, a comprehensive discussion is beyond the scope of this work.

Similar to small molecule development, these newer therapeutic modalities can also benefit substantially from the ML toolbox. For example, ML has been used to support the development of drug conjugates (Hsueh et al., 2023; Angiolini et al., 2025), and RNA-based drugs (Sato and Hamada, 2023), as well as the design of antibodies (Graves et al., 2020). In addition, for many of these applications, estimating predictive uncertainty can be highly valuable. For example, (Geylan et al., 2024) estimate the uncertainty in ML models used to predict the cell permeability of peptides. Thus, these state-of-the-art advancements offer exciting opportunities to apply UQ to a wide range of drug discovery challenges, accelerating progress in the field.

Appendix A

Appendix: Chapter 3

A.1 UQ under Temporal Distribution Shift

This supplementary material outlines the HP space explored during model selection and reports the proportion of compounds belonging to the preferred class across different assays and temporal folds. It also includes further insights into the distribution shift in both label and descriptor space. Additionally, we present an additional experiment analyzing the extent of scaffold overlap between training and test sets. The final section provides the complete numerical results of the calibration study.

A.1.1 Model Selection

Table A.1: **Summary of the HP optimization strategy.** The HPs of the baseline models were tuned for each temporal setting of each dataset individually using the BCE loss on a validation dataset. The range of the HPs considered during the HP tuning is reported in the left column. The right column lists the fixed HPs for each baseline model.

Baseline	Tuned Hyperparameter	Explored Values	Fixed Hyperparameter	Values
Random Forest (RF)	Number of Estimators	50 - 1500	ECFP Size	4096
	Maximum Depth	5 - 10000		
Mulitlayer Perceptron (MLP)	Weight Decay	0 - 0.0005		
	Dropout	0 - 0.75		
	Hidden Dimension	64 - 512	ECFP Size	4096
	Number of Layers	2 - 5	Learning Rate	0.0001
	Decreasing Dimension	True/False		
	Scheduler Factor	0.1/0.5		

A.1.2 Distribution Shift in Label Space

Table A.2: **Overview of compound ratios of the PC.** The ratios of the compounds assigned to the PC are reported for each temporal fold and assay.

Assay	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
TB-1	0.678218	0.531628	0.422665	0.236022	0.355208
TB-2	0.336185	0.401515	0.389691	0.304681	0.537486
TB-3	0.780019	0.687738	0.789397	0.878903	0.955903
TB-4	0.547912	0.869677	0.779412	0.715577	0.760392
TB-5	0.102444	0.245645	0.119409	0.176832	0.168000
TB-6	0.750672	0.657795	0.747870	0.870065	0.905208
TB-7	0.299052	0.717688	0.472853	0.189217	0.079219
A-CYPa	0.853501	0.675146	0.775178	0.817248	0.773893
A-CYPb	0.856396	0.844581	0.793540	0.823023	0.864290
A-CYPc	0.606965	0.761506	0.728951	0.635174	0.707853
A-PM	0.186431	0.195426	0.155797	0.096440	0.090965
A-SOL	0.470625	0.472610	0.488615	0.525761	0.434304
A-hERG	0.559480	0.694663	0.726091	0.719838	0.783182
A-logD	0.633528	0.600615	0.608635	0.554792	0.533047
A-MS	0.340375	0.333128	0.356558	0.421075	0.365665

A.1.3 Distribution Shift in Descriptor Space

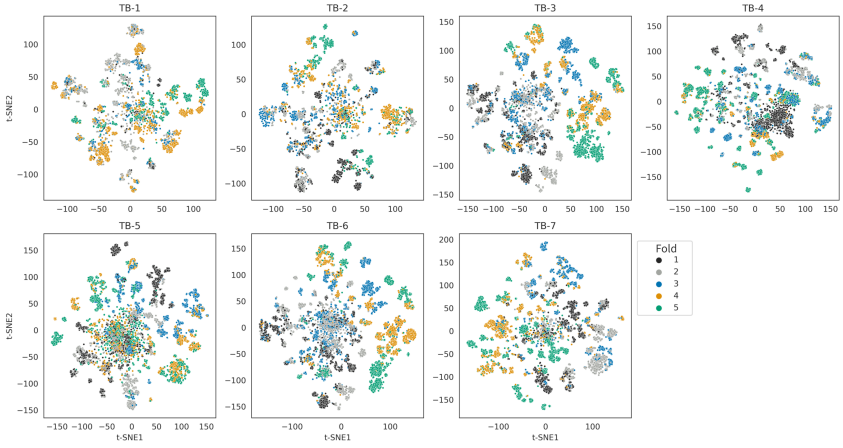

Figure A.1: **T-SNE plots of the ECFP space for all TB assays.** T-SNE plots of the ECFP space are shown for all TB assays. Compounds are colored according to the temporal fold that they were assigned to.

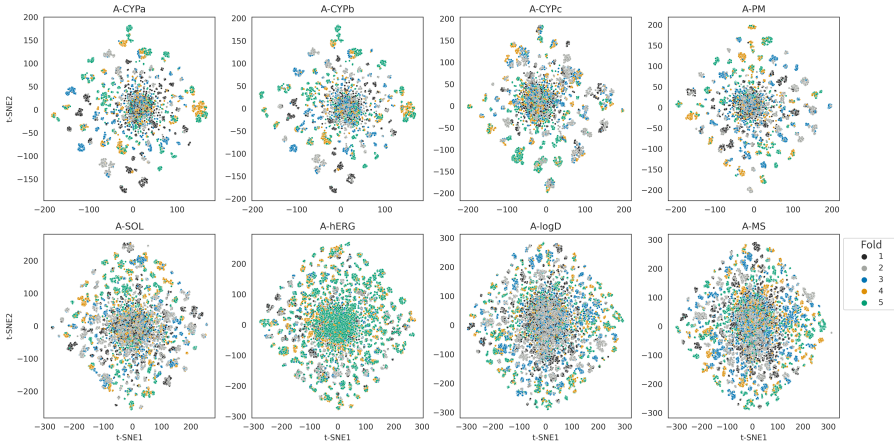

Figure A.2: **T-SNE plots of the ECFP space for all ADME-T assays.** T-SNE plots of the ECFP space are shown for all ADME-T assays. Compounds are colored according to the temporal fold that they were assigned to.

Table A.3: **Ratio of scaffold overlap between train and test set.** The ratios of scaffolds in the test set that are shared with the corresponding training set are listed across every assay and temporal setting. The mean of all temporal settings is shown in the last column.

	Temporal Setting			
Assay	1	2	3	Mean
TB				
TB-1	0.026	0.074	0.057	0.052
TB-2	0.085	0.094	0.076	0.085
TB-3	0.048	0.020	0.014	0.027
TB-4	0.037	0.015	0.039	0.030
TB-5	0.039	0.048	0.051	0.046
TB-6	0.043	0.019	0.018	0.027
TB-7	0.064	0.044	0.033	0.047
ADME-T				
A-CYPa	0.021	0.016	0.022	0.020
A-CYPb	0.019	0.018	0.024	0.020
A-CYPc	0.032	0.062	0.051	0.048
A-PM	0.014	0.016	0.014	0.015
A-SOL	0.026	0.041	0.040	0.036
A-hERG	0.038	0.049	0.053	0.047
A-logD	0.023	0.028	0.036	0.029
A-MS	0.019	0.027	0.036	0.027

Table A.4: **Quantification of the distribution shifts in descriptor space between the calibration and test set.** The shift in descriptor space is reported for each temporal setting, using the data of 1, 2, or 3 temporal folds for training. Results are shown for each assay. The shift in descriptor space is shown in terms of the MMD between the calibration and test datasets.

	Temporal Setting		
Assay	1	2	3
TB			
TB-1	0.022208	0.016968	0.057858
TB-2	0.018045	0.061332	0.050021
TB-3	0.063842	0.098653	0.073235
TB-4	0.033499	0.043690	0.020273
TB-5	0.088506	0.054487	0.035291
TB-6	0.044372	0.107566	0.062775
TB-7	0.130317	0.095124	0.048923
ADME-T			
A-CYPa	0.024293	0.028514	0.016945
A-CYPb	0.018977	0.026954	0.013106
A-CYPc	0.010841	0.010273	0.016587
A-PM	0.015786	0.016645	0.015720
A-SOL	0.006665	0.008250	0.007077
A-hERG	0.004232	0.003890	0.006738
A-logD	0.008009	0.005819	0.007059
A-MS	0.008084	0.006091	0.007487

A.1.4 Results of the Calibration Study

Table A.5: **Summary of the BCE (↓) scores for the baseline models**. BCE results for all three temporal settings of all TB and ADME-T assays are reported. The temporal setting is indicated in brackets after the assay abbreviation. Results for the baselines (RF and a MLP), Platt-scaled (RF+PS and RF+PS) and models calibrated with VA predictors (RF+VA and RF+VA) are reported. The models were trained with compounds from three temporal folds. Averages over 10 model repetitions are shown. The table is continued on the next page.

Assay	RF	RF+PS	RF+VA	MLP	RF+PS	RF+VA
TB-1[1]	0.82 ± 0.012	0.7 ± 0.006	0.7 ± 0.006	0.69 ± 0.017	0.64 ± 0.055	0.6 ± 0.064
TB-1[2]	0.66 ± 0.005	0.51 ± 0.006	0.51 ± 0.012	0.44 ± 0.012	0.43 ± 0.004	0.44 ± 0.005
TB-1[3]	0.68 ± 0.003	0.81 ± 0.036	0.77 ± 0.011	0.73 ± 0.066	0.77 ± 0.03	0.77 ± 0.036
TB-2[1]	0.63 ± 0.007	0.61 ± 0.009	0.63 ± 0.015	0.67 ± 0.038	0.67 ± 0.027	0.64 ± 0.006
TB-2[2]	0.58 ± 0.003	0.56 ± 0.005	0.56 ± 0.006	0.57 ± 0.062	0.54 ± 0.011	0.54 ± 0.009
TB-2[3]	0.75 ± 0.009	0.8 ± 0.025	0.78 ± 0.024	0.62 ± 0.024	0.66 ± 0.011	0.7 ± 0.013
TB-3[1]	0.52 ± 0.003	0.64 ± 0.025	0.68 ± 0.016	0.49 ± 0.009	0.56 ± 0.005	0.56 ± 0.008
TB-3[2]	0.46 ± 0.011	0.38 ± 0.011	0.37 ± 0.009	0.45 ± 0.016	0.44 ± 0.006	0.52 ± 0.093
TB-3[3]	0.33 ± 0.01	0.2 ± 0.009	0.2 ± 0.013	0.19 ± 0.006	0.21 ± 0.006	0.21 ± 0.004
TB-4[1]	0.57 ± 0.013	0.39 ± 0.043	0.4 ± 0.067	0.31 ± 0.018	0.31 ± 0.016	0.31 ± 0.007
TB-4[2]	0.55 ± 0.005	0.45 ± 0.02	0.47 ± 0.011	0.46 ± 0.023	0.46 ± 0.029	0.47 ± 0.021
TB-4[3]	0.5 ± 0.012	0.33 ± 0.031	0.32 ± 0.037	0.27 ± 0.082	0.24 ± 0.028	0.24 ± 0.038
TB-5[1]	0.34 ± 0.003	0.41 ± 0.005	0.41 ± 0.021	0.53 ± 0.041	0.39 ± 0.005	0.37 ± 0.003
TB-5[2]	0.46 ± 0.007	0.48 ± 0.018	0.49 ± 0.018	0.63 ± 0.076	0.53 ± 0.018	0.52 ± 0.024
TB-5[3]	0.44 ± 0.004	0.42 ± 0.008	0.41 ± 0.011	0.31 ± 0.007	0.28 ± 0.008	0.29 ± 0.006
TB-6[1]	0.55 ± 0.002	0.64 ± 0.026	0.67 ± 0.029	0.54 ± 0.004	0.65 ± 0.006	0.66 ± 0.007
TB-6[2]	0.51 ± 0.011	0.43 ± 0.011	0.43 ± 0.02	0.46 ± 0.036	0.44 ± 0.017	0.5 ± 0.062
TB-6[3]	0.43 ± 0.007	0.3 ± 0.012	0.3 ± 0.012	0.29 ± 0.007	0.28 ± 0.004	0.29 ± 0.003
TB-7[1]	0.79 ± 0.009	0.73 ± 0.024	0.69 ± 0.032	0.78 ± 0.107	0.72 ± 0.133	0.73 ± 0.125
TB-7[2]	0.57 ± 0.007	0.54 ± 0.019	0.46 ± 0.036	0.48 ± 0.041	0.47 ± 0.028	0.44 ± 0.016
TB-7[3]	0.53 ± 0.013	0.32 ± 0.013	0.34 ± 0.031	0.35 ± 0.113	0.28 ± 0.009	0.28 ± 0.011
A-CYPa[1]	0.54 ± 0.003	0.53 ± 0.005	0.52 ± 0.01	0.55 ± 0.01	0.55 ± 0.007	0.57 ± 0.023
A-CYPa[2]	0.47 ± 0.002	0.48 ± 0.003	0.5 ± 0.014	0.48 ± 0.004	0.48 ± 0.002	0.5 ± 0.005
A-CYPa[3]	0.51 ± 0.003	0.51 ± 0.005	0.51 ± 0.005	0.53 ± 0.009	0.53 ± 0.004	0.52 ± 0.005
A-CYPb[1]	0.51 ± 0.003	0.5 ± 0.007	0.49 ± 0.007	0.49 ± 0.008	0.49 ± 0.003	0.49 ± 0.002
A-CYPb[2]	0.47 ± 0.001	0.53 ± 0.01	0.56 ± 0.008	0.49 ± 0.015	0.51 ± 0.012	0.52 ± 0.02
A-CYPb[3]	0.4 ± 0.001	0.4 ± 0.001	0.4 ± 0.002	0.39 ± 0.008	0.39 ± 0.001	0.39 ± 0.002
A-CYPc[1]	0.6 ± 0.002	0.56 ± 0.005	0.57 ± 0.014	0.59 ± 0.009	0.58 ± 0.003	0.58 ± 0.002
A-CYPc[2]	0.63 ± 0.005	0.62 ± 0.012	0.6 ± 0.017	0.63 ± 0.027	0.6 ± 0.016	0.6 ± 0.019
A-CYPc[3]	0.59 ± 0.003	0.62 ± 0.017	0.62 ± 0.016	0.55 ± 0.004	0.56 ± 0.002	0.57 ± 0.002
A-PM[1]	0.44 ± 0.001	0.43 ± 0.001	0.44 ± 0.004	0.38 ± 0.002	0.38 ± 0.002	0.38 ± 0.002
A-PM[2]	0.35 ± 0.002	0.33 ± 0.005	0.34 ± 0.006	0.28 ± 0.005	0.3 ± 0.006	0.3 ± 0.023
A-PM[3]	0.34 ± 0.001	0.3 ± 0.003	0.3 ± 0.005	0.28 ± 0.006	0.27 ± 0.002	0.27 ± 0.002
A-SOL[1]	0.68 ± 0.001	0.66 ± 0.006	0.66 ± 0.007	0.6 ± 0.002	0.59 ± 0.004	0.59 ± 0.004
A-SOL[2]	0.68 ± 0.002	0.63 ± 0.007	0.63 ± 0.007	0.61 ± 0.03	0.6 ± 0.031	0.57 ± 0.02
A-SOL[3]	0.67 ± 0.0	0.64 ± 0.006	0.64 ± 0.006	0.59 ± 0.027	0.58 ± 0.026	0.56 ± 0.013
A-hERG[1]	0.61 ± 0.002	0.54 ± 0.003	0.54 ± 0.003	0.54 ± 0.024	0.52 ± 0.005	0.51 ± 0.004
A-hERG[2]	0.59 ± 0.001	0.54 ± 0.004	0.54 ± 0.004	0.5 ± 0.002	0.5 ± 0.003	0.5 ± 0.001
A-hERG[3]	0.54 ± 0.001	0.5 ± 0.003	0.51 ± 0.004	0.49 ± 0.011	0.48 ± 0.005	0.48 ± 0.004
A-logD[1]	0.64 ± 0.004	0.62 ± 0.01	0.62 ± 0.01	0.59 ± 0.008	0.57 ± 0.004	0.57 ± 0.003
A-logD[2]	0.67 ± 0.001	0.64 ± 0.003	0.64 ± 0.002	0.64 ± 0.031	0.62 ± 0.022	0.59 ± 0.016
A-logD[3]	0.69 ± 0.001	0.66 ± 0.007	0.65 ± 0.008	0.53 ± 0.007	0.5 ± 0.004	0.5 ± 0.005
A-MS[1]	0.64 ± 0.001	0.63 ± 0.004	0.63 ± 0.004	0.61 ± 0.018	0.6 ± 0.02	0.6 ± 0.028
A-MS[2]	0.69 ± 0.002	0.66 ± 0.007	0.66 ± 0.006	0.68 ± 0.011	0.65 ± 0.015	0.65 ± 0.008
A-MS[3]	0.64 ± 0.001	0.61 ± 0.004	0.61 ± 0.003	0.64 ± 0.014	0.64 ± 0.006	0.6 ± 0.005

Table A.6: **Continuation of Table A.5**. Results for the uncalibrated uncertainty quantification models (Ensemble, MC–Dropout, and BBB), their PS counterparts (Ensemble+PS, MC–Dropout+PS and BBB+PS) and the models calibrated with VA predictors (Ensemble+VA, MC–Dropout+VA and BBB+VA) are reported.

