

MARKUS KÄNGSEPP

Uncertainty calibration and evaluation
in deep neural networks



MARKUS KÄNGSEPP

Uncertainty calibration and evaluation
in deep neural networks



UNIVERSITY OF TARTU

Press

Institute of Computer Science, Faculty of Science and Technology, University of Tartu, Estonia.

Dissertation has been accepted for the commencement of the degree of Doctor of Philosophy (PhD) in Computer Science on September 1, 2026 by the Council of the Institute of Computer Science, University of Tartu.

Supervisor

Prof. Meelis Kull
University of Tartu
Estonia

Opponents

Prof. Ulf Brefeld
Leuphana University of Lüneburg
Germany

Prof. Fredrik Lindsten
Linköping University
Sweden

The public defense will take place on November 17, 2026 at 10:15 in Narva Rd. 18-2046.

The publication of this dissertation was financed by the Institute of Computer Science, University of Tartu.

ISSN 2613-5906 (print)

ISSN 2806-2345 (pdf)

ISBN 978-9908-57-376-2 (print)

ISBN 978-9908-57-377-9 (pdf)

Copyright © 2026 by Markus Kängsepp

University of Tartu Press

<http://www.tyk.ee/>

To family and friends, and to the ability to detect overconfidence

ABSTRACT

Machine learning (ML) increasingly impacts many aspects of everyday life. Applications such as translation, chatbots, autonomous vehicles, fraud detection, loan decisions, and hiring are already influenced by ML, and the role of ML is expected to grow further. ML algorithms learn from data to perform specific tasks, such as automatically responding to customer queries. Modern models are larger, more complex, and more capable than ever before. As a result, they are increasingly deployed in safety-critical settings, where the consequences of errors can be significant. Therefore, it is essential to interpret model outputs and to understand when a model is uncertain about its predictions in order to avoid costly mistakes.

Machine learning models can output probabilities associated with their predictions, which can be used for decision-making. For example, the probability of rain in the next hour can guide decisions, such as increasing energy production to offset reduced solar output. In safety-critical applications, such as autonomous driving and medical diagnosis, predicted probabilities may determine whether an action is taken, withheld, or referred for human review. Poorly estimated probabilities may therefore lead to unsafe or costly decisions.

The reliability of these probability estimates is characterised by calibration. Probabilities are more reliable when they are well-calibrated, meaning that predicted probabilities correspond to observed frequencies. For instance, among all cases where a model predicts an 80% probability of rain, approximately 80 out of 100 should indeed result in rain. In practice, predicted probabilities are often not well-calibrated and therefore need to be evaluated and, when necessary, adjusted. Reliability diagrams and metrics such as the estimated calibration error (ECE) are commonly used to assess calibration quality. When a model is poorly calibrated, post-hoc calibration methods, including temperature scaling for deep neural networks, can be applied to improve probability estimates.

In this work, we examine several aspects of uncertainty calibration. First, we propose a new method for multiclass calibration, along with a novel regularisation technique specifically designed for calibration methods with high number of parameters. Second, we introduce a new perspective on calibration evaluation, which we define as the fit-on-test view. We show that calibration evaluation inherently involves fitting a calibration map, that is, a mapping from uncalibrated to calibrated probabilities, and propose a new evaluation method. Third, we propose a framework for representing uncertainty in perception systems, with a focus on autonomous driving. We further show how to generate the uncertainty estimates defined by the framework, demonstrate that these are not well-calibrated by default, and present methods for their calibration.