Assay	Ensemble	Ensemble+PS	Ensemble+VA	MC–Dropout	MC–Dropout+PS	MC–Dropout+VA	BBB	BBB+PS	BBB+VA
TB-1[1]	0.7 ± 0.008	0.63 ± 0.021	0.56 ± 0.009	0.69 ± 0.014	0.65 ± 0.05	0.61 ± 0.065	0.7 ± 0.003	0.71 ± 0.001	0.78 ± 0.137
TB-1[2]	0.44 ± 0.003	0.43 ± 0.001	0.44 ± 0.001	0.44 ± 0.013	0.43 ± 0.004	0.44 ± 0.004	0.74 ± 0.021	0.62 ± 0.004	0.62 ± 0.037
TB-1[3]	0.67 ± 0.01	0.8 ± 0.004	0.77 ± 0.007	0.66 ± 0.023	0.79 ± 0.032	0.77 ± 0.038	0.69 ± 0.002	0.69 ± 0.001	0.75 ± 0.064
TB-2[1]	0.64 ± 0.009	0.67 ± 0.006	0.64 ± 0.003	0.63 ± 0.011	0.65 ± 0.015	0.64 ± 0.007	0.69 ± 0.001	0.67 ± 0.0	0.72 ± 0.047
TB-2[2]	0.54 ± 0.003	0.53 ± 0.004	0.53 ± 0.003	0.59 ± 0.053	0.53 ± 0.007	0.54 ± 0.009	0.68 ± 0.006	0.63 ± 0.002	0.7 ± 0.101
TB-2[3]	0.63 ± 0.005	0.66 ± 0.003	0.7 ± 0.005	0.64 ± 0.024	0.67 ± 0.011	0.7 ± 0.013	0.72 ± 0.02	0.81 ± 0.007	0.81 ± 0.007
TB-3[1]	0.5 ± 0.006	0.56 ± 0.001	0.55 ± 0.002	0.5 ± 0.014	0.55 ± 0.005	0.56 ± 0.008	0.53 ± 0.004	0.54 ± 0.004	0.54 ± 0.023
TB-3[2]	0.45 ± 0.004	0.45 ± 0.003	0.46 ± 0.002	0.46 ± 0.015	0.45 ± 0.006	0.46 ± 0.005	0.43 ± 0.006	0.38 ± 0.008	0.35 ± 0.005
TB-3[3]	0.19 ± 0.002	0.21 ± 0.001	0.21 ± 0.001	0.27 ± 0.012	0.2 ± 0.004	0.21 ± 0.003	0.28 ± 0.009	0.21 ± 0.011	0.21 ± 0.001
TB-4[1]	0.3 ± 0.005	0.3 ± 0.002	0.3 ± 0.002	0.37 ± 0.013	0.31 ± 0.012	0.31 ± 0.009	0.4 ± 0.01	0.37 ± 0.006	0.32 ± 0.003
TB-4[2]	0.45 ± 0.004	0.45 ± 0.005	0.46 ± 0.004	0.42 ± 0.018	0.44 ± 0.027	0.46 ± 0.018	0.43 ± 0.005	0.45 ± 0.006	0.52 ± 0.014
TB-4[3]	0.25 ± 0.01	0.22 ± 0.004	0.22 ± 0.001	0.35 ± 0.058	0.23 ± 0.017	0.23 ± 0.013	0.33 ± 0.008	0.28 ± 0.003	0.22 ± 0.003
TB-5[1]	0.52 ± 0.006	0.39 ± 0.001	0.37 ± 0.001	0.5 ± 0.036	0.39 ± 0.004	0.37 ± 0.003	0.46 ± 0.008	0.42 ± 0.002	0.43 ± 0.001
TB-5[2]	0.57 ± 0.03	0.53 ± 0.003	0.52 ± 0.005	0.46 ± 0.022	0.52 ± 0.02	0.52 ± 0.022	0.49 ± 0.017	0.47 ± 0.012	0.51 ± 0.02
TB-5[3]	0.3 ± 0.002	0.28 ± 0.001	0.29 ± 0.001	0.31 ± 0.007	0.28 ± 0.008	0.29 ± 0.006	0.38 ± 0.032	0.38 ± 0.024	0.35 ± 0.037
TB-6[1]	0.54 ± 0.003	0.65 ± 0.006	0.66 ± 0.006	0.56 ± 0.008	0.65 ± 0.008	0.66 ± 0.008	0.58 ± 0.007	0.58 ± 0.002	0.58 ± 0.029
TB-6[2]	0.45 ± 0.004	0.44 ± 0.004	0.45 ± 0.005	0.49 ± 0.025	0.44 ± 0.021	0.43 ± 0.027	0.45 ± 0.01	0.41 ± 0.014	0.39 ± 0.003
TB-6[3]	0.29 ± 0.001	0.28 ± 0.0	0.29 ± 0.001	0.3 ± 0.006	0.28 ± 0.004	0.29 ± 0.003	0.37 ± 0.01	0.3 ± 0.002	0.31 ± 0.003
TB-7[1]	0.66 ± 0.02	0.61 ± 0.026	0.58 ± 0.004	0.72 ± 0.032	0.72 ± 0.134	0.72 ± 0.125	0.69 ± 0.0	0.82 ± 0.001	0.83 ± 0.098
TB-7[2]	0.46 ± 0.007	0.46 ± 0.005	0.42 ± 0.002	0.52 ± 0.027	0.47 ± 0.016	0.44 ± 0.012	0.6 ± 0.029	0.55 ± 0.019	0.46 ± 0.008
TB-7[3]	0.3 ± 0.009	0.27 ± 0.002	0.27 ± 0.002	0.37 ± 0.048	0.27 ± 0.009	0.27 ± 0.008	0.69 ± 0.002	0.33 ± 0.002	0.36 ± 0.076
A-CYPa[1]	0.54 ± 0.003	0.55 ± 0.004	0.56 ± 0.008	0.56 ± 0.012	0.55 ± 0.01	0.56 ± 0.016	0.6 ± 0.01	0.56 ± 0.002	0.57 ± 0.025
A-CYPa[2]	0.48 ± 0.004	0.52 ± 0.023	0.58 ± 0.032	0.49 ± 0.01	0.48 ± 0.003	0.5 ± 0.004	0.47 ± 0.002	0.48 ± 0.002	0.5 ± 0.004
A-CYPa[3]	0.52 ± 0.004	0.53 ± 0.001	0.52 ± 0.001	0.52 ± 0.012	0.52 ± 0.004	0.53 ± 0.004	0.52 ± 0.004	0.53 ± 0.006	0.53 ± 0.006
A-CYPb[1]	0.49 ± 0.002	0.49 ± 0.0	0.49 ± 0.001	0.51 ± 0.008	0.5 ± 0.002	0.5 ± 0.003	0.51 ± 0.002	0.52 ± 0.001	0.51 ± 0.003
A-CYPb[2]	0.48 ± 0.003	0.51 ± 0.002	0.51 ± 0.003	0.48 ± 0.015	0.52 ± 0.02	0.52 ± 0.025	0.46 ± 0.0	0.47 ± 0.003	0.49 ± 0.002
A-CYPb[3]	0.38 ± 0.001	0.39 ± 0.0	0.39 ± 0.0	0.42 ± 0.018	0.39 ± 0.001	0.39 ± 0.002	0.39 ± 0.005	0.39 ± 0.002	0.38 ± 0.001
A-CYPc[1]	0.59 ± 0.001	0.58 ± 0.001	0.58 ± 0.001	0.58 ± 0.006	0.58 ± 0.003	0.58 ± 0.002	0.63 ± 0.002	0.59 ± 0.0	0.59 ± 0.002
A-CYPc[2]	0.6 ± 0.007	0.6 ± 0.004	0.58 ± 0.002	0.59 ± 0.01	0.62 ± 0.013	0.6 ± 0.017	0.62 ± 0.029	0.63 ± 0.032	0.62 ± 0.04
A-CYPc[3]	0.55 ± 0.001	0.56 ± 0.0	0.56 ± 0.001	0.55 ± 0.002	0.56 ± 0.002	0.56 ± 0.002	0.54 ± 0.004	0.55 ± 0.004	0.55 ± 0.003
A-PM[1]	0.38 ± 0.001	0.38 ± 0.0	0.38 ± 0.001	0.38 ± 0.003	0.37 ± 0.002	0.38 ± 0.002	0.42 ± 0.003	0.42 ± 0.002	0.37 ± 0.002
A-PM[2]	0.28 ± 0.001	0.29 ± 0.001	0.29 ± 0.001	0.36 ± 0.018	0.29 ± 0.006	0.3 ± 0.005	0.3 ± 0.007	0.31 ± 0.012	0.29 ± 0.001
A-PM[3]	0.27 ± 0.001	0.27 ± 0.0	0.27 ± 0.001	0.28 ± 0.006	0.27 ± 0.002	0.27 ± 0.002	0.29 ± 0.004	0.28 ± 0.003	0.26 ± 0.008
A-SOL[1]	0.6 ± 0.001	0.59 ± 0.001	0.59 ± 0.001	0.59 ± 0.002	0.59 ± 0.004	0.59 ± 0.004	0.62 ± 0.039	0.6 ± 0.012	0.59 ± 0.013
A-SOL[2]	0.56 ± 0.004	0.54 ± 0.005	0.54 ± 0.004	0.6 ± 0.013	0.59 ± 0.025	0.58 ± 0.023	0.69 ± 0.001	0.69 ± 0.001	0.72 ± 0.049
A-SOL[3]	0.55 ± 0.004	0.55 ± 0.003	0.55 ± 0.003	0.57 ± 0.022	0.58 ± 0.022	0.56 ± 0.015	0.69 ± 0.001	0.7 ± 0.003	0.72 ± 0.068
A-hERG[1]	0.52 ± 0.003	0.51 ± 0.001	0.51 ± 0.001	0.55 ± 0.022	0.52 ± 0.005	0.51 ± 0.005	0.63 ± 0.002	0.58 ± 0.0	0.57 ± 0.001
A-hERG[2]	0.5 ± 0.001	0.49 ± 0.001	0.5 ± 0.0	0.51 ± 0.004	0.49 ± 0.002	0.5 ± 0.001	0.56 ± 0.017	0.54 ± 0.017	0.53 ± 0.014
A-hERG[3]	0.47 ± 0.001	0.48 ± 0.001	0.47 ± 0.001	0.49 ± 0.01	0.48 ± 0.005	0.48 ± 0.004	0.51 ± 0.006	0.5 ± 0.004	0.5 ± 0.002
A-logD[1]	0.58 ± 0.001	0.56 ± 0.001	0.57 ± 0.001	0.56 ± 0.003	0.56 ± 0.003	0.57 ± 0.003	0.61 ± 0.006	0.6 ± 0.004	0.6 ± 0.004
A-logD[2]	0.6 ± 0.012	0.6 ± 0.009	0.57 ± 0.003	0.61 ± 0.022	0.6 ± 0.019	0.59 ± 0.009	0.68 ± 0.005	0.66 ± 0.008	0.66 ± 0.01
A-logD[3]	0.52 ± 0.001	0.49 ± 0.003	0.49 ± 0.002	0.51 ± 0.005	0.5 ± 0.004	0.5 ± 0.004	0.51 ± 0.006	0.52 ± 0.004	0.5 ± 0.005
A-MS[1]	0.6 ± 0.002	0.59 ± 0.001	0.59 ± 0.001	0.6 ± 0.018	0.6 ± 0.02	0.6 ± 0.028	0.61 ± 0.008	0.61 ± 0.004	0.6 ± 0.005
A-MS[2]	0.67 ± 0.003	0.64 ± 0.002	0.64 ± 0.003	0.67 ± 0.009	0.65 ± 0.015	0.65 ± 0.015	0.66 ± 0.013	0.64 ± 0.007	0.64 ± 0.006
A-MS[3]	0.62 ± 0.01	0.63 ± 0.008	0.61 ± 0.006	0.61 ± 0.01	0.61 ± 0.008	0.6 ± 0.005	0.58 ± 0.009	0.58 ± 0.007	0.58 ± 0.005

Table A.7: **Summary of the ACE (↓) scores for the baseline models**. ACE results for all three temporal settings of all TB and ADME-T assays are reported. The temporal setting is indicated in brackets after the assay abbreviation. Results for the baselines (Random Forests (RF) and a MLP), Platt-scaled models (RF+PS and RF+PS), and models calibrated with VA predictors (RF+VA and RF+VA) are reported. The models were trained with compounds from three temporal folds. Averages over 10 model repetitions are shown. The table is continued on the next page.

Assay	RF	RF+PS	RF+VA	MLP	RF+PS	RF+VA
TB-1[1]	0.26 ± 0.011	0.16 ± 0.027	0.14 ± 0.018	0.23 ± 0.055	0.18 ± 0.039	0.12 ± 0.019
TB-1[2]	0.27 ± 0.006	0.13 ± 0.011	0.13 ± 0.009	0.09 ± 0.016	0.08 ± 0.003	0.09 ± 0.008
TB-1[3]	0.14 ± 0.018	0.21 ± 0.023	0.21 ± 0.006	0.16 ± 0.034	0.18 ± 0.028	0.19 ± 0.02
TB-2[1]	0.09 ± 0.01	0.06 ± 0.014	0.09 ± 0.018	0.1 ± 0.016	0.11 ± 0.02	0.09 ± 0.008
TB-2[2]	0.11 ± 0.011	0.06 ± 0.008	0.07 ± 0.019	0.14 ± 0.067	0.1 ± 0.012	0.09 ± 0.014
TB-2[3]	0.23 ± 0.007	0.28 ± 0.011	0.27 ± 0.012	0.16 ± 0.025	0.2 ± 0.004	0.23 ± 0.006
TB-3[1]	0.08 ± 0.014	0.23 ± 0.015	0.26 ± 0.01	0.08 ± 0.016	0.18 ± 0.005	0.17 ± 0.006
TB-3[2]	0.2 ± 0.016	0.12 ± 0.014	0.11 ± 0.007	0.16 ± 0.018	0.15 ± 0.005	0.2 ± 0.061
TB-3[3]	0.21 ± 0.009	0.07 ± 0.01	0.07 ± 0.01	0.06 ± 0.008	0.08 ± 0.007	0.09 ± 0.003
TB-4[1]	0.24 ± 0.033	0.08 ± 0.018	0.08 ± 0.024	0.05 ± 0.016	0.05 ± 0.018	0.04 ± 0.007
TB-4[2]	0.21 ± 0.026	0.09 ± 0.014	0.11 ± 0.015	0.11 ± 0.008	0.1 ± 0.012	0.1 ± 0.009
TB-4[3]	0.25 ± 0.018	0.09 ± 0.015	0.06 ± 0.017	0.09 ± 0.066	0.07 ± 0.02	0.07 ± 0.032
TB-5[1]	0.06 ± 0.009	0.16 ± 0.018	0.16 ± 0.025	0.09 ± 0.001	0.07 ± 0.001	0.07 ± 0.001
TB-5[2]	0.07 ± 0.006	0.07 ± 0.013	0.08 ± 0.012	0.15 ± 0.012	0.12 ± 0.005	0.12 ± 0.005
TB-5[3]	0.07 ± 0.01	0.06 ± 0.012	0.05 ± 0.009	0.05 ± 0.005	0.02 ± 0.003	0.03 ± 0.003
TB-6[1]	0.08 ± 0.013	0.21 ± 0.019	0.23 ± 0.024	0.09 ± 0.011	0.23 ± 0.007	0.23 ± 0.006
TB-6[2]	0.23 ± 0.008	0.15 ± 0.008	0.15 ± 0.019	0.17 ± 0.035	0.15 ± 0.013	0.19 ± 0.047
TB-6[3]	0.2 ± 0.005	0.04 ± 0.01	0.04 ± 0.009	0.06 ± 0.013	0.04 ± 0.004	0.05 ± 0.004
TB-7[1]	0.22 ± 0.006	0.16 ± 0.029	0.11 ± 0.035	0.16 ± 0.074	0.16 ± 0.107	0.17 ± 0.098
TB-7[2]	0.21 ± 0.014	0.18 ± 0.016	0.09 ± 0.03	0.13 ± 0.045	0.12 ± 0.027	0.08 ± 0.02
TB-7[3]	0.31 ± 0.009	0.11 ± 0.013	0.12 ± 0.026	0.11 ± 0.078	0.07 ± 0.012	0.07 ± 0.013
A-CYPa[1]	0.09 ± 0.003	0.11 ± 0.011	0.09 ± 0.01	0.09 ± 0.014	0.1 ± 0.005	0.11 ± 0.027
A-CYPa[2]	0.04 ± 0.013	0.06 ± 0.017	0.1 ± 0.029	0.06 ± 0.01	0.08 ± 0.004	0.08 ± 0.008
A-CYPa[3]	0.08 ± 0.004	0.07 ± 0.009	0.06 ± 0.006	0.06 ± 0.012	0.07 ± 0.004	0.06 ± 0.005
A-CYPb[1]	0.08 ± 0.005	0.07 ± 0.004	0.06 ± 0.006	0.06 ± 0.012	0.07 ± 0.003	0.07 ± 0.004
A-CYPb[2]	0.04 ± 0.007	0.12 ± 0.011	0.14 ± 0.009	0.08 ± 0.016	0.11 ± 0.008	0.11 ± 0.01
A-CYPb[3]	0.05 ± 0.005	0.05 ± 0.006	0.05 ± 0.006	0.05 ± 0.017	0.05 ± 0.003	0.05 ± 0.004
A-CYPc[1]	0.12 ± 0.003	0.06 ± 0.013	0.07 ± 0.015	0.08 ± 0.003	0.07 ± 0.003	0.07 ± 0.001
A-CYPc[2]	0.11 ± 0.012	0.1 ± 0.008	0.08 ± 0.007	0.12 ± 0.021	0.1 ± 0.005	0.09 ± 0.006
A-CYPc[3]	0.07 ± 0.013	0.13 ± 0.011	0.13 ± 0.012	0.05 ± 0.012	0.09 ± 0.002	0.09 ± 0.002
A-PM[1]	0.05 ± 0.003	0.04 ± 0.004	0.05 ± 0.003	0.02 ± 0.005	0.02 ± 0.001	0.02 ± 0.003
A-PM[2]	0.1 ± 0.003	0.06 ± 0.005	0.07 ± 0.006	0.04 ± 0.007	0.07 ± 0.007	0.08 ± 0.024
A-PM[3]	0.09 ± 0.001	0.02 ± 0.005	0.03 ± 0.008	0.04 ± 0.01	0.03 ± 0.002	0.03 ± 0.002
A-SOL[1]	0.1 ± 0.009	0.05 ± 0.009	0.03 ± 0.005	0.06 ± 0.004	0.03 ± 0.003	0.02 ± 0.003
A-SOL[2]	0.13 ± 0.006	0.04 ± 0.007	0.04 ± 0.007	0.09 ± 0.021	0.09 ± 0.027	0.04 ± 0.007
A-SOL[3]	0.13 ± 0.008	0.07 ± 0.005	0.07 ± 0.006	0.08 ± 0.019	0.06 ± 0.023	0.04 ± 0.005
A-hERG[1]	0.17 ± 0.005	0.04 ± 0.004	0.05 ± 0.006	0.09 ± 0.031	0.03 ± 0.007	0.02 ± 0.003
A-hERG[2]	0.11 ± 0.004	0.04 ± 0.008	0.04 ± 0.01	0.05 ± 0.004	0.03 ± 0.009	0.02 ± 0.003
A-hERG[3]	0.1 ± 0.001	0.03 ± 0.003	0.05 ± 0.006	0.07 ± 0.01	0.06 ± 0.007	0.05 ± 0.007
A-logD[1]	0.08 ± 0.013	0.03 ± 0.008	0.03 ± 0.008	0.08 ± 0.008	0.05 ± 0.002	0.04 ± 0.002
A-logD[2]	0.09 ± 0.007	0.05 ± 0.006	0.05 ± 0.007	0.1 ± 0.021	0.08 ± 0.02	0.06 ± 0.016
A-logD[3]	0.12 ± 0.012	0.08 ± 0.01	0.05 ± 0.005	0.07 ± 0.007	0.03 ± 0.003	0.03 ± 0.003
A-MS[1]	0.07 ± 0.011	0.03 ± 0.006	0.03 ± 0.009	0.08 ± 0.022	0.04 ± 0.006	0.05 ± 0.02
A-MS[2]	0.11 ± 0.007	0.05 ± 0.009	0.04 ± 0.004	0.12 ± 0.017	0.06 ± 0.017	0.06 ± 0.007
A-MS[3]	0.11 ± 0.006	0.06 ± 0.006	0.04 ± 0.003	0.08 ± 0.01	0.08 ± 0.009	0.05 ± 0.005

Table A.8: **Continuation of Table A.7**. Results for the uncalibrated uncertainty quantification models (Ensemble, MC–Dropout, and BBB, their PS counterparts (Ensemble+PS, MC–Dropout+PS and BBB+PS) and the models calibrated with VA predictors (Ensemble+VA, MC–Dropout+VA and BBB+VA) are reported.

Assay	Ensemble	Ensemble+PS	Ensemble+VA	MC–Dropout	MC–Dropout+PS	MC–Dropout+VA	BBB	BBB+PS	BBB+VA
TB-1[1]	0.28 ± 0.006	0.21 ± 0.021	0.13 ± 0.015	0.22 ± 0.055	0.17 ± 0.039	0.13 ± 0.022	0.09 ± 0.007	0.11 ± 0.004	0.18 ± 0.105
TB-1[2]	0.1 ± 0.004	0.08 ± 0.001	0.09 ± 0.005	0.1 ± 0.019	0.08 ± 0.003	0.09 ± 0.008	0.31 ± 0.021	0.21 ± 0.014	0.2 ± 0.031
TB-1[3]	0.13 ± 0.013	0.21 ± 0.002	0.18 ± 0.004	0.1 ± 0.022	0.2 ± 0.019	0.19 ± 0.018	0.15 ± 0.003	0.12 ± 0.002	0.21 ± 0.034
TB-2[1]	0.09 ± 0.008	0.1 ± 0.004	0.09 ± 0.004	0.08 ± 0.012	0.1 ± 0.01	0.08 ± 0.01	0.11 ± 0.003	0.04 ± 0.011	0.17 ± 0.05
TB-2[2]	0.11 ± 0.006	0.09 ± 0.006	0.08 ± 0.006	0.18 ± 0.051	0.09 ± 0.007	0.08 ± 0.01	0.18 ± 0.009	0.09 ± 0.004	0.2 ± 0.101
TB-2[3]	0.17 ± 0.005	0.2 ± 0.001	0.23 ± 0.002	0.18 ± 0.023	0.21 ± 0.004	0.23 ± 0.005	0.2 ± 0.022	0.28 ± 0.007	0.24 ± 0.007
TB-3[1]	0.09 ± 0.009	0.18 ± 0.004	0.17 ± 0.002	0.1 ± 0.022	0.17 ± 0.007	0.17 ± 0.007	0.08 ± 0.012	0.1 ± 0.007	0.09 ± 0.04
TB-3[2]	0.15 ± 0.005	0.15 ± 0.002	0.16 ± 0.002	0.18 ± 0.016	0.15 ± 0.006	0.16 ± 0.008	0.19 ± 0.006	0.14 ± 0.013	0.1 ± 0.007
TB-3[3]	0.06 ± 0.002	0.08 ± 0.001	0.09 ± 0.001	0.16 ± 0.011	0.07 ± 0.004	0.08 ± 0.003	0.16 ± 0.009	0.09 ± 0.012	0.09 ± 0.002
TB-4[1]	0.04 ± 0.007	0.03 ± 0.003	0.04 ± 0.003	0.11 ± 0.012	0.03 ± 0.012	0.04 ± 0.006	0.16 ± 0.009	0.12 ± 0.005	0.07 ± 0.003
TB-4[2]	0.1 ± 0.003	0.1 ± 0.002	0.1 ± 0.003	0.07 ± 0.006	0.09 ± 0.015	0.11 ± 0.011	0.06 ± 0.005	0.09 ± 0.012	0.14 ± 0.006
TB-4[3]	0.08 ± 0.012	0.05 ± 0.004	0.05 ± 0.003	0.18 ± 0.044	0.06 ± 0.009	0.05 ± 0.005	0.15 ± 0.007	0.1 ± 0.005	0.05 ± 0.003
TB-5[1]	0.09 ± 0.0	0.07 ± 0.0	0.07 ± 0.0	0.09 ± 0.002	0.07 ± 0.001	0.07 ± 0.002	0.18 ± 0.009	0.13 ± 0.004	0.13 ± 0.004
TB-5[2]	0.14 ± 0.009	0.12 ± 0.001	0.12 ± 0.002	0.08 ± 0.024	0.12 ± 0.006	0.12 ± 0.005	0.1 ± 0.016	0.08 ± 0.007	0.1 ± 0.014
TB-5[3]	0.05 ± 0.002	0.02 ± 0.001	0.03 ± 0.002	0.05 ± 0.005	0.02 ± 0.002	0.03 ± 0.003	0.08 ± 0.015	0.09 ± 0.011	0.03 ± 0.016
TB-6[1]	0.1 ± 0.005	0.23 ± 0.004	0.24 ± 0.006	0.12 ± 0.015	0.22 ± 0.01	0.23 ± 0.009	0.09 ± 0.018	0.09 ± 0.006	0.08 ± 0.053
TB-6[2]	0.16 ± 0.004	0.14 ± 0.003	0.15 ± 0.004	0.2 ± 0.023	0.14 ± 0.017	0.14 ± 0.024	0.2 ± 0.01	0.16 ± 0.017	0.11 ± 0.002
TB-6[3]	0.05 ± 0.001	0.04 ± 0.001	0.05 ± 0.001	0.07 ± 0.012	0.04 ± 0.004	0.05 ± 0.003	0.15 ± 0.011	0.04 ± 0.005	0.06 ± 0.006
TB-7[1]	0.19 ± 0.023	0.12 ± 0.03	0.05 ± 0.015	0.14 ± 0.08	0.16 ± 0.105	0.17 ± 0.09	0.03 ± 0.003	0.24 ± 0.001	0.25 ± 0.063
TB-7[2]	0.12 ± 0.008	0.12 ± 0.006	0.06 ± 0.006	0.18 ± 0.023	0.12 ± 0.013	0.07 ± 0.017	0.25 ± 0.024	0.21 ± 0.015	0.08 ± 0.011
TB-7[3]	0.11 ± 0.01	0.07 ± 0.002	0.07 ± 0.004	0.18 ± 0.041	0.06 ± 0.009	0.06 ± 0.011	0.42 ± 0.001	0.11 ± 0.003	0.14 ± 0.067
A-CYPa[1]	0.08 ± 0.003	0.1 ± 0.005	0.09 ± 0.01	0.11 ± 0.025	0.1 ± 0.009	0.09 ± 0.017	0.18 ± 0.013	0.11 ± 0.007	0.11 ± 0.023
A-CYPa[2]	0.08 ± 0.011	0.12 ± 0.019	0.14 ± 0.02	0.09 ± 0.022	0.08 ± 0.006	0.08 ± 0.006	0.04 ± 0.01	0.07 ± 0.006	0.11 ± 0.006
A-CYPa[3]	0.05 ± 0.001	0.07 ± 0.002	0.07 ± 0.003	0.07 ± 0.025	0.06 ± 0.002	0.06 ± 0.004	0.04 ± 0.008	0.06 ± 0.009	0.06 ± 0.002
A-CYPb[1]	0.06 ± 0.003	0.07 ± 0.001	0.07 ± 0.002	0.09 ± 0.019	0.06 ± 0.003	0.07 ± 0.003	0.05 ± 0.005	0.06 ± 0.004	0.05 ± 0.002
A-CYPb[2]	0.06 ± 0.003	0.11 ± 0.002	0.11 ± 0.003	0.08 ± 0.021	0.11 ± 0.01	0.12 ± 0.012	0.04 ± 0.004	0.06 ± 0.01	0.09 ± 0.002
A-CYPb[3]	0.04 ± 0.003	0.05 ± 0.001	0.05 ± 0.003	0.09 ± 0.032	0.05 ± 0.002	0.05 ± 0.004	0.05 ± 0.011	0.05 ± 0.006	0.04 ± 0.003
A-CYPc[1]	0.08 ± 0.001	0.07 ± 0.001	0.07 ± 0.001	0.06 ± 0.003	0.07 ± 0.003	0.07 ± 0.001	0.17 ± 0.003	0.1 ± 0.004	0.07 ± 0.007
A-CYPc[2]	0.12 ± 0.006	0.11 ± 0.003	0.09 ± 0.001	0.09 ± 0.016	0.11 ± 0.008	0.09 ± 0.005	0.07 ± 0.007	0.09 ± 0.005	0.08 ± 0.012
A-CYPc[3]	0.05 ± 0.002	0.08 ± 0.001	0.09 ± 0.002	0.05 ± 0.008	0.08 ± 0.002	0.09 ± 0.003	0.06 ± 0.01	0.08 ± 0.01	0.08 ± 0.004
A-PM[1]	0.02 ± 0.002	0.01 ± 0.001	0.02 ± 0.002	0.03 ± 0.006	0.01 ± 0.001	0.02 ± 0.002	0.13 ± 0.004	0.13 ± 0.004	0.06 ± 0.006
A-PM[2]	0.05 ± 0.003	0.07 ± 0.001	0.07 ± 0.001	0.13 ± 0.02	0.06 ± 0.007	0.07 ± 0.007	0.07 ± 0.01	0.08 ± 0.015	0.07 ± 0.001
A-PM[3]	0.04 ± 0.002	0.03 ± 0.001	0.03 ± 0.001	0.05 ± 0.009	0.03 ± 0.002	0.03 ± 0.002	0.07 ± 0.007	0.04 ± 0.005	0.02 ± 0.003
A-SOL[1]	0.05 ± 0.001	0.04 ± 0.001	0.02 ± 0.003	0.05 ± 0.004	0.04 ± 0.004	0.02 ± 0.004	0.09 ± 0.047	0.04 ± 0.008	0.02 ± 0.003
A-SOL[2]	0.08 ± 0.007	0.04 ± 0.009	0.03 ± 0.004	0.09 ± 0.016	0.07 ± 0.012	0.04 ± 0.01	0.04 ± 0.005	0.03 ± 0.007	0.08 ± 0.068
A-SOL[3]	0.04 ± 0.006	0.05 ± 0.003	0.03 ± 0.002	0.06 ± 0.018	0.07 ± 0.017	0.04 ± 0.004	0.05 ± 0.003	0.1 ± 0.008	0.11 ± 0.066
A-hERG[1]	0.06 ± 0.009	0.03 ± 0.001	0.02 ± 0.002	0.11 ± 0.031	0.04 ± 0.006	0.02 ± 0.003	0.16 ± 0.003	0.05 ± 0.002	0.03 ± 0.006
A-hERG[2]	0.04 ± 0.001	0.03 ± 0.003	0.02 ± 0.002	0.06 ± 0.008	0.03 ± 0.003	0.02 ± 0.003	0.08 ± 0.005	0.04 ± 0.013	0.02 ± 0.005
A-hERG[3]	0.05 ± 0.003	0.06 ± 0.001	0.05 ± 0.001	0.09 ± 0.016	0.06 ± 0.005	0.05 ± 0.008	0.06 ± 0.014	0.03 ± 0.014	0.03 ± 0.002
A-logD[1]	0.08 ± 0.001	0.05 ± 0.001	0.04 ± 0.001	0.05 ± 0.006	0.05 ± 0.002	0.04 ± 0.003	0.07 ± 0.002	0.05 ± 0.01	0.04 ± 0.003
A-logD[2]	0.08 ± 0.011	0.08 ± 0.009	0.05 ± 0.002	0.08 ± 0.018	0.08 ± 0.016	0.06 ± 0.006	0.11 ± 0.01	0.08 ± 0.005	0.06 ± 0.003
A-logD[3]	0.07 ± 0.002	0.03 ± 0.001	0.03 ± 0.001	0.05 ± 0.007	0.03 ± 0.003	0.03 ± 0.003	0.03 ± 0.011	0.06 ± 0.008	0.02 ± 0.003
A-MS[1]	0.08 ± 0.005	0.04 ± 0.001	0.05 ± 0.002	0.07 ± 0.014	0.04 ± 0.007	0.05 ± 0.02	0.06 ± 0.011	0.07 ± 0.008	0.04 ± 0.003
A-MS[2]	0.12 ± 0.003	0.05 ± 0.002	0.06 ± 0.002	0.1 ± 0.016	0.06 ± 0.007	0.06 ± 0.005	0.11 ± 0.016	0.06 ± 0.017	0.06 ± 0.001
A-MS[3]	0.07 ± 0.006	0.08 ± 0.006	0.05 ± 0.002	0.06 ± 0.012	0.07 ± 0.009	0.05 ± 0.005	0.04 ± 0.013	0.04 ± 0.009	0.04 ± 0.003

A.2 Enhancing UQ with Censored Regression Labels

This document provides details about the distributions of each assay in terms of feature space and label space, a description of the model selection including the HP optimization, the full results from the ablation study, and complementary results from the model comparison in terms of NLL, ENCE, calibration curves and c_v.

A.2.1 Temporal distributions

The following section details supporting information about each assay's distribution of the feature space and the label space over time. In Figure A.3 the distributions of the observed labels from every assay, are shown for each of the five folds that constitute the three temporal settings illustrated in Figure 3.1. Similarly, in Figure A.4 the feature space is shown as t-SNE projections and colored by the five temporal folds. Generally, ADME-T assays exhibit smaller shifts over time compared to TB assays. This is true in terms of both the feature space and the label space.

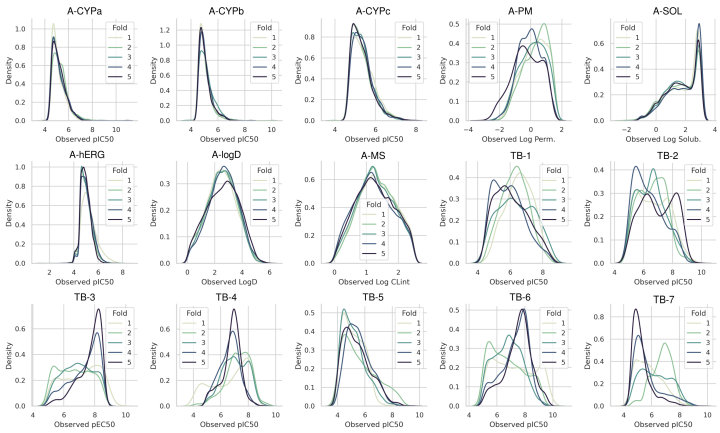

Figure A.3: **Temporal distribution of the label-space.** Observed experimental labels colored by temporal folds.

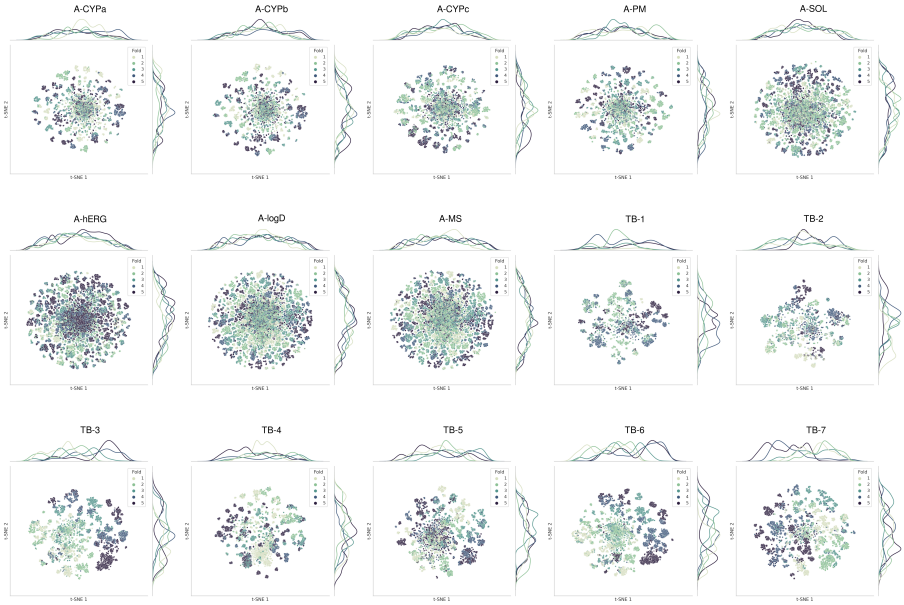

Figure A.4: **Temporal distribution of the feature-space.** A t-SNE projection of each assay colored by temporal fold.

A.2.2 Model selection

For the model selection of all models used in this work, we optimized the HPs described in Table A.9 using a grid search according to the loss function of each base model. Each temporal setting was optimized individually. As such, the RF model and the base NN architecture used in the Ensemble and the MC-Dropout model were all optimized using the validation MSE for observed datasets or the validation CensoredMSE described in the Methods section for censored datasets. For simplicity, we also used the resulting NN architecture for the Bayes by Backprop model and the evidential deep learning framework. The base NN architecture for the Gaussian models was optimized using the validation Gaussian NLL.

During the grid search for the NN architectures, we used a maximum of 100 epochs per experiment. However, a maximum of 500 epochs was used for the final experiments, inspired by the neural architecture search proposed by Jiang et al. (2024). Additionally, the NNs were trained using the Adam optimizer with a weight decay of 0.0005, the learning rate was reduced when plateauing with a patience of 50 epochs. The HP named *decreasing dimension* for the NNs determines if the hidden dimension is decreased by a factor of two for every layer. Thus, for HPs: 4 number of layers, hidden dimension of size 512, and decreasing dimension True, the resulting hidden layers have sizes [512, 256, 128, 64].

Table A.9: **Model selection.** Considered HP space for model selection of RF and the base NN architectures during grid search based on validation loss.

Base model	Hyperparameter	Explored space
Random Forest	n_estimators	{50, 100, 250, 500, 1000}
	min_samples_leaf	{2, 10, 0.25, 0.5, 0.75}
	min_samples_split	{1, 25, 50, 100, 250, 500}
Neural Network	Learning rate	{0.00005, 0.0001, 0.0005, 0.001}
	Scheduler Factor	{0.1, 0.5}
	Number of hidden layers	{2, 3, 4}
	Hidden dimension	{64, 128, 256, 512}
	Decreasing dimension	{False, True}
	Dropout	{0.25, 0.5, 0.75}

A.2.3 Full ablation study

Figure A.5 shows the ΔMSE and ΔNLL results on only observed test labels. Additionally, the detailed underlying results for Figure 3.12 and Figure A.5 are provided in Figure A.6 and A.7.