CONTENTS

List of original publications	14
1. Introduction	16
2. Background and preliminaries	19
2.1. Notation	19
2.2. Uncertainty of probabilistic classifier	20
2.3. Calibration of probabilistic classifier	20
2.3.1. Calibration methods	22
2.3.2. Post-hoc calibrators	22
2.3.3. Binary calibration methods	25
2.3.4. Multiclass calibration methods	26
2.3.5. Training-time calibrators	28
2.4. Evaluation of calibration	30
2.4.1. Proper scoring rules	30
2.4.2. Estimating calibration error	31
2.4.3. Other evaluation measures	33
2.4.4. Visualisation and interpretability	33
3. Multiclass calibration method - Dirichlet calibration (Publication I)	38
3.1. Introduction	38
3.2. Motivation	38
3.3. Methodology	38
3.3.1. Interpretability	40
3.3.2. Comparison to other methods	41
3.3.3. Implementation	42
3.3.4. Off-diagonal and intercept regularisation	43
3.4. Experiments	44
3.5. Summary and limitations	47
4. Rethinking evaluation of calibration: fit-on-test (Publication II)	48
4.1. Introduction	48
4.2. Motivation	48
4.3. Fit-on-test	49
4.4. Piecewise linear method for estimating the calibration map	51
4.5. Experiments and results	54
4.6. Summary and limitations	58
5. Combining uncertainty information of semantic segmentation for autonomous perception systems (Publication III)	59
5.1. Motivation	59
5.2. Uncertainty-aware perception system	60

5.3. Methods: extracting uncertainty estimates from the BEV semantic segmentation	64
5.3.1. Probabilistic object centre location	64
5.3.2. Probabilistic object presence	65
5.3.3. Probabilities of region occupancy queries	67
5.4. Experiments	67
5.4.1. Results on probabilistic grid-based view	68
5.4.2. Results on detected and undetected objects	69
5.4.3. Downstream task	71
5.5. Summary and limitations	72
6. Conclusion	74
6.1. Trends and opportunities for future research	76
Bibliography	78
Acknowledgements	84
Sisukokkuvõte (Summary in Estonian)	85
Publications	89
Beyond temperature scaling: obtaining well-calibrated multiclass probabilities with Dirichlet calibration	92
On the usefulness of the fit-on-test view on evaluating calibration of classifiers	122
Calibrating perception uncertainty for autonomous driving	200
Curriculum Vitae	227
Elulookirjeldus (Curriculum Vitae in Estonian)	229

LIST OF FIGURES

1. Example of true calibration map for a binary classifier.	23
2. Calibration pipeline for post-hoc calibrators.	24
3. Example of reliability diagram variants: (a) equal-width binning showing only average labels $\bar{y}_k$, (b) equal-width binning with red dots indicating bin centres ($\bar{p}_k$) and (c) equal-width binning with an added histogram of sample frequencies and vertical lines indicating calibration gaps. Adapted from Publication II.	35
4. Example of a reliability diagram with equal-size binning, each bin has the same number of elements. The figure on the right (b) is zoomed in to show the last 3 bins.	35
5. Example of reliability diagram with deviations instead of showing average accuracy and average probability for each bin (Vaicenavicius et al., 2019). The blue bars indicate the sample frequencies in each bin.	36
6. Example of a probability simplex for visualising 3-class calibration. The arrows mark the change of probabilities after calibration. Adapted from Publication I.	36
7. Reliability diagrams for WideNet32 (Zagoruyko and Komodakis, 2016) on CIFAR-10. Subfigure (a) shows the confidence reliability diagram prior to calibration, while (b) presents the same after temperature scaling. Panels (c) and (d) display classwise reliability diagrams for class 2 after temperature scaling and Dirichlet calibration, respectively. Adapted from Publication I.	39
8. Interpretation of a Dirichlet calibration mapping: (a) the coefficient matrix in the canonical parameterisation for SVHN_convnet; and (b) the resulting changes in the confusion matrix after applying the calibration map. The matrix entries describe relative contributions to the calibrated logits and should not be interpreted as direct marginal changes in the calibrated probabilities. Adapted from Publication I.	41
9. Implementation example of Dirichlet calibration method using neural networks.	43
10. Pipeline of fit-on-test estimation of calibration error. Adapted from Publication II.	50
11. A synthetic example demonstrating ECE as a fit-on-test estimator of calibration error. (a) Reliability diagram for a test set of six instances, yielding $ECE = 0.133$; (b) piecewise-constant estimator with $CE = 0.167$; (c) piecewise slope-1 estimator recovering the same $ECE = 0.133$ as in (a) while visualising per-instance errors; and (d) the proposed <i>reliability diagram with diagonal filling</i> , combining (a) and (c). Adapted from Publication II.	51