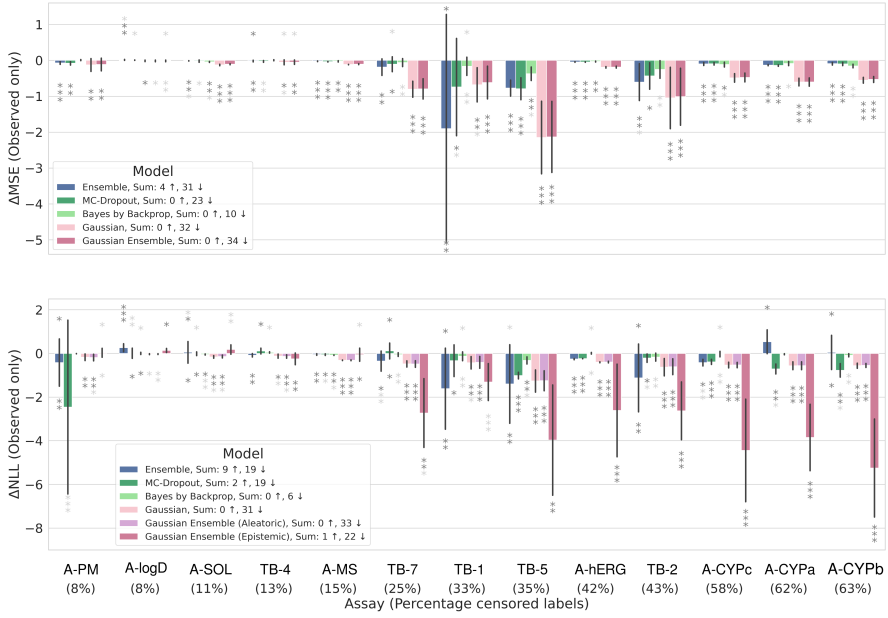

Figure A.5: **Ablation study on observed labels only.** Difference in MSE (top) and NLL (bottom) for models trained with and without censored labels. The error bars aggregate the three temporal settings. The stars above the bars indicate that the censored model was significantly better ($p < 0.05$), with Bonferroni correction in dark gray and without in light gray. In contrast, the stars below the bars indicate the opposite.

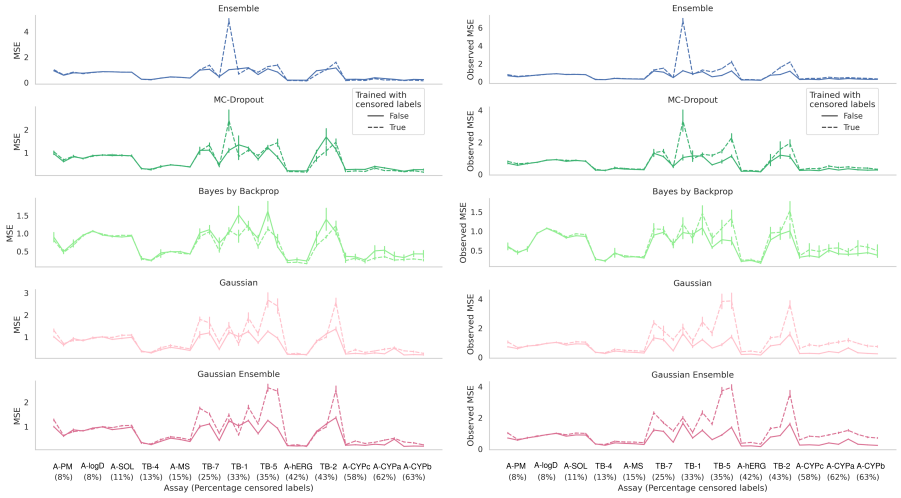

Figure A.6: **Full ablation study of MSE.** Comparison of MSE for models trained with and without additional censored labels.

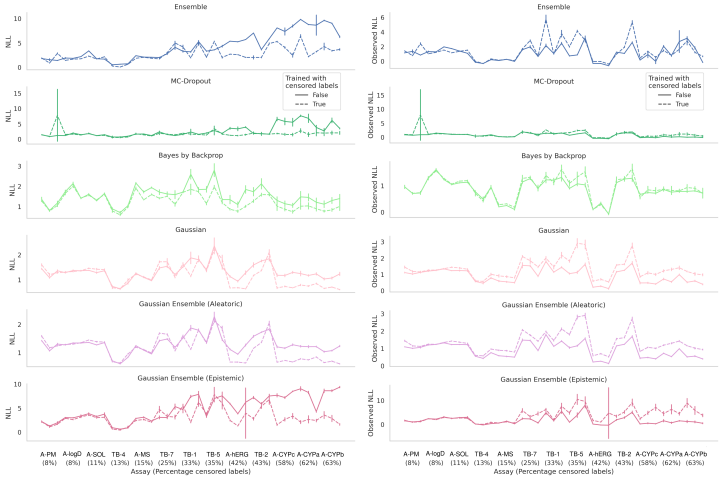

Figure A.7: **Full ablation study of NLL.** Comparison of NLL for models trained with and without additional censored labels.

A.2.4 Additional model comparison

In the following section, additional results from the model comparison are presented. First, the NLL scores for all aleatoric estimates are provided in Figure A.8. Next, the ENCE scores are provided in Figure A.9 to complement the NLL scores as a second metric for the global, intertwined evaluation of predictive performance and calibration of uncertainty estimates. Next, the confidence-based calibration curves from the first two temporal settings are shown in Figure A.10, as a complement to Figure 3.14. Finally, Figure A.11 presents the coefficients of dispersion for the uncertainty estimates by each model.

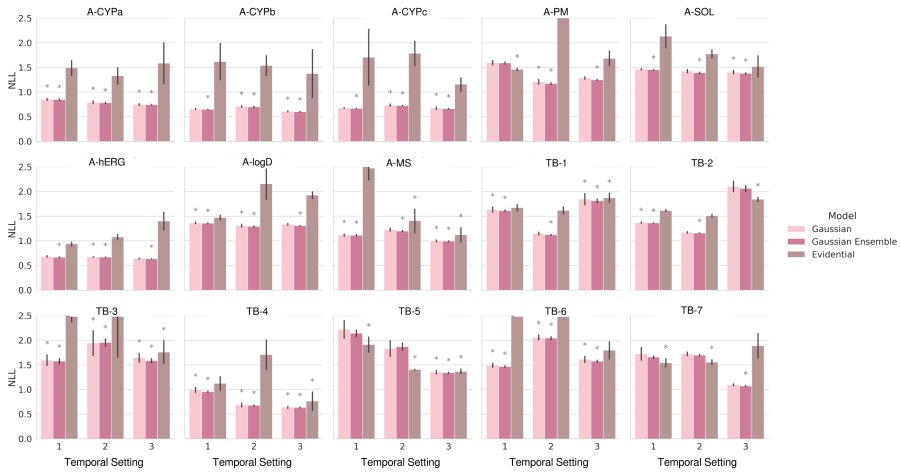

Figure A.8: **Combined accuracy of uncertainty estimation and predictive performance.** Comparing the NLL of all aleatoric uncertainty estimating models, aggregated over 10 experiments. For each dataset, the best model in terms of average NLL is marked with a star together with any other models not statistically worse based on a one-sided Mann-Whitney-Wilcoxon test with Bonferroni correction. Apart from the RF and Evidential models, all other models are trained with censored labels where available.

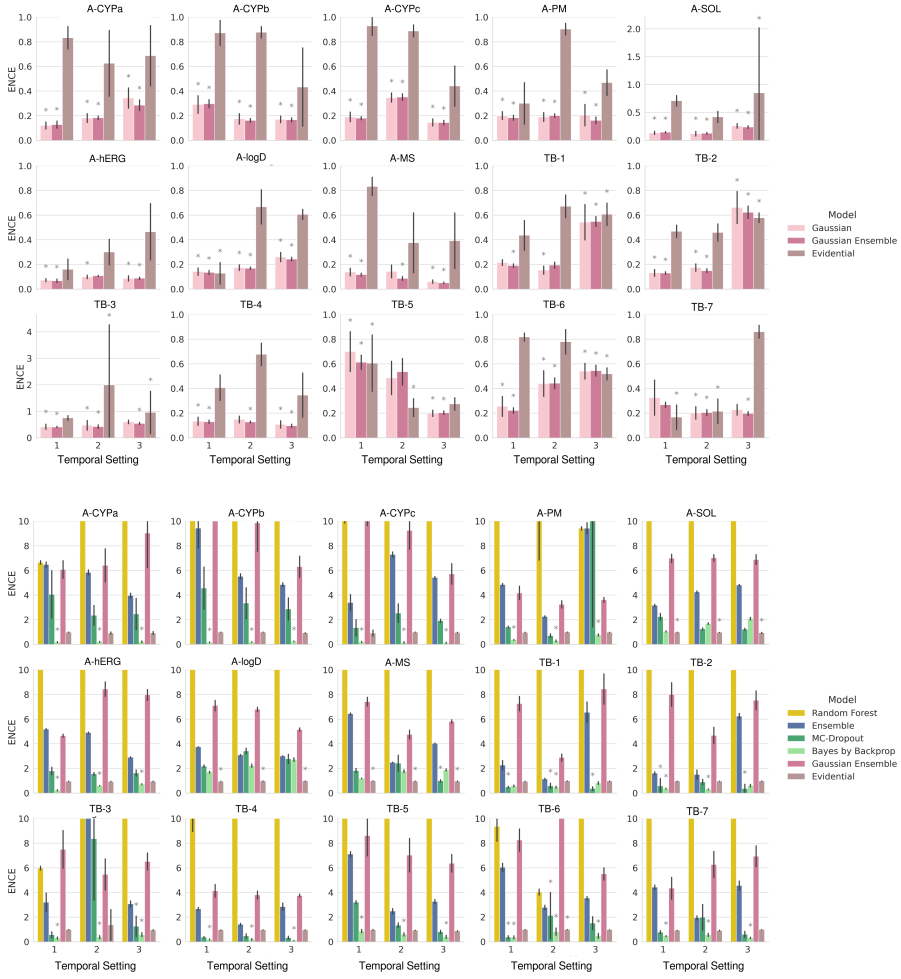

Figure A.9: **Combined accuracy of uncertainty estimation and predictive performance.** Comparing the ENCE of all aleatoric uncertainty estimating models (top) and all epistemic uncertainty estimating models (bottom), aggregated over 10 experiments. For each dataset, the best model in terms of average ENCE is marked with a star together with any other models that are not statistically worse based on a one-sided Mann-Whitney-Wilcoxon test with Bonferroni correction. Apart from the RF and Evidential models, all other models are trained with censored labels where available.

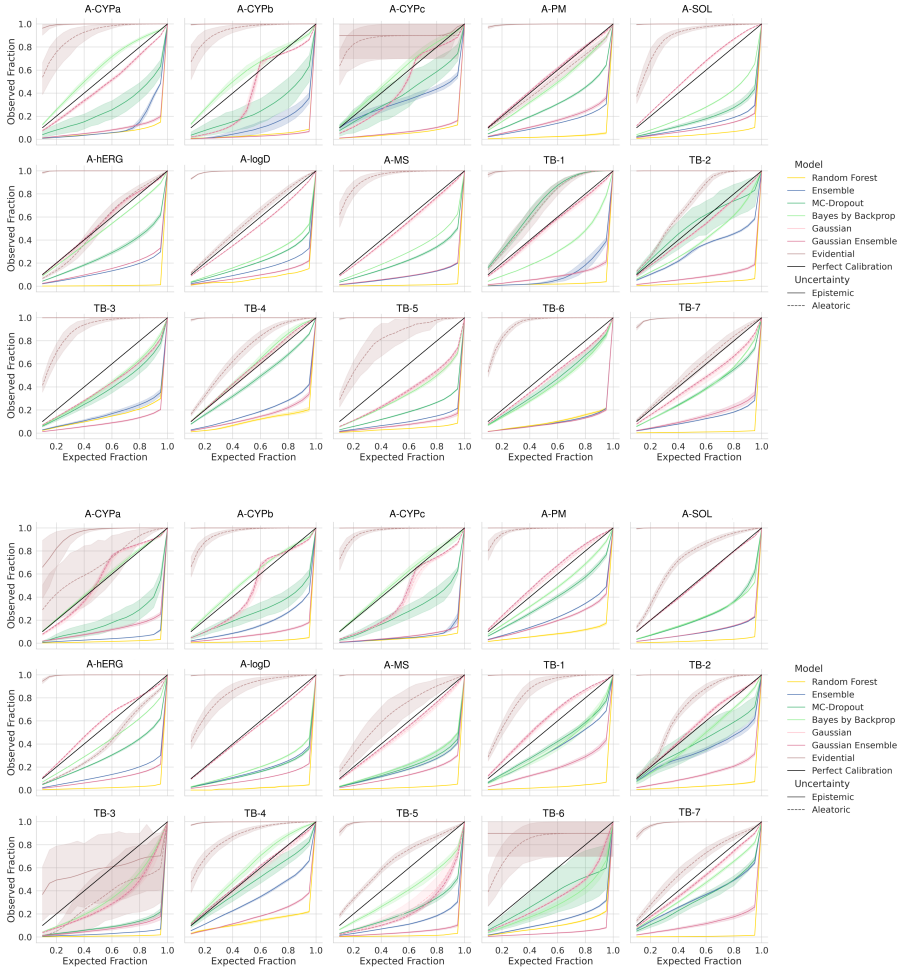

Figure A.10: **Confidence-based calibration curves.** Remaining calibration curves for all uncertainty estimates on the first (top) and second (bottom) temporal setting containing one and two folds respectively in the training set, aggregated over 10 experiments. Apart from the RF and Evidential models, all other models are trained with censored labels.



Figure A.11: **Coefficient of variation.** Comparing the c_v of all aleatoric uncertainty estimating models (top) and all epistemic uncertainty estimating models (bottom), aggregated over 10 experiments. Apart from the RF and Evidential models, all other models are trained with censored labels where available.

Appendix B

Appendix: Chapter 4

B.1 A Comprehensive Calibration Study via HP Tuning and HMC Bayesian Last Layer

B.1.1 Probability Calibration Study

Table B.1: **Results of the probability calibration study for all targets.** The performance of the models is evaluated on the test set. The model HPs were tuned to optimize the AUC score on a validation set. Calibration errors, BSs and AUC values are shown across targets for all probability calibration methods. Results are averaged over 10 model repetitions, except for the MLP ensemble models, for which 5 model repeats were computed.

HP-Metric: AUC

Target	Model	ECE	ACE	BS	AUC
CYP3A4	MLP	0.1799 ± 0.0111	0.1709 ± 0.0121	0.1951 ± 0.0103	0.7682 ± 0.0087
	MLP+PS	0.0415 ± 0.0072	0.0438 ± 0.0068	0.1586 ± 0.0048	0.7682 ± 0.0087
	MC–Dropout	0.2262 ± 0.0964	0.2262 ± 0.0948	0.2395 ± 0.0539	0.6968 ± 0.0276
	Ensemble	0.1142 ± 0.0017	0.1054 ± 0.0024	0.1602 ± 0.0002	0.7957 ± 0.0015
	Ensemble+PS	0.0551 ± 0.0046	0.0542 ± 0.004	0.1503 ± 0.0002	0.7957 ± 0.0015
	HBLL	0.0452 ± 0.0114	0.0448 ± 0.0103	0.1589 ± 0.0046	0.7684 ± 0.0083
	HBLL+PS	0.0452 ± 0.0114	0.0448 ± 0.0103	0.1589 ± 0.0046	0.7684 ± 0.0083
MAO-A	MLP	0.2379 ± 0.0185	0.2281 ± 0.017	0.252 ± 0.016	0.695 ± 0.0134
	MLP+PS	0.0574 ± 0.0125	0.0619 ± 0.0129	0.1926 ± 0.0058	0.695 ± 0.0134
	MC–Dropout	0.0902 ± 0.0322	0.0942 ± 0.0337	0.2063 ± 0.0123	0.6562 ± 0.0196
	Ensemble	0.1544 ± 0.0053	0.1437 ± 0.0029	0.2099 ± 0.0016	0.7187 ± 0.0028
	Ensemble+PS	0.0643 ± 0.0044	0.0578 ± 0.0101	0.1861 ± 0.0011	0.7187 ± 0.0028
	HBLL	0.0607 ± 0.0148	0.0647 ± 0.0128	0.1959 ± 0.0047	0.6824 ± 0.0139
	HBLL+PS	0.0607 ± 0.0148	0.0647 ± 0.0128	0.1959 ± 0.0047	0.6824 ± 0.0139
hERG	MLP	0.0763 ± 0.0061	0.0721 ± 0.0085	0.079 ± 0.0052	0.761 ± 0.0139
	MLP+PS	0.0102 ± 0.0024	0.015 ± 0.0023	0.0576 ± 0.0005	0.761 ± 0.0139
	MC–Dropout	0.3175 ± 0.1577	0.3172 ± 0.1578	0.1875 ± 0.1368	0.6205 ± 0.0526
	Ensemble	0.0539 ± 0.0033	0.0535 ± 0.0031	0.0662 ± 0.0008	0.8088 ± 0.0032
	Ensemble+PS	0.0126 ± 0.0007	0.0199 ± 0.0026	0.0559 ± 0.0002	0.8088 ± 0.0032
	HBLL	0.0126 ± 0.0007	0.0199 ± 0.0026	0.0559 ± 0.0002	0.8088 ± 0.0032
	HBLL+PS	0.0126 ± 0.0007	0.0199 ± 0.0026	0.0559 ± 0.0002	0.8088 ± 0.0032

Table B.2: **Results of the probability calibration study for all targets.** The performance of the models is evaluated on the test set. The model HPs were tuned to optimize the accuracy on a validation set. Calibration errors, BSs and AUC values are shown across targets for all model types. Results are averaged over 10 model repetitions, except for the MLP ensemble models, for which 5 model repeats were computed.

HP-Metric: ACC

Target	Model	ECE	ACE	BS	AUC
CYP3A4	MLP	0.1488 ± 0.0031	0.1435 ± 0.0036	0.1759 ± 0.0015	0.7798 ± 0.0055
	MLP+PS	0.037 ± 0.0053	0.0371 ± 0.004	0.1536 ± 0.0018	0.7798 ± 0.0055
	MC–Dropout	0.0453 ± 0.0123	0.0457 ± 0.0096	0.1559 ± 0.0021	0.7781 ± 0.0049
	Ensemble	0.1369 ± 0.0031	0.1333 ± 0.0032	0.1703 ± 0.0003	0.7877 ± 0.0008
	Ensemble+PS	0.0345 ± 0.0012	0.0342 ± 0.0012	0.1517 ± 0.0001	0.7877 ± 0.0008
	HBLL	0.0373 ± 0.0075	0.046 ± 0.0051	0.158 ± 0.0023	0.7741 ± 0.0067
	HBLL+PS	0.0373 ± 0.0075	0.046 ± 0.0051	0.158 ± 0.0023	0.7741 ± 0.0067
MAO-A	MLP	0.2207 ± 0.0232	0.2136 ± 0.0252	0.2391 ± 0.0224	0.7082 ± 0.0287
	MLP+PS	0.0544 ± 0.0159	0.0604 ± 0.015	0.188 ± 0.0107	0.7082 ± 0.0287
	MC–Dropout	0.1247 ± 0.0284	0.1275 ± 0.0256	0.2126 ± 0.0088	0.66 ± 0.0292
	Ensemble	0.139 ± 0.0028	0.1345 ± 0.0023	0.2005 ± 0.0011	0.728 ± 0.0038
	Ensemble+PS	0.0568 ± 0.0044	0.0628 ± 0.0059	0.1826 ± 0.001	0.728 ± 0.0038
	HBLL	0.0563 ± 0.0166	0.0638 ± 0.0137	0.1897 ± 0.0078	0.7039 ± 0.0292
	HBLL+PS	0.0563 ± 0.0166	0.0638 ± 0.0137	0.1897 ± 0.0078	0.7039 ± 0.0292
hERG	MLP	0.1028 ± 0.0991	0.1019 ± 0.0997	0.0792 ± 0.0324	0.6928 ± 0.0806
	MLP+PS	0.0091 ± 0.0016	0.0205 ± 0.0045	0.0591 ± 0.0024	0.6928 ± 0.0806
	MC–Dropout	0.2654 ± 0.0859	0.2645 ± 0.0866	0.138 ± 0.0486	0.6281 ± 0.078
	Ensemble	0.0851 ± 0.0135	0.0766 ± 0.0127	0.0626 ± 0.0023	0.741 ± 0.0064
	Ensemble+PS	0.0089 ± 0.0016	0.0217 ± 0.003	0.0566 ± 0.0003	0.741 ± 0.0064
	HBLL	0.0302 ± 0.0128	0.0304 ± 0.0128	0.0595 ± 0.0036	0.6917 ± 0.0835
	HBLL+PS	0.0302 ± 0.0128	0.0304 ± 0.0128	0.0595 ± 0.0036	0.6917 ± 0.0835

Table B.3: **Results of the probability calibration study for all targets.** The performance of the models is evaluated on the test set. The model HPs were tuned to optimize the BCE loss on a validation set. Calibration errors, BSs and AUC values are shown across targets for all model types. Results are averaged over 10 model repetitions, except for the MLP ensemble models, for which 5 model repeats were computed.

HP-Metric: BCE loss

Target	Model	ECE	ACE	BS	AUC
CYP3A4	MLP	0.0698 ± 0.0056	0.0663 ± 0.005	0.1506 ± 0.0014	0.7975 ± 0.0028
	MLP+PS	0.0373 ± 0.0036	0.039 ± 0.0046	0.1469 ± 0.0007	0.7975 ± 0.0028
	MC–Dropout	0.1476 ± 0.0246	0.1523 ± 0.0225	0.1783 ± 0.0095	0.7797 ± 0.0077
	Ensemble	0.0674 ± 0.0012	0.0611 ± 0.001	0.1496 ± 0.0004	0.8004 ± 0.0008
	Ensemble+PS	0.0366 ± 0.0018	0.0375 ± 0.001	0.1461 ± 0.0003	0.8004 ± 0.0008
	HBLL	0.0585 ± 0.0333	0.0604 ± 0.0327	0.1521 ± 0.0079	0.7966 Âś 0.0029
	HBLL+PS	0.0393 ± 0.0063	0.0398 ± 0.0048	0.1473 ± 0.0008	0.7966 Âś 0.0029
MAO-A	MLP	0.1696 ± 0.0116	0.1663 ± 0.0122	0.212 ± 0.006	0.7219 ± 0.0057
	MLP+PS	0.0473 ± 0.0084	0.0455 ± 0.0061	0.1838 ± 0.0017	0.7219 ± 0.0057
	MC–Dropout	0.1268 ± 0.0061	0.1259 ± 0.0067	0.2019 ± 0.0043	0.7142 ± 0.0081
	Ensemble	0.1729 ± 0.0016	0.1701 ± 0.0007	0.2124 ± 0.0006	0.7212 ± 0.0007
	Ensemble+PS	0.0428 ± 0.0011	0.0446 ± 0.0044	0.1838 ± 0.0002	0.7212 ± 0.0007
	HBLL	0.0465 ± 0.0061	0.0439 ± 0.0037	0.1851 ± 0.0018	0.7254 Âś 0.0054
	HBLL+PS	0.0355 ± 0.0061	0.0318 ± 0.0049	0.1838 ± 0.0017	0.7254 Âś 0.0054
hERG	MLP	0.0289 ± 0.0079	0.0254 ± 0.0063	0.0541 ± 0.0018	0.8061 ± 0.0031
	MLP+PS	0.0148 ± 0.0023	0.0204 ± 0.0017	0.0547 ± 0.0003	0.8061 ± 0.0031
	MC–Dropout	0.0815 ± 0.0186	0.0792 ± 0.0191	0.0607 ± 0.0043	0.8061 ± 0.0072
	Ensemble	0.0294 ± 0.0012	0.0285 ± 0.001	0.0541 ± 0.0002	0.8083 ± 0.0002
	Ensemble+PS	0.0133 ± 0.0006	0.021 ± 0.0004	0.0545 ± 0.0001	0.8083 ± 0.0002
	HBLL	0.0111 ± 0.0037	0.0112 ± 0.0066	0.0534 ± 0.0009	0.7991 Âś 0.0053
	HBLL+PS	0.0133 ± 0.001	0.02 ± 0.002	0.055 ± 0.0004	0.7991 Âś 0.0053

Table B.4: **Results of the probability calibration study for all targets.** The performance of the models is evaluated on the test set. Calibration errors, BSs and AUC values are shown across targets for all probability model types. The model HPs were tuned to optimize the ACE on a validation set. Results are averaged over 10 model repetitions, except for the MLP ensemble models, for which 5 model repeats were computed.

HP-Metric: ACE

Target	Model	ECE	ACE	BS	AUC
CYP3A4	MLP	0.068 ± 0.0062	0.0635 ± 0.0073	0.1548 ± 0.0039	0.7872 ± 0.0072
	MLP+PS	0.0419 ± 0.0118	0.0429 ± 0.0119	0.1522 ± 0.0031	0.7872 ± 0.0072
	MC–Dropout	0.2551 ± 0.0158	0.2542 ± 0.0142	0.2437 ± 0.0075	0.6717 ± 0.0267
	Ensemble	0.0457 ± 0.0026	0.0415 ± 0.0007	0.1451 ± 0.0004	0.8043 ± 0.0007
	Ensemble+PS	0.0423 ± 0.0035	0.0435 ± 0.0037	0.1451 ± 0.0004	0.8043 ± 0.0007
	HBLL	0.0461 ± 0.0062	0.047 ± 0.0047	0.1536 ± 0.003	0.7872 ± 0.0078
	HBLL+PS	0.0461 ± 0.0062	0.047 ± 0.0047	0.1536 ± 0.003	0.7872 ± 0.0078
MAO-A	MLP	0.0999 ± 0.0076	0.0962 ± 0.009	0.1808 ± 0.0021	0.7461 ± 0.0035
	MLP+PS	0.0642 ± 0.0044	0.0677 ± 0.0069	0.1744 ± 0.0011	0.7461 ± 0.0035
	MC–Dropout	0.0671 ± 0.0064	0.0677 ± 0.0098	0.1771 ± 0.0018	0.7396 ± 0.0051
	Ensemble	0.085 ± 0.0025	0.0921 ± 0.0012	0.1797 ± 0.0004	0.7469 ± 0.0007
	Ensemble+PS	0.0713 ± 0.0027	0.0703 ± 0.0015	0.1743 ± 0.0002	0.7469 ± 0.0007
	HBLL	0.0638 ± 0.006	0.0848 ± 0.0073	0.1786 ± 0.0017	0.746 ± 0.0035
	HBLL+PS	0.0638 ± 0.006	0.0848 ± 0.0073	0.1786 ± 0.0017	0.746 ± 0.0035
hERG	MLP	0.0328 ± 0.0109	0.0317 ± 0.0111	0.0586 ± 0.0033	0.7742 ± 0.0348
	MLP+PS	0.0112 ± 0.0051	0.0166 ± 0.0046	0.0569 ± 0.0015	0.7742 ± 0.0348
	MC–Dropout	0.4349 ± 0.1518	0.4343 ± 0.1531	0.2737 ± 0.1126	0.5771 ± 0.0763
	Ensemble	0.0157 ± 0.002	0.0165 ± 0.0031	0.0533 ± 0.0003	0.8154 ± 0.0024
	Ensemble+PS	0.012 ± 0.0013	0.0195 ± 0.0014	0.0542 ± 0.0002	0.8154 ± 0.0024
	HBLL	0.0188 ± 0.0083	0.0224 ± 0.0096	0.0573 ± 0.0027	0.7797 ± 0.0268
	HBLL+PS	0.0188 ± 0.0083	0.0224 ± 0.0096	0.0573 ± 0.0027	0.7797 ± 0.0268

B.1.2 Additional Experiments

Performance of deeper MLPs.

Nr. of Layers	Validation Loss	Validation AUC
1	0.2956 ± 0.0014	0.9191 ± 0.0009
2	0.2932 ± 0.0043	0.9180 ± 0.0021
3	0.3051 ± 0.0032	0.9118 ± 0.0005
4	0.3151 ± 0.0062	0.9180 ± 0.0024

Table B.5: BCE loss and AUC on the validation dataset of the hERG model with different numbers of layers. The average and SD of 10 model repeats are shown.

Model Performance on Training vs. Test set

	AUC		BS		ACE	
HP Metric	Train	Test	Train	Test	Train	Test
CYP3A4						
ACC	0.9999 ± 0.0001	0.7798 ± 0.0055	0.0020 ± 0.0001	0.1759 ± 0.0015	0.0048 ± 0.0003	0.1435 ± 0.0036
AUC	0.9999 ± 0.0001	0.7682 ± 0.0087	0.0023 ± 0.0002	0.1951 ± 0.0103	0.0009 ± 0.0002	0.1709 ± 0.0121
BCE	0.9967 ± 0.0002	0.7975 ± 0.0028	0.0299 ± 0.0003	0.1506 ± 0.0014	0.0503 ± 0.0026	0.0663 ± 0.005
ACE	0.9800 ± 0.0030	0.7872 ± 0.0072	0.0516 ± 0.0017	0.1548 ± 0.0039	0.0314 ± 0.003	0.0635 ± 0.0073
MAO-A						
ACC	0.9999 ± 0.0001	0.7082 ± 0.0287	0.0040 ± 0.0001	0.2391 ± 0.0224	0.0006 ± 0.0001	0.2136 ± 0.0252
AUC	0.9999 ± 0.0001	0.6950 ± 0.0134	0.0039 ± 0.0001	0.2520 ± 0.016	0.0002 ± 0.0017	0.2281 ± 0.0134
BCE	0.9996 ± 0.0001	0.7219 ± 0.0057	0.0078 ± 0.0002	0.2120 ±0.006	0.0031 0.0002	0.1663 ± 0.0122
ACE	0.9961 ± 0.0002	0.7461 ± 0.0035	0.0299 ± 0.0004	0.1808 ± 0.0021	0.0486 ± 0.0007	0.0962 ± 0.0009
hERG						
ACC	0.9999 ± 0.0001	0.6928 ± 0.0134	0.0009 ± 0.0004	0.0792 ± 0.0324	0.0001 ± 0.0008	0.1019 ± 0.0997
AUC	0.9999 ± 0.0001	0.7610 ± 0.0139	0.0012 ± 0.0002	0.0790 ± 0.0052	0.0001 ± 0.0001	0.0721 ± 0.0085
BCE	0.9994 ± 0.0001	0.8061 ± 0.0031	0.0055 ± 0.0001	0.0541 ± 0.0018	0.0027 ± 0.0002	0.0254 ± 0.0063
ACE	0.9409 ± 0.001	0.7742 ± 0.0348	0.0392 ± 0.0005	0.0586 ± 0.0033	0.0070 ± 0.0005	0.0586 ± 0.0111

Table B.6: Model performance on the training vs test set for models trained with the CYP3A4, MAO-A, and hERG datasets. Models trained with different HP metrics (ACC, AUC, BCE, ACE) are shown. Model performance is reported in terms of AUC, BS, and ACE. Averages and SDs of 10 model repeats are reported.

Table B.7: **Overview of the parameter space screened during HP tuning of the baseline MLP model.** The HP were tuned for each dataset using four different HP metrics. To assess the performance on the validation set, the average of ten model repetitions was obtained. The overview presents the range of HP explored during the tuning process.

Model	Hyperparameter	Explored Space
Baseline MLP	Learning Rate	{1e-5, 1r-4, 1e-3, 1e-2, 0.1}
	Hidden Size	{5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130}
	Weight Decay	{1e-5, 1r-4, 1e-3, 1e-2, 0.1}
	Dropout	{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

B.1.3 Hyperparameter Tuning

Table B.8: **Overview of the parameter space used for HBLL tuning.** The prior of the parameters from a Bayesian Logistic regression was tuned, which was used as the last layer in the HBLL models. More precisely, the precision of the Gaussian prior of the model parameters was optimized using the validation set.

Model	Hyperparameter	Explored Space
HBLL	Precision of the Gaussian Prior	{100, 150, 200, 250, 300, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000}

B.1.4 Computation Time of Models

Table B.9: **Time required for models included in the study.** The left column lists the time needed for training (T_{train}), and the right column the time needed for inference ($T_{inference}$). The meanings of the abbreviations used in the table are provided below. Typical training times for models (T_{train}) used in this study are listed in the last column.

Model	T_{train}	$T_{inference}$	Typical Training Time
MLP	$n_m * (T_f + T_b)$	T_f	$\sim$ 7:30 min
MC–Dropout	$n_m * (T_f + T_b)$	$n_f * T_f$	$\sim$ 7:30 min
Ensemble	$n_e * n_m * (T_f + T_b)$	$n_e * T_f$	$\sim$ 188 min
HBLL	$n_m * (T_f + T_b) + L * n_s * (T_{flast} + T_s)$	$n_s * T_f$	$\sim$ 175 min

Abbreviations
1) Required time T for ...
T_f: one forward pass, T_b: one backpropagation step,
T_s: one HMC step, T_{flast}: forward pass of last layer
2) Number of..
n_m: mini-batches, n_f: forward passes in MC–Dropout, n_e: base estimators in Ensemble, n_s: samples in HMC, L: simulation steps in HMC

B.2 Fully Bayesian NN with HMC Sampling for UQ in Drug-Target Interaction Prediction

B.2.1 Model Selection.

Table B.10: **HP tuning of the baseline MLP.** The HP of the MLPs were tuned for each dataset using the average BCE loss of ten model repetitions on a validation dataset.

Model	Hyperparameter	Explored Space
Baseline MLP	Nr. of Layers	{1, 2, 3, 4, 5}
	Hidden Size	{4, 6 ,8 ,10 ,12 , 14, 16, 18, 20, 22, 24, 26, 28, 30}
	Learning Rate	{1r-4, 1e-3, 1e-2, 0.1}
	Weight Decay	{1r-4, 1e-3, 1e-2, 0.1}

Table B.11: **Hyperparameter configuration of the base MLPs.** The HPs that minimize the BCE loss on a validation dataset are shown or each target.

Hyperparameter	CYP3A4	MAO-A	hERG
Nr. of Layers	1	2	1
Hidden Size	4	4	14
Learning Rate	0.01	0.1	0.1
Weight Decay	0.01	0.0001	0.001

B.2.2 Evaluation of HMC Chains.

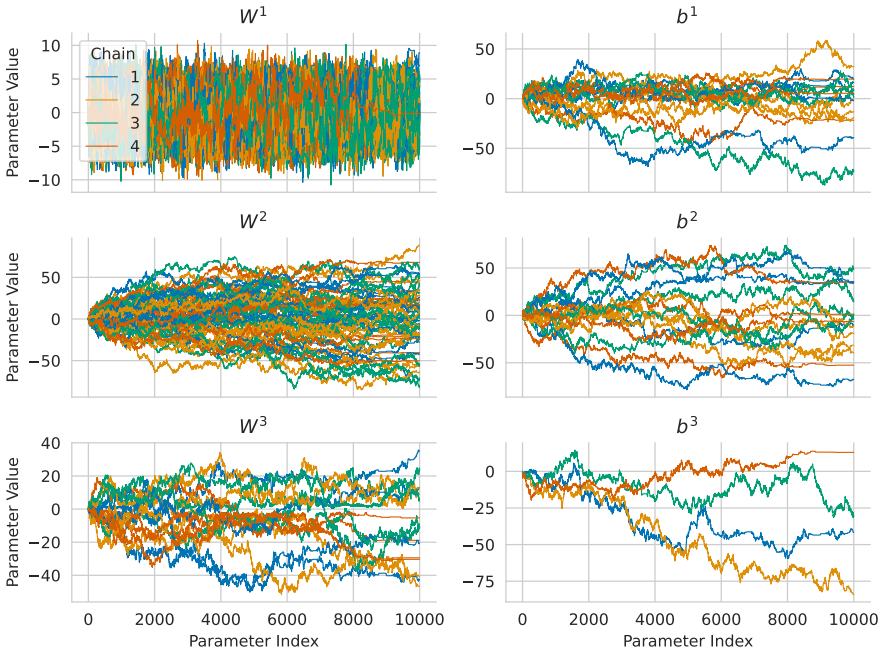

Figure B.1: **Traceplot of the MAO-A assay.** The traceplot of the first 10000 parameter samples for the MAO-A model is shown. The values for each parameter subset are shown in separate plot. The values are plotted for evolving HMC chains, indicated by increasing parameter indices. The chains responsible for the generated sample are marked by color.