12. Different reliability diagrams on synthetic data, with the number of bins or segments selected via cross-validation. Subfigure (a) shows a reliability diagram with diagonal filling using 14 bins; (b) a piecewise linear reliability diagram with 3 segments; and (c) a piecewise linear reliability diagram in logit–logit space with 2 segments. Adapted from Publication II.	52
13. Motivation for piecewise linear in logit-logit space (PL3). Results are compared in probability space (left column) and logit–logit space (right column) for the ResNet110 (He et al., 2015) model on the CIFAR-5m dataset for the “cats vs. rest” task. Adapted from Publication II.	55
14. Dependence of grid-cell correlation on Euclidean distance in metres. For each value of D , correlations are evaluated using pairs of cells with distances greater than $D - 1$ and at most D . Results are derived from bird’s-eye-view (BEV) semantic-segmentation outputs of the FIERY model. Adapted from Publication III.	61
15. Overview of the proposed uncertainty framework, illustrating the uncertainties extracted from BEV semantic segmentation. The blue dashed outline highlights the scope of this work. Adapted from Publication III.	62
16. Bird’s-eye-view (BEV) perception uncertainty based on the FIERY model. Subfigures (a) and (b) show FIERY semantic segmentation in log-scale (natural logarithm) for improved visibility, clipped at -12 and -5, respectively. Subfigure (c) illustrates the proposed additional information on object locations, represented as ellipsoids, along with an example region occupancy query (ROQ) region indicated by the dashed red outline. The blue rectangle at (100,100) denotes the ego-vehicle. Note that subfigure (c) is zoomed in. Adapted from Publication III.	63
17. Reliability diagrams illustrating pixel-wise semantic segmentation calibration. Subfigure (a) corresponds to the uncalibrated model, and subfigure (b) shows pixel-wise calibration with equal-width binning into 15 bins. Adapted from Publication III.	69
18. Regression calibration plots (Kuleshov et al., 2018) illustrating object location probabilities. Subfigure (a) corresponds to direction error and subfigure (b) to distance. Results are shown for uncalibrated (uncal), pixel-wise calibrated (pw-cal), and object-wise calibrated (obj-cal) settings. The black dashed line denotes perfect calibration. Adapted from Publication III.	70

19. Reliability diagrams illustrating object presence probabilities. Subfigures (a), (b), and (c) correspond to uncalibrated, pixel-wise calibrated, and object-wise calibrated cases, respectively. The histograms beneath the diagrams display the number of elements in each bin, with a logarithmic y-axis for better comparison. Adapted from Publication III.	71
---	----

LIST OF TABLES

1. Comparison of calibration methods.	28
2. Classwise-ECE rankings of non-neural models, computed over 21 UCI datasets. Adapted from Publication I.	45
3. Log-loss (NLL) rankings of non-neural models, computed over 21 UCI datasets. Adapted from Publication I.	45
4. Comparison of calibration methods using classwise-ECE scores and rankings across deep neural network experiments. Adapted from Publication I.	46
5. Comparison of calibration methods using log-loss (NLL) scores and rankings across deep neural network experiments. Adapted from Publication I.	46
6. Evaluation of post-hoc calibrators based on $ \hat{c} - c^* (\times 10^3)$ in approximating the true calibration map for DenseNet40, ResNet110, and WideNet32 models on CIFAR-5m. Each calibrator is trained on 5000 instances, with results averaged over the three classifiers. Adopted from Publication II.	56
7. Evaluation of calibration estimators for reliability diagrams based on $ \hat{c} - c^* (\times 10^3)$ on CIFAR-5m. Unlike Table 6, this comparison is performed on predictions that have already undergone post-hoc calibration. Adopted from Publication II.	57
8. Evaluation of calibration estimators for true calibration error, measured by $ \text{ECE} - \text{CE} (\times 10^3)$, together with the ranking of six calibrators according to rankcorrel(ECE) on CIFAR-5m. Adapted from Publication II.	57
9. Evaluation of trajectory estimation performance for three approaches: uncalibrated (uncal), pixel-wise calibrated (pw-cal), and object-level calibration with region occupancy query probabilities (ROQ (obj-cal)). Best results are indicated in bold. Adapted from Publication III.	72