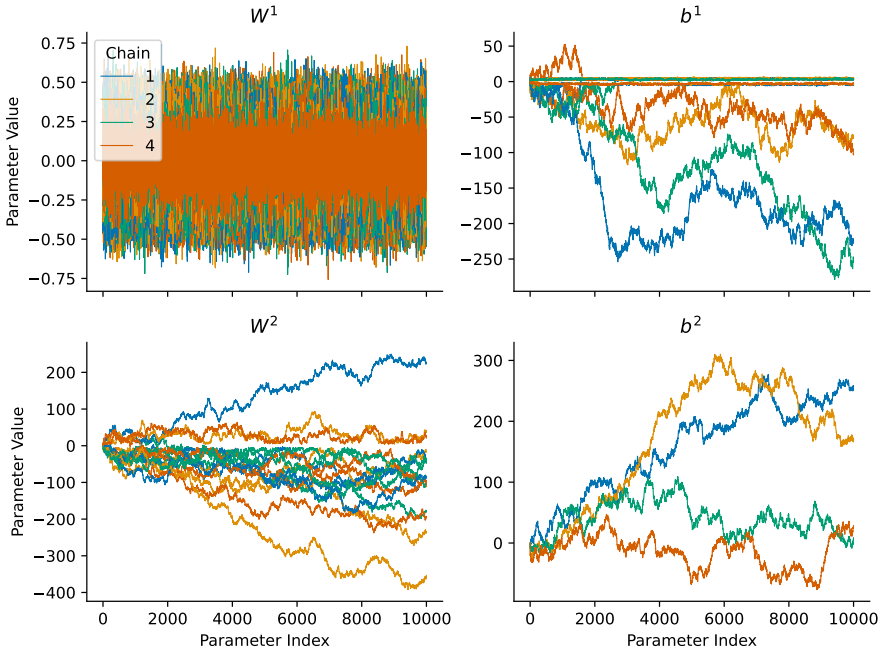

Figure B.2: **Traceplot of the CYP3A4 assay.** The traceplot of the first 10000 parameter samples for the CYP3A4 model is shown. The values for each parameter subset are shown in separate plots. The values are plotted for evolving HMC chains, indicated by increasing parameter indices. The chains responsible for the generated sample are marked by color.

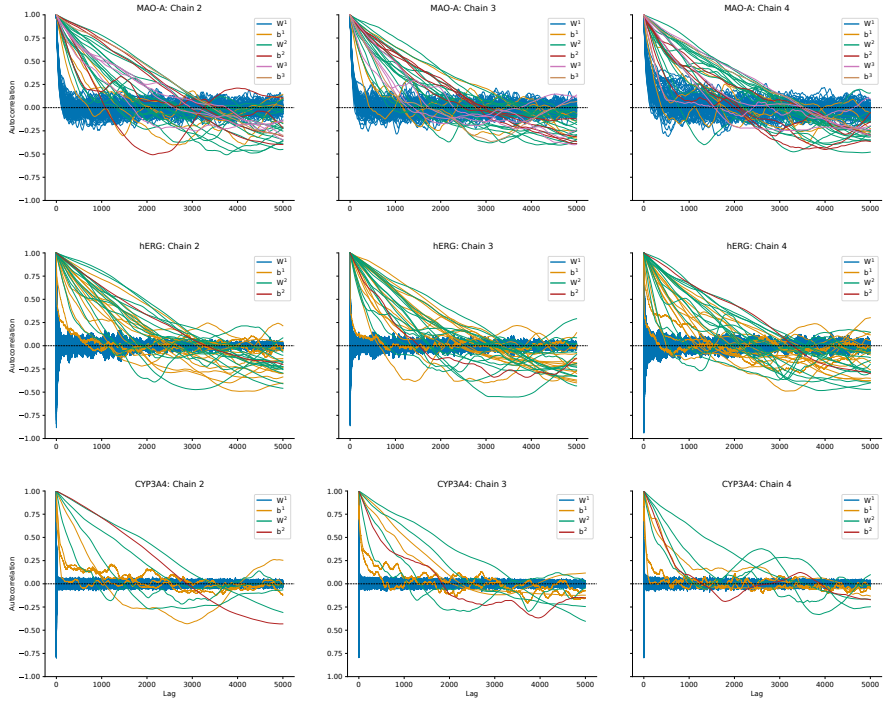

Figure B.3: **Chain 2–4: Autocorrelation between HMC samples across assays.** The autocorrelation between the fist 5000 HMC samples of the second, third and fourth chain is shown for each assay. Subsets of parameters are distinguished by color coding.

Bibliography

Alley, E.C., Khimulya, G., Biswas, S., AlQuraishi, M., Church, G.M., 2019. Unified rational protein engineering with sequence-based deep representation learning. Nature methods 16, 1315–1322.

Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., Aljaaf, A.J., 2020. A systematic review on supervised and unsupervised machine learning algorithms for data science. Springer. pp. 3–21.

Amini, A., Schwarting, W., Soleimany, A., Rus, D., 2020. Deep Evidential Regression, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 14927–14937.

Angiolini, L., Manetti, F., Spiga, O., Tafi, A., Visibelli, A., Petricci, E., 2025. Machine learning for predicting the drug-to-antibody ratio (dar) in the synthesis of antibody–drug conjugates (adcs). Journal of Chemical Information and Modeling .

Apostolakis, G., 1990. The Concept of Probability in Safety Assessments of Technological Systems. Science 250, 1359–1364. URL: http://www.jstor.org/stable/2878382.

Arany, A., Simm, J., Oldenhof, M., Moreau, Y., 2022. SparseChem: Fast and Accurate Machine Learning Model for Small Molecules. arXiv preprint arXiv:2203.04676 .

Begoli, E., Bhattacharya, T., Kusnezov, D., 2019. The need for uncertainty quantification in machine-assisted medical decision making. Nature Machine Intelligence 1, 20 – 23. doi:doi:10.1038/s42256-018-0004-1.

Bejani, M.M., Ghatee, M., 2021. A systematic review on overfitting control in shallow and deep neural networks. Artificial Intelligence Review 54, 6391–6438.

Bengs, V., Hüllermeier, E., Waegeman, W., 2023. On second-order scoring rules for epistemic uncertainty quantification, in: International Conference on Machine Learning, PMLR. pp. 2078–2091.

Benker, M., Furtner, L., Semm, T., Zaeh, M.F., 2021. Utilizing uncertainty information in remaining useful life estimation via bayesian neural networks and hamiltonian monte carlo. Journal of Manufacturing Systems 61, 799–807.

Bertolini, M., Clevert, D.A., Montanari, F., 2023. Explaining, Evaluating and Enhancing Neural NetworksâĂŹ Learned Representations, in: International Conference on Artificial Neural Networks, Springer. pp. 269–287.

Betancourt, M., 2017. A conceptual introduction to hamiltonian monte carlo. arXiv preprint arXiv:1701.02434 .

Bingham, E., Chen, J.P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., Goodman, N.D., 2019. Pyro: Deep universal probabilistic programming. Journal of machine learning research 20, 1–6.

Bishop, C.M., 1994. Mixture Density Networks. Technical Report. Aston University, Birmingham .

Bishop, C.M., Nasrabadi, N.M., 2006. Pattern recognition and machine learning. volume 4. Springer.

Bleakley, K., Yamanishi, Y., 2009. Supervised Prediction of Drug–Target Interactions Using Bipartite Local Models. Bioinform. 25, 2397–2403.

Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D., 2015. Weight Uncertainty in Neural Network, in: International Conference on Machine Learning, PMLR. pp. 1613–1622.

Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.P., Schölkopf, B., Smola, A.J., 2006. Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics 22, e49–e57.

Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

Brier, G.W., 1950. Verification of forecasts expressed in terms of probability. Monthly Weather Review 78, 1 – 3. URL: https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

Bröcker, J., 2009. Reliability, sufficiency, and the decomposition of proper scores. Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography 135, 1512–1519.

Brown, S.P., Muchmore, S.W., Hajduk, P.J., 2009. Healthy skepticism: assessing realistic model performance. Drug discovery today 14, 420–427.

Brunzema, P., Jordahn, M., Willes, J., Trimpe, S., Snoek, J., Harrison, J., 2024. Variational last layers for bayesian optimization, in: NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty.

Busk, J., Jørgensen, P.B., Bhowmik, A., Schmidt, M.N., Winther, O., Vegge, T., 2021. Calibrated Uncertainty for Molecular Property Prediction using Ensembles of Message Passing Neural Networks. Machine Learning: Science and Technology 3, 015012.

Cao, L., 2022. Beyond iid: Non-iid thinking, informatics, and learning. IEEE Intelligent Systems 37, 5–17.

Caruana, R., 1997. Multitask learning. Machine learning 28, 41–75.

Chapelle, O., Schölkopf, B., Zien, A., 2006. Semi-Supervised Learning. The MIT Press. URL: https://doi.org/10.7551/mitpress/9780262033589.001.0001.

Chauhan, V.K., Thakur, A., OâĂŹDonoghue, O., Rohanian, O., Molaei, S., Clifton, D.A., 2024. Continuous patient state attention model for addressing irregularity in electronic health records. BMC Medical Informatics and Decision Making 24, 117.

Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., Blaschke, T., 2018. The rise of deep learning in drug discovery. Drug Discovery Today 23, 1241–1250. URL: https://doi.org/10.1016/j.drudis.2018.01.039.

Chen, M., Borlak, J., Tong, W., 2013. High lipophilicity and high daily dose of oral medications are associated with significant risk for drug-induced liver injury. Hepatology 58, 388–396.

Cherkasov, A., Muratov, E.N., Fourches, D., Varnek, A., Baskin, I.I., Cronin, M., Dearden, J., Gramatica, P., Martin, Y.C., Todeschini, R., et al., 2014. Qsar modeling: where have you been? where are you going to? Journal of medicinal chemistry 57, 4977–5010.

Choi, S., Lee, K., Lim, S., Oh, S., 2018. Uncertainty-Aware Learning from Demonstration Using Mixture Density Networks with Sampling-Free Variance Modeling, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 6915–6922.

Cobb, A.D., Baydin, A.G., Jalaian, B., 2023. hamiltorch. URL: https://github.com/AdamCobb/hamiltorch.

Cobb, A.D., Jalaian, B., 2021. Scaling hamiltonian monte carlo inference for bayesian neural networks with symmetric splitting, in: Uncertainty in Artificial Intelligence, PMLR. pp. 675–685.

Coley, C.W., Green, W.H., Jensen, K.F., 2018. Machine learning in computer-aided synthesis planning. Accounts of chemical research 51, 1281–1289.

Coley, C.W., Thomas III, D.A., Lummiss, J.A., Jaworski, J.N., Breen, C.P., Schultz, V., Hart, T., Fishman, J.S., Rogers, L., Gao, H., et al., 2019. A robotic platform for flow synthesis of organic compounds informed by ai planning. Science 365, eaax1566.

Dahl, G.E., Jaitly, N., Salakhutdinov, R., 2014. Multi-task neural networks for qsar predictions. arXiv preprint arXiv:1406.1231 .

Dara, S., Dhamercherla, S., Jadav, S.S., Babu, C.M., Ahsan, M.J., 2022. Machine learning in drug discovery: a review. Artificial intelligence review 55, 1947–1999.

Deodhar, M., Al Rihani, S.B., Arwood, M.J., Darakjian, L., Dow, P., Turgeon, J., Michaud, V., 2020. Mechanisms of cyp450 inhibition: understanding drug-drug interactions due to mechanism-based inhibition in clinical practice. Pharmaceutics 12, 846.

Depeweg, S., Hernandez-Lobato, J.M., Doshi-Velez, F., Udluft, S., 2018. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning, in: International conference on machine learning, PMLR. pp. 1184–1193.

Der Kiureghian, A., Ditlevsen, O., 2009. Aleatory or epistemic? does it matter? Structural safety 31, 105–112.

Dheur, V., Taieb, S.B., 2023. A large-scale study of probabilistic calibration in neural network regression, in: International Conference on Machine Learning, PMLR. pp. 7813–7836.

Di, L., Fish, P.V., Mano, T., 2012. Bridging solubility between drug discovery and development. Drug discovery today 17, 486–495.

Di, L., Kerns, E.H., Hong, Y., Kleintop, T.A., Mc Connell, O.J., Huryn, D.M., 2003. Optimization of a higher throughput microsomal stability screening assay for profiling drug discovery candidates. SLAS Discovery 8, 453–462.

Dietterich, T., 1995. Overfitting and undercomputing in machine learning. ACM computing surveys (CSUR) 27, 326–327.

Dietterich, T.G., et al., 2002. Ensemble learning. The handbook of brain theory and neural networks 2, 110–125.

DiMasi, J.A., 2001. Risks in new drug development: approval success rates for investigational drugs. Clinical Pharmacology & Therapeutics 69, 297–307.

Dundar, M., Krishnapuram, B., Bi, J., Rao, R.B., 2007. Learning classifiers when the training data is not iid., in: IJCAI, Citeseer. pp. 756–61.

Dunn, O.J., 1961. Multiple Comparisons Among Means. J. Am. Stat. Assoc. 56, 52–64.

Dutschmann, T.M., Kinzel, L., Ter Laak, A., Baumann, K., 2023. Large-Scale Evaluation of k-Fold Cross-Validation Ensembles for Uncertainty Estimation. J. Cheminf. 15, 49. URL: https://doi.org/10.1186/s13321-023-00709-9.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al., 2021. Prottrans: Toward understanding the language of life through self-supervised learning. IEEE transactions on pattern analysis and machine intelligence 44, 7112–7127.

Engkvist, O., Mervin, L.H., Chen, H., Ran, T., 2023. Machine learning in drug design, in: The Handbook of Medicinal Chemistry: Principles and Practice. The Royal Society of Chemistry. URL: https://doi.org/10.1039/9781788018982-00231.

Fan, Y.J., Allen, J.E., McLoughlin, K.S., Shi, D., Bennion, B.J., Zhang, X., Lightstone, F.C., 2023. Evaluating point-prediction uncertainties in neural networks for protein-ligand binding prediction. Artificial intelligence chemistry 1, 100004.

Fan, Z., Yu, J., Zhang, X., Chen, Y., Sun, S., Zhang, Y., Chen, M., Xiao, F., Wu, W., Li, X., et al., 2024. Reducing overconfident errors in molecular property classification using posterior network. Patterns .

Fiedler, F., Lucia, S., 2023. Improved uncertainty quantification for neural networks with bayesian last layer. IEEE Access .

Fluetsch, A., Di Lascio, E., Gerebtzoff, G., Rodríguez-Pérez, R., 2024. Adapting Deep Learning QSPR Models to Specific Drug Discovery Projects. Mol. Pharmaceutics 21, 1817–1826. URL: https://doi.org/10.1021/acs.molpharmaceut.3c01124.

Fort, S., Hu, H., Lakshminarayanan, B., 2020. Deep ensembles: A loss landscape perspective. URL: https://arxiv.org/abs/1912.02757, arXiv:1912.02757.

Friesacher, H.R., Engkvist, O., Mervin, L., Moreau, Y., Arany, A., 2025a. Achieving well-informed decision-making in drug discovery: a comprehensive calibration study using neural network-based structure-activity models. Journal of Cheminformatics 17, 29.

Friesacher, H.R., Svensson, E., Arany, A., Mervin, L., Engkvist, O., 2024. Temporal evaluation of probability calibration with experimental errors, in: International Workshop on AI in Drug Discovery, Springer. pp. 13–20.

Friesacher, H.R., Svensson, E., Winiwarter, S., Mervin, L., Arany, A., Engkvist, O., 2025b. Temporal distribution shift in real-world pharmaceutical data: Implications for uncertainty quantification in qsar models. Artificial Intelligence in the Life Sciences 8, 100132. URL: https://www.sciencedirect.com/science/article/pii/S266731852500008X.

Furge, L.L., Guengerich, F.P., 2006. Cytochrome p450 enzymes in drug metabolism and chemical toxicology: An introduction. Biochemistry and Molecular Biology Education 34, 66–74.

Gal, Y., 2016. Uncertainty in Deep Learning. PhD thesis, Department of Engineering, University of Cambridge .

Gal, Y., Ghahramani, Z., 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, in: International Conference on Machine Learning, PMLR. pp. 1050–1059.

Gao, Q., Miedema, D.C., Zhao, Y., Weber, J.M., Tao, Q., Schweidtmann, A.M., 2025. Bayesian uncertainty quantification of graph neural networks using stochastic gradient hamiltonian monte carlo. Systems and Control Transactions , 1360–1364.

Gao, R., Liu, F., Zhang, J., Han, B., Liu, T., Niu, G., Sugiyama, M., 2021. Maximum mean discrepancy test is aware of adversarial attacks, in: International Conference on Machine Learning, PMLR. pp. 3564–3575.

Garg, S., Wu, Y., Balakrishnan, S., Lipton, Z., 2020. A unified view of label shift estimation. Advances in Neural Information Processing Systems 33, 3290–3300.

Gawlikowski, J., Tassi, C.R.N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., et al., 2023. A survey of uncertainty in deep neural networks. Artificial Intelligence Review 56, 1513–1589.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., Rubin, D., 2013. Bayesian Data Analysis, Third Edition. Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis. URL: https://books.google.at/books?id=ZXL6AQAAQBAJ.

Gelman, A., Rubin, D.B., 1992. Inference from iterative simulation using multiple sequences. Statistical science 7, 457–472.

Genheden, S., Thakkar, A., Chadimová, V., Reymond, J.L., Engkvist, O., Bjerrum, E., 2020. Aizynthfinder: a fast, robust and flexible open-source software for retrosynthetic planning. Journal of cheminformatics 12, 70.

Geyer, C.J., 2011. Introduction to markov chain monte carlo. Handbook of markov chain monte carlo 20116022, 22.

Geylan, G., De Maria, L., Engkvist, O., David, F., Norinder, U., 2024. A methodology to correctly assess the applicability domain of cell membrane permeability predictors for cyclic peptides. Digital Discovery 3, 1761–1775.

Girolami, M., Calderhead, B., 2011. Riemann manifold langevin and hamiltonian monte carlo methods. Journal of the Royal Statistical Society Series B: Statistical Methodology 73, 123–214.

Gneiting, T., Raftery, A.E., 2007. Strictly proper scoring rules, prediction, and estimation. Journal of the American statistical Association 102, 359–378. URL: https://doi.org/10.1198/016214506000001437.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.

Graves, J., Byerly, J., Priego, E., Makkapati, N., Parish, S.V., Medellin, B., Berrondo, M., 2020. A review of deep learning methods for antibodies. Antibodies 9, 12.

Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A., 2012. A kernel two-sample test. The Journal of Machine Learning Research 13, 723–773.

Gruber, C., Schenk, P.O., Schierholz, M., Kreuter, F., Kauermann, G., 2023. Sources of Uncertainty in Machine Learning–A Statisticians' View. arXiv preprint arXiv:2305.16703 .

Gruber, S., Buettner, F., 2022. Better uncertainty calibration via proper scores for classification and beyond. Advances in Neural Information Processing Systems 35, 8618–8632.

Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q., 2017. On Calibration of Modern Neural Networks, in: International Conference on Machine Learning, PMLR. pp. 1321–1330.

Gupta, K., Rahimi, A., Ajanthan, T., Mensink, T., Sminchisescu, C., Hartley, R., 2021. Calibration of neural networks using splines, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=eQe8DEWNN2W.

Gustafsson, F.K., Danelljan, M., Schon, T.B., 2020. Evaluating scalable bayesian deep learning methods for robust computer vision, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp. 318–319.

Ha, D., Dai, A.M., Le, Q.V., 2017. Hypernetworks, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=rkpACe1lx.

Hanley, J.A., McNeil, B.J., 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology 143, 29–36.

Harrison, J., Willes, J., Snoek, J., 2024. Variational bayesian last layers. URL: https://arxiv.org/abs/2404.11599, arXiv:2404.11599.

Hartog, P.B.R., Svensson, E., Mervin, L., Genheden, S., Engkvist, O., Tetko, I.V., 2024. Registries in Machine Learning-Based Drug Discovery: A Shortcut to Code Reuse, in: International Workshop on AI in Drug Discovery, Springer. pp. 98–115. URL: https://doi.org/10.1007/978-3-031-72381-0_9.

Hastings, W.K., 1970. Monte carlo sampling methods using markov chains and their applications .

He, T., Heidemeyer, M., Ban, F., Cherkasov, A., Ester, M., 2017. Simboost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. Journal of cheminformatics 9, 24.

Heid, E., Greenman, K.P., Chung, Y., Li, S.C., Graff, D.E., Vermeire, F.H., Wu, H., Green, W.H., McGill, C.J., 2023a. Chemprop: a machine learning package for chemical property prediction. Journal of Chemical Information and Modeling 64, 9–17.

Heid, E., McGill, C.J., Vermeire, F.H., Green, W.H., 2023b. Characterizing Uncertainty in Machine Learning for Chemistry. J. Chem. Inf. Model. 63, 4012–4029. URL: https://doi.org/10.1021/acs.jcim.3c00373.

Hendrickson, A.D., Buehler, R.J., 1971. Proper scores for probability forecasters. The Annals of Mathematical Statistics , 1916–1921.

Hendrycks, D., Gimpel, K., 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks, in: International Conference

on Learning Representations. URL: https://openreview.net/forum?id=Hkg4TI9xl.

Hertzberg, R.P., Pope, A.J., 2000. High-Throughput Screening: New Technology for the 21st Century. Curr. Opin. Chem. Biol. 4, 445–451.

Heyndrickx, W., Mervin, L., Morawietz, T., Sturm, N., Friedrich, L., Zalewski, A., Pentina, A., Humbeck, L., Oldenhof, M., Niwayama, R., Schmidtke, P., Fechner, N., Simm, J., Arany, A., Drizard, N., Jabal, R., Afanasyeva, A., Loeb, R., Verma, S., Harnqvist, S., Holmes, M., Pejo, B., Telenczuk, M., Holway, N., Dieckmann, A., Rieke, N., Zumsande, F., Clevert, D.A., Krug, M., Luscombe, C., Green, D., Ertl, P., Antal, P., Marcus, D., Do Huu, N., Fuji, H., Pickett, S., Acs, G., Boniface, E., Beck, B., Sun, Y., Gohier, A., Rippmann, F., Engkvist, O., Göller, A.H., Moreau, Y., Galtier, M.N., Schuffenhauer, A., Ceulemans, H., 2023. MELLODDY: Cross-pharma Federated Learning at Unprecedented Scale Unlocks Benefits in QSAR without Compromising Proprietary Information. J. Chem. Inf. Model. 64, 2331–2344. URL: https://doi.org/10.1021/acs.jcim.3c00799.

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 .

Hirschfeld, L., Swanson, K., Yang, K., Barzilay, R., Coley, C.W., 2020. Uncertainty Quantification Using Neural Networks for Molecular Property Prediction. J. Chem. Inf. Model. 60, 3770–3780. URL: https://doi.org/10.1021/acs.jcim.0c00502.

Hochreiter, S., Klambauer, G., Rarey, M., 2018. Machine learning in drug discovery.

Hoffman, M.D., Gelman, A., et al., 2014. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. J. Mach. Learn. Res. 15, 1593–1623.

Hollander, M., Proschan, F., Sconing, J., 1987. Measuring Information in Right-Censored Models. Nav. Res. Logist. 34, 669–681. URL: https://doi.org/10.1002/1520-6750(198710)34:5<669::AID-NAV3220340507>3.0.CO;2-L.

Holliday, J.D., Hu, C., Willett, P., 2002. Grouping of coefficients for the calculation of inter-molecular similarity and dissimilarity using 2d fragment bit-strings. Combinatorial chemistry & high throughput screening 5, 155–166.

Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A., 2021. Meta-learning in neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence 44, 5149–5169.

Hsueh, H.T., Chou, R.T., Rai, U., Liyanage, W., Kim, Y.C., Appell, M.B., Pejavar, J., Leo, K.T., Davison, C., Kolodziejski, P., et al., 2023. Machine learning-driven multifunctional peptide engineering for sustained ocular drug delivery. Nature communications 14, 2509.

Hubschneider, C., Hutmacher, R., Zöllner, J.M., 2019. Calibrating Uncertainty Models for Steering Angle Estimation, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE. pp. 1511–1518.

Hüllermeier, E., Waegeman, W., 2021. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. Machine Learning 110, 457–506. URL: https://doi.org/10.1007/s10994-021-05946-3.

Hung, C., Gini, G., 2021. Qsar modeling without descriptors using graph convolutional neural networks: the case of mutagenicity prediction. Molecular diversity 25, 1283–1299.

Hüttel, F.B., Riis, C., Rodrigues, F., Pereira, F.C., 2024. Bayesian Active Learning for Censored Regression. arXiv preprint arXiv:2402.11973 .

Ioannides, C., 1996. Cytochromes P450: metabolic and toxicological aspects. Crc Press.

Izmailov, P., Vikram, S., Hoffman, M.D., Wilson, A.G.G., 2021. What are bayesian neural network posteriors really like?, in: International conference on machine learning, PMLR. pp. 4629–4640.

Janet, J.P., Duan, C., Yang, T., Nandy, A., Kulik, H.J., 2019. A Quantitative Uncertainty Metric Controls Error in Neural Network-Driven Chemical Discovery. Chem. Sci. 10, 7913–7922.

Ji, S., Marttinen, P., 2023. Patient outcome and zero-shot diagnosis prediction with hypernetwork-guided multitask learning, in: Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pp. 589–598.

Jiang, D., Wu, Z., Hsieh, C.Y., Chen, G., Liao, B., Wang, Z., Shen, C., Cao, D., Wu, J., Hou, T., 2021. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. Journal of cheminformatics 13, 1–23.

Jiang, S., Qin, S., Van Lehn, R.C., Balaprakash, P., Zavala, V.M., 2024. Uncertainty Quantification for Molecular Property Predictions with Graph Neural Architecture Search. Digital Discovery .

Juergens, M., Meinert, N., Bengs, V., Hüllermeier, E., Waegeman, W., 2024. Is Epistemic Uncertainty Faithfully Represented by Evidential Deep Learning Methods?, in: International Conference on Machine Learning, PMLR. pp. 22624–22642.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al., 2021. Highly accurate protein structure prediction with alphafold. nature 596, 583–589.

Kaelbling, L.P., Littman, M.L., Moore, A.W., 1996. Reinforcement learning: A survey. Journal of artificial intelligence research 4, 237–285.

Kalliokoski, T., Kramer, C., Vulpetti, A., Gedeck, P., 2013. Comparability of mixed ic50 data–a statistical analysis. PloS one 8, e61007.

Kamya, P., Ozerov, I.V., Pun, F.W., Tretina, K., Fokina, T., Chen, S., Naumov, V., Long, X., Lin, S., Korzinkin, M., et al., 2024. Pandaomics: an ai-driven platform for therapeutic target and biomarker discovery. Journal of chemical information and modeling 64, 3961–3969.

Keating, M.T., Sanguinetti, M.C., 1996. Molecular genetic insights into cardiovascular disease. Science 272, 681–685.

Kendall, A., Gal, Y., 2017. What uncertainties do we need in bayesian deep learning for computer vision? Advances in neural information processing systems 30.

Kim, H., Kim, E., Lee, I., Bae, B., Park, M., Nam, H., 2020. Artificial intelligence in drug discovery: A comprehensive review of data-driven and machine learning approaches. Biotechnology and Bioprocess Engineering 25, 895–930. URL: https://doi.org/10.1007/s12257-020-0049-y.

Kim, Q., Ko, J.H., Kim, S., Park, N., Jhe, W., 2021. Bayesian Neural Network with Pretrained Protein Embedding Enhances Prediction Accuracy of Drug-Protein Interaction. Bioinform. 37, 3428–3435. URL: https://doi.org/10.1093/bioinformatics/btab346.

Kingma, D.P., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Kirchhof, M., Kasneci, G., Kasneci, E., 2025. Reexamining the aleatoric and epistemic uncertainty dichotomy, in: ICLR Blogposts 2025. URL: https://iclr-blogposts.github.io/2025/blog/reexamining-the-aleatoric-and-epistemic-uncertainty-dichotomy/. https://iclr-blogposts.github.io/2025/blog/reexamining-the-aleatoric-and-epistemic-uncertainty-dichotomy/.

Klambauer, G., Hochreiter, S., Rarey, M., 2019. Machine learning in drug discovery.

Koh, P.W., Sagawa, S., Marklund, H., Xie, S.M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R.L., Gao, I., et al., 2021. Wilds: A benchmark of in-the-wild distribution shifts, in: International conference on machine learning, PMLR. pp. 5637–5664.

Kramer, C., Dahl, G., Tyrchan, C., Ulander, J., 2016. A comprehensive company database analysis of biological assay variability. Drug discovery today 21, 1213–1221.

Kramer, C., Kalliokoski, T., Gedeck, P., Vulpetti, A., 2012. The experimental uncertainty of heterogeneous public k i data. Journal of medicinal chemistry 55, 5165–5173.

Krogh, A., Hertz, J., 1991. A simple weight decay can improve generalization. Advances in neural information processing systems 4.

Krueger, D., Huang, C.W., Islam, R., Turner, R., Lacoste, A., Courville, A., 2017. Bayesian hypernetworks. arXiv preprint arXiv:1710.04759 .

Kuleshov, V., Fenner, N., Ermon, S., 2018. Accurate uncertainties for deep learning using calibrated regression, in: International conference on machine learning, PMLR. pp. 2796–2804.

Kulinski, S., Inouye, D.I., 2023. Towards Explaining Distribution Shifts, in: International Conference on Machine Learning, PMLR. pp. 17931–17952.

Kull, M., Flach, P., 2015. Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 68–85.

Kullback, S., Leibler, R.A., 1951. On information and sufficiency. The annals of mathematical statistics 22, 79–86.

Kumar, A., Liang, P.S., Ma, T., 2019. Verified uncertainty calibration. Advances in neural information processing systems 32.

Kumar, A., Sarawagi, S., Jain, U., 2018. Trainable calibration measures for neural networks from kernel mean embeddings, in: International Conference on Machine Learning, PMLR. pp. 2805–2814.

Lakshminarayanan, B., Pritzel, A., Blundell, C., 2017. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.

Landrum, G., 2006. RDKit: Open-Source Cheminformatics. URL: http://www.rdkit.org, doi:doi:10.5281/zenodo.6961488.

Lapinsh, M., Prusis, P., Gutcaits, A., Lundstedt, T., Wikberg, J.E., 2001. Development of proteo-chemometrics: a novel technology for the analysis of drug-receptor interactions. Biochimica et Biophysica Acta (BBA)-General Subjects 1525, 180–190.

Lee, S., Jo, J., Hwang, S.J., 2023. Exploring chemical space with score-based out-of-distribution generation, in: International Conference on Machine Learning, PMLR. pp. 18872–18892.

Lenselink, E.B., Ten Dijke, N., Bongers, B., Papadatos, G., Van Vlijmen, H.W., Kowalczyk, W., IJzerman, A.P., Van Westen, G.J., 2017. Beyond the Hype: Deep Neural Networks Outperform Established Methods Using a ChEMBL Bioactivity Benchmark Set. J. Cheminf. 9, 1–14. URL: https://doi.org/10.1186/s13321-017-0232-0.

Levi, D., Gispan, L., Giladi, N., Fetaya, E., 2022. Evaluating and Calibrating Uncertainty Prediction in Regression Tasks. Sensors 22, 5540. URL: https://doi.org/10.3390/s22155540.

Li, Y., Gal, Y., 2017. Dropout inference in bayesian neural networks with alpha-divergences, in: International conference on machine learning, PMLR. pp. 2052–2061.

Lodwich, A., Rangoni, Y., Breuel, T., 2009. Evaluation of robustness and performance of early stopping rules with multi layer perceptrons, in: 2009 international joint conference on Neural Networks, IEEE. pp. 1877–1884.

Madhukar, N.S., Khade, P.K., Huang, L., Gayvert, K., Galletti, G., Stogniew, M., Allen, J.E., Giannakakou, P., Elemento, O., 2019. A bayesian machine learning approach for drug target identification using diverse data types. Nature communications 10, 5221.

Maggiora, G.M., 2006. On Outliers and Activity Cliffs Why QSAR Often Disappoints. J. Chem. Inf. Model. 46, 1535–1535. URL: https://doi.org/10.1021/ci060117s.

Malinin, A., Gales, M., 2018. Predictive uncertainty estimation via prior networks. Advances in neural information processing systems 31.

Mann, H.B., Whitney, D.R., 1947. On a Test of Whether One of Two Random Variables is Stochastically Larger Than the Other. The Annals of Mathematical Statistics , 50–60.

Masimirembwa, C.M., Thompson, R., Andersson, T.B., 2001. In vitro high throughput screening of compounds for favorable metabolic properties in drug discovery. Combinatorial chemistry & high throughput screening 4, 245–263.

Mayr, A., Klambauer, G., Unterthiner, T., Steijaert, M., Wegner, J.K., Ceulemans, H., Clevert, D.A., Hochreiter, S., 2018. Large-Scale Comparison of Machine Learning Methods for Drug Target Prediction on ChEMBL. Chem. Sci. 9, 5441–5451. URL: https://doi.org/10.1039/C8SC00148K.

Mehrtens, H.A., Kurz, A., Bucher, T.C., Brinker, T.J., 2023. Benchmarking common uncertainty estimation methods with histopathological images under domain shift and label noise. Medical image analysis 89, 102914.

Mendez, D., Gaulton, A., Bento, A.P., Chambers, J., De Veij, M., Félix, E., Magarinos, M.P., Mosquera, J.F., Mutowo, P., Nowotka, M., Gordillo-Maranon, M., Hunter, F., Junco, L., Mugumbate, G., Rodriguez-Lopez, M., Atkinson, F., Bosc, N., Radoux, C.J., Segura-Cabrera, A., Hersey, A., Leach, A.R., 2018. ChEMBL: towards direct deposition of bioassay data. Nucleic Acids Research 47, D930–D940. URL: https://doi.org/10.1093/nar/gky1075.

Mervin, L.H., Afzal, A.M., Engkvist, O., Bender, A., 2020. Comparison of scaling methods to obtain calibrated probabilities of activity for protein–ligand predictions. Journal of Chemical Information and Modeling 60, 4546–4559.

Mervin, L.H., Johansson, S., Semenova, E., Giblin, K.A., Engkvist, O., 2021a. Uncertainty Quantification in Drug Design. Drug Discovery Today 26, 474–489. URL: https://doi.org/10.1016/j.drudis.2020.11.027.

Mervin, L.H., Trapotsi, M.A., Afzal, A.M., Barrett, I.P., Bender, A., Engkvist, O., 2021b. Probabilistic random forest improves bioactivity predictions close to the classification threshold by taking into account experimental uncertainty. Journal of Cheminformatics 13, 62.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. The journal of chemical physics 21, 1087–1092.

Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., Lucic, M., 2021. Revisiting the calibration of modern neural networks. Advances in Neural Information Processing Systems 34, 15682–15694.

Morgan, H.L., 1965. The Generation of a Unique Machine Description for Chemical Structures – A Technique Developed at Chemical Abstracts Service. J. Chem. Doc. 5, 107–113. URL: https://doi.org/10.1021/c160017a018.

Mucsányi, B., Kirchhof, M., Oh, S.J., 2024. Benchmarking uncertainty disentanglement: Specialized uncertainties for specialized tasks. Advances in neural information processing systems 37, 50972–51038.

Mukhoti, J., Kirsch, A., Van Amersfoort, J., Torr, P.H., Gal, Y., 2023. Deep deterministic uncertainty: A new simple baseline, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 24384–24394.

Muratov, E.N., Bajorath, J., Sheridan, R.P., Tetko, I.V., Filimonov, D., Poroikov, V., Oprea, T.I., Baskin, I.I., Varnek, A., Roitberg, A., et al., 2020. Qsar without borders. Chemical Society Reviews 49, 3525–3564.

Nachmani, E., Wolf, L., 2020. Molecule property prediction and classification with graph hypernetworks. arXiv preprint arXiv:2002.00240 .

Naeem, S., Ali, A., Anam, S., Ahmed, M.M., 2023. An unsupervised machine learning algorithms: Comprehensive review. International Journal of Computing and Digital Systems .

Naeini, M.P., Cooper, G., Hauskrecht, M., 2015. Obtaining well calibrated probabilities using bayesian binning, in: Proceedings of the AAAI conference on artificial intelligence.

Neal, R.M., et al., 2011. Mcmc using hamiltonian dynamics. Handbook of markov chain monte carlo 2, 2.

Nix, D.A., Weigend, A.S., 1994. Estimating the Mean and Variance of the Target Probability Distribution, in: Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94), IEEE. pp. 55–60.

Nixon, J., Dusenberry, M.W., Zhang, L., Jerfel, G., Tran, D., 2019. Measuring calibration in deep learning., in: CVPR workshops.

Nowlan, S.J., Hinton, G.E., 2018. Simplifying neural networks by soft weight sharing, in: The mathematics of generalization. CRC Press, pp. 373–394.

Ober, S.W., Rasmussen, C.E., 2019. Benchmarking the neural linear model for regression. URL: https://arxiv.org/abs/1912.08416, arXiv:1912.08416.

Ouyang, L., Key, A., 2021. Maximum mean discrepancy for generalization in the presence of distribution and missingness shift. arXiv preprint arXiv:2111.10344 .

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., Snoek, J., 2019. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.