LIST OF ABBREVIATIONS

Acronyms

- AV** autonomous vehicle. 59
- BBQ** Bayesian binning into quantiles. 25, 28
- BEV** bird’s-eye-view. 10, 59, 61–64, 68, 71
- BS** Brier score. 23, 30, 71
- CE** calibration error. 17, 18, 31, 33, 53, 54
- ECE** estimated calibration error. 23, 32–34, 48–53, 57, 58, 68, 70, 73, 75, 77
- ETS** ensembled temperature scaling. 27, 28
- GMM** Gaussian Mixture Model. 64–66, 69
- IOP** intra order-preserving. 27, 28, 56
- IRM** multiclass isotonic regression. 27, 28
- KCE** kernel calibration error. 33
- KDE** kernel density estimation. 33
- NLL** negative log-likelihood. 23, 30, 44, 45, 69, 71
- ODIR** off-diagonal and intercept regularisation. 38, 43–47, 74
- OOD** out-of-distribution. 20, 21
- PL** piecewise linear. 52–54, 56–58, 75, 77
- PL3** piecewise linear in logit-logit space. 10, 54–58, 73, 75, 77
- ROQ** region occupancy query. 60, 61, 63, 64, 67, 71, 72, 75

LIST OF ORIGINAL PUBLICATIONS

Publications included in the thesis

- I Kull, Meelis, Perello Nieto, Miquel, **Kängsepp, Markus**, Silva Filho, Telmo, Song, Hao, and Flach, Peter (2019). “Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration”. In: *Advances in Neural Information Processing Systems* 32, pp. 12295–12305.
- II **Kängsepp, Markus**, Valk, Kaspar, and Kull, Meelis (2025). “On the usefulness of the fit-on-test view on evaluating calibration of classifiers”. In: *Machine Learning* 114.4, p. 105. DOI: 10.1007/s10994-024-06652-6.
- III **Kängsepp, Markus** and Kull, Meelis (2026). “Calibrating Perception Uncertainty for Autonomous Driving”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 34.05, pp. 621–645. DOI: 10.1142/S0218488526500212.

Publications not included in the thesis

- IV Vasser, Madis, **Kängsepp, Markus**, Magomedkerimov, Murad, Kilvits, Kälver, Stafinjak, Vladislav, Kivisik, Taavi, Vicente, Raul, and Aru, Jaan (2017). “VREX: an open-source toolbox for creating 3D virtual reality experiments”. In: *BMC psychology* 5.4, pp. 1–8. DOI: 10.1186/s40359-017-0173-4.
- V Ingel, Anti, Shahroudi, Novin, **Kängsepp, Markus**, Tättar, Andre, Komisarenko, Viacheslav, and Kull, Meelis (2020). “Correlated daily time series and forecasting in the M4 competition”. In: *International Journal of Forecasting* 36.1, pp. 121–128. DOI: 10.1016/j.ijforecast.2019.02.018.
- VI Sabir, Ahmed, **Kängsepp, Markus**, and Sharma, Rajesh (2026). “The Confidence Trap: Gender Bias and Predictive Certainty in LLMs”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 40.46, pp. 39173–39181. DOI: 10.1609/aaai.v40i46.41265.
- VII Del, Maksym, **Kängsepp, Markus**, Domnich, Marharyta, Tampuu, Ardi, Yankovskaya, Lisa, Kull, Meelis, and Fishel, Mark (2026). *How Uncertainty Estimation Scales with Sampling in Reasoning Models*. arXiv: 2603.19118 [cs.AI].

Author’s contribution to the publications

In Publication I, the author was mainly responsible for the design of neural network experiments, the implementation of calibration using a neural network architecture, and the development and implementation of ODIR regularisation. The author also contributed to writing the methods, implementation details, and experimental results.

In Publications II and III, the author was involved in the entire research process, including hypothesis formulation, methodology, experimentation, and writing. An exception is Publication II, where the synthetic experiments were designed and conducted by the second author.

1. INTRODUCTION

Machine learning (ML) models are employed across a wide range of domains, including economic, industrial, military, and medical applications. A model may confidently produce incorrect outputs without immediate financial consequences, particularly in applications such as language translation, chatbots, or image enhancement, where errors primarily affect user experience. However, in other domains, such as ad targeting, ML models are directly tied to revenue generation, and their failures can result in significant financial losses. In safety-critical domains, these models are responsible for tasks such as preventing traffic collisions or supporting medical diagnoses, where failures can result in severe harm or even loss of human life. With the growing adoption of machine learning in such high-stakes settings, it is increasingly important to develop models that are not only high-performing but also interpretable and trustworthy.

Machine learning models should be reliable to support informed decision-making and thereby reduce potential losses in safety-critical domains. It is essential to understand when a model is uncertain about its predictions and to act accordingly based on its uncertainty estimates. For example, if an autonomous vehicle’s perception model is not sufficiently confident that the road is clear, it should reduce its speed or request confirmation from a safety driver. In such settings, predicted probabilities may directly determine whether an action is taken, withheld, or referred for human review.

Many models can output predictions with uncertainty, even if it is just a probability that the predicted class is correct. Although probabilistic outputs provide additional information, they might lead to incorrect conclusions, especially if they do not reflect reality. This means that the model might not be well-calibrated. Thus, we cannot trust the model’s predicted uncertainties unless we first check whether they are well-calibrated. For example, a model is considered well-calibrated when it assigns a confidence level (e.g., 80%) to its predictions and approximately the same proportion of those predictions (e.g., 80 out of 100) are correct. On the other hand, when the model is not well-calibrated, it is unknown whether the predicted probability is reliable, meaning the actual probability of being in the correct class might be lower or higher than predicted.

Most models are not calibrated out of the box. Thus, their uncertainties need to be modified. This adjustment of predicted probabilities is called calibration. Existing post-hoc calibration methods often trade flexibility for data efficiency: simple methods may be unable to correct class-specific or inter-class calibration errors, whereas more flexible methods may overfit, particularly in multiclass settings. This motivates the development of calibration methods that can model complex relationships between classes while remaining interpretable and well regularised. In addition to correcting predicted probabilities, it is necessary to evaluate how well calibrated they are. A widely used measure for evaluating calibration is the estimated calibration error (ECE) (Bröcker, 2012; Naeini et al.,

2015). ECE compares average confidence and accuracy within probability bins. However, its value depends on the number and placement of the bins, may conceal local or class-specific errors, and does not identify how the probabilities should be corrected.

These limitations reveal three related methodological challenges. First, flexible and interpretable calibration methods are needed to estimate how predicted probabilities should be corrected, particularly for multiclass models. Second, calibration evaluation depends on how the unknown calibration map is estimated, and selecting a method that provides an appropriate representation and reliable fit is challenging. Third, calibrated uncertainty must ultimately be represented and incorporated into downstream decision-making. Calibration, therefore, involves more than computing a single error measure: it requires methods for correcting uncertainty estimates, evaluating the quality of those corrections, and determining whether the resulting probabilities support reliable decisions.

This dissertation aims to advance the understanding of uncertainty in machine learning through several contributions, guided by the following research questions (RQs):

- **RQ1:** Can we improve multiclass calibration for deep learning models?
- **RQ2:** Why is calibration evaluation a difficult problem to solve, and how can we improve evaluation methods?
- **RQ3:** What uncertainty information supports reliable decision-making in an autonomous perception system?

These research questions are answered in 3 contribution chapters 3–5. Before the contribution chapters, the necessary background information is introduced in **Chapter 2**. That chapter covers background information about uncertainty, calibration, calibration methods, and how to evaluate calibration.

Chapter 3 addresses RQ1 by proposing a novel multiclass post-hoc calibration method for probabilistic classifiers. The proposed method, called Dirichlet calibration, is a generalisation of a flexible binary calibration method, beta calibration. The method helps alleviate issues with calibrating only predicted class confidence by also calibrating classwise relations, making it more versatile and compatible with a wider range of calibration map shapes. The method is likely to overfit; thus, a special regularisation method called ODIR is introduced to keep the model competitive while avoiding overfitting. As a result, Dirichlet calibration combined with ODIR regularisation outperforms temperature scaling on deep neural networks. Chapter 3 presents Publication I, entitled ***Beyond temperature scaling: obtaining well-calibrated multi-class probabilities with Dirichlet calibration*** (Kull, Perello Nieto, et al., 2019).