Ovcharov, E.Y., 2018. Proper scoring rules and Bregman divergence. Bernoulli 24, 53 – 79. URL: https://doi.org/10.3150/16-BEJ857.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.

Pearce, T., Jeong, J.H., Jia, Y., Zhu, J., 2022. Censored Quantile Regression Neural Networks for Distribution-Free Survival Analysis, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 7450–7461.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, 2825–2830. URL: http://jmlr.org/papers/v12/pedregosa11a.html.

Platt, J., 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Adv. Large Margin Classif. 10.

Polikar, R., 2006. Ensemble based systems in decision making. IEEE Circuits and systems magazine 6, 21–45.

Popordanoska, T., Gruber, S.G., Tiulpin, A., Buettner, F., Blaschko, M.B., 2024. Consistent and asymptotically unbiased estimation of proper calibration errors, in: International Conference on Artificial Intelligence and Statistics, PMLR. pp. 3466–3474.

Powell, J.L., 1986. Censored Regression Quantiles. J. Econom. 32, 143–155. URL: https://doi.org/10.1016/0304-4076(86)90016-3.

Rabanser, S., Günnemann, S., Lipton, Z., 2019. Failing loudly: An empirical study of methods for detecting dataset shift. Advances in Neural Information Processing Systems 32.

Rahaman, R., Thiery, A.H., 2021. Uncertainty Quantification and Deep Ensembles, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 20063–20075.

Ramsundar, B., Liu, B., Wu, Z., Verras, A., Tudor, M., Sheridan, R.P., Pande, V., 2017. Is multitask deep learning practical for pharma? Journal of chemical information and modeling 57, 2068–2076.

Rasmussen, M.H., Duan, C., Kulik, H.J., Jensen, J.H., 2023. Uncertain of Uncertainties? A comparison of Uncertainty Quantification Metrics for Chemical Data Sets. J. Cheminf. 15, 121. URL: https://doi.org/10.1186/s13321-023-00790-0.

Reis, I., Baron, D., Shahaf, S., 2018. Probabilistic random forest: A machine learning algorithm for noisy data sets. The Astronomical Journal 157, 16.

Renz, P., Hochreiter, S., Klambauer, G., 2019. Uncertainty estimation methods to support decision-making in early phases of drug discovery, in: Workshop on Safety and Robustness in Decisionmaking.

Rodríguez-Pérez, R., Trunzer, M., Schneider, N., Faller, B., Gerebtzoff, G., 2022. Multispecies machine learning predictions of in vitro intrinsic clearance with uncertainty quantification analyses. Molecular Pharmaceutics 20, 383–394.

Rogers, D., Hahn, M., 2010. Extended-connectivity fingerprints. Journal of chemical information and modeling 50, 742–754.

Roth, J.P., Bajorath, J., 2024. Relationship between prediction accuracy and uncertainty in compound potency prediction using deep neural networks and control models. Scientific Reports 14, 6536.

Ryu, S., Kwon, Y., Kim, W.Y., 2019. Uncertainty quantification of molecular property prediction with bayesian neural networks. arXiv preprint arXiv:1903.08375 .

Sadybekov, A.V., Katritch, V., 2023. Computational approaches streamlining drug discovery. Nature 616, 673–685.

Sale, Y., Bengs, V., Caprio, M., Hüllermeier, E., 2024. Second-Order Uncertainty Quantification: A Distance-Based Approach, in: International Conference on Machine Learning, PMLR.

Sarkar, C., Das, B., Rawat, V.S., Wahlang, J.B., Nongpiur, A., Tiewsoh, I., Lyngdoh, N.M., Das, D., Bidarolli, M., Sony, H.T., 2023. Artificial intelligence and machine learning technology driven modern drug discovery and development. International Journal of Molecular Sciences 24, 2026.

Sato, K., Hamada, M., 2023. Recent trends in rna informatics: a review of machine learning and deep learning for rna secondary structure prediction and rna drug discovery. Briefings in Bioinformatics 24, bbad186.

Schimunek, J., Friedrich, L., Kuhn, D., Rippmann, F., Hochreiter, S., Klambauer, G., 2021. A generalized framework for embedding-based few-shot learning methods in drug discovery, in: ELLIS Machine Learning for Molecules Workshop.

Schlander, M., Hernandez-Villafuerte, K., Cheng, C.Y., Mestre-Ferrandiz, J., Baumann, M., 2021. How much does it cost to research and develop a new drug? a systematic review and assessment. PharmacoEconomics 39, 1–27. URL: https://doi.org/10.1007/s40273-021-01065-y.

Schweighofer, K., Aichberger, L., Ielanskyi, M., Hochreiter, S., 2024. On information-theoretic measures of predictive uncertainty. arXiv preprint arXiv:2410.10786 .

Schweighofer, K., Aichberger, L., Ielanskyi, M., Klambauer, G., Hochreiter, S., 2023. Quantification of Uncertainty with Adversarial Models, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.

Shah, P., Jogani, V., Bagchi, T., Misra, A., 2006. Role of caco-2 cell monolayers in prediction of intestinal drug absorption. Biotechnology progress 22, 186–198.

Sheinkman, A., Wade, S., 2025. Understanding the trade-offs in accuracy and uncertainty quantification: Architecture and inference choices in bayesian neural networks, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 129–146.

Sheridan, R.P., 2012. Three Useful Dimensions for Domain Applicability in QSAR Models Using Random Forest. J. Chem. Inf. Model. 52, 814–823. URL: https://doi.org/10.1021/ci300004n.

Sheridan, R.P., 2013. Time-Split Cross-Validation as a Method for Estimating the Goodness of Prospective Prediction. J. Chem. Inf. Model. 53, 783–790. URL: https://doi.org/10.1021/ci400084k.

Sheridan, R.P., Feuston, B.P., Maiorov, V.N., Kearsley, S.K., 2004. Similarity to Molecules in the Training Set is a Good Discriminator for Prediction Accuracy in QSAR. J. Chem. Inf. Comput. Sci. 44, 1912–1928. URL: https://doi.org/10.1021/ci049782w.

Simm, J., Klambauer, G., Arany, A., Steijaert, M., Wegner, J.K., Gustin, E., Chupakhin, V., Chong, Y.T., Vialard, J., Buijnsters, P., Velter, I., Vapirev, A., Singh, S., Carpenter, A.E., Wuyts, R., Hochreiter, S., Moreau, Y., Ceulemans, H., 2018. Repurposing high-throughput image assays enables biological activity prediction for drug discovery. Cell Chemical Biology 25, 611–618.e3. doi:doi:10.1016/j.chembiol.2018.01.015.

Simões, R.S., Maltarollo, V.G., Oliveira, P.R., Honorio, K.M., 2018. Transfer and multi-task learning in qsar modeling: advances and challenges. Frontiers in pharmacology 9, 74.

Smith, L., Gal, Y., 2018. Understanding measures of uncertainty for adversarial example detection. arXiv preprint arXiv:1803.08533 .

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., Adams, R., 2015. Scalable bayesian optimization using deep neural networks, in: International conference on machine learning, PMLR. pp. 2171–2180.

Sommer, E., Wimmer, L., Papamarkou, T., Bothmann, L., Bischl, B., Rügamer, D., 2024. Connecting the dots: Is mode-connectedness the key to feasible sample-based inference in bayesian neural networks? URL: https://arxiv.org/abs/2402.01484, arXiv:2402.01484.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15, 1929–1958.

Stan, D.T., 2024. Stan reference manual. URL: https://mc-stan.org/docs/2_37/reference-manual/.

Stokes, J.M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N.M., MacNair, C.R., French, S., Carfrae, L.A., Bloom-Ackermann, Z., et al., 2020. A deep learning approach to antibiotic discovery. Cell 180, 688–702.

Stoyanova, R., Katzberger, P.M., Komissarov, L., Khadhraoui, A., Sach-Peltason, L., Groebke Zbinden, K., Schindler, T., Manevski, N., 2023. Computational predictions of nonclinical pharmacokinetics at the drug design stage. Journal of Chemical Information and Modeling 63, 442–458.

Sutton, R.S., Barto, A.G., et al., 1998. Reinforcement learning: An introduction. volume 1. MIT press Cambridge.

Svensson, E., Friesacher, H.R., Arany, A., Mervin, L., Engkvist, O., 2024a. Temporal Evaluation of Uncertainty Quantification Under Distribution Shift, in: International Workshop on AI in Drug Discovery, Springer. pp. 132–148. URL: https://doi.org/10.1007/978-3-031-72381-0_11.

Svensson, E., Friesacher, H.R., Winiwarter, S., Mervin, L., Arany, A., Engkvist, O., 2025. Enhancing uncertainty quantification in drug discovery with censored regression labels. Artificial Intelligence in the Life Sciences , 100128.

Svensson, E., Hoedt, P.J., Hochreiter, S., Klambauer, G., 2024b. HyperPCM: Robust Task-Conditioned Modeling of Drug–Target Interactions. J. Chem. Inf. Model. 64, 2539–2553. URL: https://doi.org/10.1021/acs.jcim.3c01417.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2013. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 .

Tetko, I.V., Sushko, I., Pandey, A.K., Zhu, H., Tropsha, A., Papa, E., Oberg, T., Todeschini, R., Fourches, D., Varnek, A., 2008. Critical Assessment of QSAR Models of Environmental Toxicity Against Tetrahymena Pyriformis: Focusing on Applicability Domain and Overfitting by Variable Selection. J. Chem. Inf. Model. 48, 1733–1746. URL: https://doi.org/10.1021/ci800151m.

Tobin, J., 1958. Estimation of Relationships for Limited Dependent Variables. Econometrica: Journal of the Econometric Society 26, 24–36. URL: https://doi.org/10.2307/1907382.

Toccaceli, P., Nouretdinov, I., Luo, Z., Vovk, V., Carlsson, L., Gammerman, A., 2016. Excape wp1-probabilistic prediction.

Valeur, E., Jimonet, P., 2018. New modalities, technologies, and partnerships in probe and lead generation: enabling a mode-of-action centric paradigm. Journal of Medicinal Chemistry 61, 9004–9029.

Van De Waterbeemd, H., Gifford, E., 2003. Admet in silico modelling: towards prediction paradise? Nature reviews Drug discovery 2, 192–204.

Van Tilborg, D., Alenicheva, A., Grisoni, F., 2022. Exposing the Limitations of Molecular Machine Learning with Activity Cliffs. J. Chem. Inf. Model. 62, 5938–5951. URL: https://doi.org/10.1021/acs.jcim.2c01073.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., Bürkner, P.C., 2021. Rank-normalization, folding, and localization: An improved r ÌĆ for assessing convergence of mcmc (with discussion). Bayesian analysis 16, 667–718.

Vergara, D., Hernández, S., Valdenegro-Toro, M., Jorquera, F., 2018. Improving predictive uncertainty estimation using dropout–hamiltonian monte carlo. arXiv preprint arXiv:1805.04756 .

Vovk, V., Petej, I., 2014. Venn-abers predictors. doi:doi:10.48550/arXiv.1211.0025, arXiv:1211.0025.

Wang, C., 2025. Calibration in deep learning: A survey of the state-of-the-art. URL: https://arxiv.org/abs/2308.01222, arXiv:2308.01222.

Wang, D., Wu, Z., Shen, C., Bao, L., Luo, H., Wang, Z., Yao, H., Kong, D.X., Luo, C., Hou, T., 2023a. Learning with uncertainty to accelerate the discovery of histone lysine-specific demethylase 1a (kdm1a/lsd1) inhibitors. Briefings in Bioinformatics 24, bbac592.

Wang, D., Yu, J., Chen, L., Li, X., Jiang, H., Chen, K., Zheng, M., Luo, X., 2021. A Hybrid Framework for Improving Uncertainty Quantification in Deep Learning-Based QSAR Regression Modeling. J. Cheminf. 13, 1–17. URL: https://doi.org/10.1186/s13321-021-00551-x.

Wang, R., Liu, Z., Gong, J., Zhou, Q., Guan, X., Ge, G., 2023b. An Uncertainty-Guided Deep Learning Method Facilitates Rapid Screening of CYP3A4 Inhibitors. J. Chem. Inf. Model. 63, 7699–7710. URL: https://doi.org/10.1021/acs.jcim.3c01241.

Waring, M.J., 2010. Lipophilicity in drug discovery. Expert Opinion on Drug Discovery 5, 235–248.

Watson, J., Lin, J.A., Klink, P., Pajarinen, J., Peters, J., 2021. Latent derivative bayesian last layer networks, in: International Conference on Artificial Intelligence and Statistics, PMLR. pp. 1198–1206.

Weininger, D., 1988. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. Journal of chemical information and computer sciences 28, 31–36.

Widmann, D., Lindsten, F., Zachariah, D., 2019. Calibration tests in multi-class classification: A unifying framework. Advances in neural information processing systems 32.

Wilson, A.G., Izmailov, P., 2020. Bayesian deep learning and a probabilistic perspective of generalization. Advances in neural information processing systems 33, 4697–4708.

Winter, R., Montanari, F., Noé, F., Clevert, D.A., 2019. Learning Continuous and Data-Driven Molecular Descriptors by Translating Equivalent Chemical Representations. Chem. Sci. 10, 1692–1701. URL: https://doi.org/10.1039/C8SC04175J.

Wouters, O.J., McKee, M., Luyten, J., 2020. Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018. JAMA 323, 844–853. URL: https://doi.org/10.1001/jama.2020.1166.

Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y., Zuo, W., 2017. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2272–2281.

Yang, C.I., Li, Y.P., 2023. Explainable Uncertainty Quantifications for Deep Learning-Based Molecular Property Prediction. J. Cheminf. 15, 13. URL: https://doi.org/10.1186/s13321-023-00682-3.

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., 2019. Analyzing Learned Molecular Representations for Property Prediction. J. Chem. Inf. Model. 59, 3370–3388. URL: https://doi.org/10.1021/acs.jcim.9b00237.

Yang, W., Gadgil, P., Krishnamurthy, V.R., Landis, M., Mallick, P., Patel, D., Patel, P.J., Reid, D.L., Sanchez-Felix, M., 2020. The evolving druggability and developability space: chemically modified new modalities and emerging small molecules. The AAPS journal 22, 21.

Yin, T., Panapitiya, G., Coda, E.D., Saldanha, E.G., 2023. Evaluating Uncertainty-Based Active Learning for Accelerating the Generalization of Molecular Property Prediction. J. Cheminf. 15, 105. URL: https://doi.org/10.1186/s13321-023-00753-5.

Ying, X., 2019. An overview of overfitting and its solutions, in: Journal of physics: Conference series, IOP Publishing. p. 022022.

Zadrozny, B., Elkan, C., 2001. Obtaining Calibrated Probability Estimates from Decision Trees and Naive Bayesian Classifiers, in: International Conference on Machine Learning, PMLR. pp. 609–616.

Zadrozny, B., Elkan, C., 2002. Transforming classifier scores into accurate multiclass probability estimates, in: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 694–699.

Zdrazil, B., Felix, E., Hunter, F., Manners, E.J., Blackshaw, J., Corbett, S., de Veij, M., Ioannidis, H., Lopez, D.M., Mosquera, J.F., et al., 2024. The chembl database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. Nucleic acids research 52, D1180–D1192.

Zeng, X., Wang, H., Zhao, L., Cheng, Y., Zhou, D., Shi, S., 2025. Uncertainty quantification and temperature scaling calibration for protein-rna binding site prediction. Journal of Chemical Information and Modeling .

Zhu, Y., Zhuang, F., Wang, J., Ke, G., Chen, J., Bian, J., Xiong, H., He, Q., 2020. Deep subdomain adaptation network for image classification. IEEE transactions on neural networks and learning systems 32, 1713–1722.

Funding

This project was funded by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie Actions grant agreement 'Advanced machine learning for Innovative Drug Discovery (AIDD)' No. 956832. Hannah Rosa Friesacher, Adam Arany and Yves Moreau are affiliated to Leuven.AI and received funding from the Flemish Government (AI Research Program). Computational resources and services used in this work were partly provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government-department EWI.

Statement on the use of Generative AI

The text, code, and images in this thesis are my own (unless otherwise specified). Generative AI has only been used in accordance with the KU Leuven guidelines and appropriate references have been added. I have reviewed and edited the content as needed and I take full responsibility for the content of the thesis.

Curriculum

Hannah Rosa Friesacher holds a BSc in Biology and an MSc in Drug Discovery and Development from the University of Vienna, with a specialization in computational tools for drug research. Following her studies, she became part of the 'Advanced Machine Learning for Innovative Drug Discovery' (AIDD) consortium to pursue a PhD, during which she joined the Yves Moreau research group at the Department of Electrical Engineering (ESAT) at KU Leuven, Belgium, and the Molecular AI group at AstraZeneca in Gothenburg, Sweden. Her research was conducted under the supervision of Adam Arany, Prof. Ola Engkvist, and Prof. Yves Moreau.

List of Publications

- Friesacher, H.R., Engkvist, O., Mervin, L., Moreau, Y., Arany, A., 2025a. Achieving well-informed decision-making in drug discovery: a comprehensive calibration study using neural network-based structure-activity models. Journal of Cheminformatics 17, 29 (Friesacher et al., 2025a)

- Friesacher, H.R., Svensson, E., Arany, A., Mervin, L., Engkvist, O., 2024. Temporal evaluation of probability calibration with experimental errors, in: International Workshop on AI in Drug Discovery, Springer. pp. 13âĂŞ20. (Friesacher et al., 2024)

- Friesacher, H.R., Svensson, E., Winiwarter, S., Mervin, L., Arany, A., Engkvist, O., 2025b. Temporal Distribution shift in real-world pharmaceutical data: Implications for uncertainty quantification in qsar models. Artificial Intelligence in the Life Sciences 8, 100132. (Friesacher et al., 2025b)

- Svensson, E., Friesacher, H.R., Arany, A., Mervin, L., Engkvist, O., 2024a. Temporal Evaluation of Uncertainty Quantification Under Distribution Shift, in: International Workshop on AI in Drug Discovery, Springer. pp. 132âĂŞ148. (Svensson et al., 2024a)

- Svensson, E., Friesacher, H.R., Winiwarter, S., Mervin, L., Arany, A., Engkvist, O., 2025. Enhancing uncertainty quantification in drug discovery with censored regression labels. Artificial Intelligence in the Life Sciences , 100128 (Svensson et al., 2025)

- Meyer, M. J., Neumann, V. E., Friesacher, H. R., Zdrazil, B., BrockmoÌĹller, J., Tzvetkov, M. V. (2019). Opioids as substrates and inhibitors of the genetically highly variable organic cation transporter OCT1. Journal of medicinal chemistry, 62(21), 9890-9905.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
Celestijnenlaan 200A box 2402
B-3001 Leuven