Chapter 4 addresses RQ2 by introducing a fit-on-test view for calibration evaluation and a novel piecewise linear evaluation method. The fit-on-test view states that the true calibration map is not directly available. Instead, it is approximated using a test dataset, and this approximation is then used to compute calibration error

(CE). Thus, the fit-on-test view highlights several issues: the CE measure depends on the method’s ability to approximate the true calibration map, and the user has no way of assessing how accurate the resulting CE estimate is. Viewing CE estimation as a fit-on-test procedure, we propose using standard machine learning techniques to ensure good generalisation and fit, such as cross-validation and regularisation.

We further propose, as an alternative to synthetic and real data, using pseudo-real data with millions of samples to fit a true calibration map as well as possible, and then using it to better understand the quality of the CE measure. We conduct extensive experiments to identify which CE measure is best for post-hoc calibration, calibration map fitting, CE measurement (ECE), and ranking methods. Chapter 4 presents Publication II, entitled *On the usefulness of the fit-on-test view on evaluating calibration of classifiers* (Kängsepp, Markus, Valk, et al., 2025).

Chapter 5 addresses RQ3 by proposing a framework of uncertainties needed for a safe and reliable autonomous perception system. The proposed framework accounts for uncertainties in detected objects, occupancy maps, and region occupancy queries. Uncertainties about detected objects include location, presence, size, shape, and trajectory probabilities. Region occupancy queries indicate the likelihood that an object is present in a given area, such as a pedestrian crosswalk. The framework is important for capturing different aspects of uncertainty to ensure a reliable and safe system. We propose a way to acquire these uncertainties using the bird’s-eye-view semantic segmentation model FIERY. As we know from previous work, probabilistic models are usually not well calibrated, and this is also true in this setting; thus, these uncertainties in the framework require post-hoc calibration. Finally, we test whether these extra uncertainties are useful for the downstream task of planning, comparing uncalibrated predictions, calibrated semantic segmentation and calibrated object detections. Chapter 5 presents Publication III, entitled *Calibrating perception uncertainty for autonomous driving* (Kängsepp, Markus and Kull, 2026).

The dissertation ends with a discussion of key findings, potential limitations and directions for future work. Together, the three interconnected contributions help better understand how to calibrate different models and what method might work the best; secondly, to understand the limitations of calibration error estimation and how to improve the reliability of estimates across different tasks; and finally, to practically see how calibration affects results in downstream tasks and how to make a safer and more reliable uncertainty framework in perception systems.

2. BACKGROUND AND PRELIMINARIES

This chapter provides background knowledge for the following chapters. The description is from the perspective of the probabilistic classifier, a classification model that outputs probabilities for each predicted class. The following sections cover notation, prediction uncertainty, calibration, and evaluation of calibration of a probabilistic classifier.

2.1. Notation

This section introduces the necessary notation for the binary and multiclass classifiers. Binary classification is a special case of multiclass classification with two classes. Nevertheless, the distinction is retained because many calibration methods and evaluation procedures are defined naturally for a single scalar probability. Methods such as beta calibration, histogram binning, and estimated calibration error (ECE) can be applied directly to binary predictions. Applying analogous procedures to multiclass probability vectors requires additional choices, such as reducing the problem to the confidence of the predicted class or calibrating each class separately. These reductions do not preserve the complete joint structure of the multiclass probability vector. Common concepts are therefore introduced jointly below, while binary- and multiclass-specific notation is distinguished only where necessary.

We begin with a binary classifier $f : \mathcal{X} \rightarrow [0, 1]$ that outputs the predicted probability of a positive class. Let $\mathcal{X}$ denote the input space, and let X be a random variable in the feature space $\mathcal{X}$ representing an input instance. Let Y denote the corresponding ground-truth label in the label space $\{0, 1\}$. The model prediction is $\hat{P} = f(X)$. Let $\hat{P}$ denote the estimated probability of the positive class. So its values are in the range $[0, 1]$, and the negative class probability is $1 - \hat{P}$.

Secondly, a multiclass classifier has a vector of probabilities - $\hat{\mathbf{P}} = (\hat{p}_1, \hat{p}_2, \dots, \hat{p}_k)$ for k classes. Probabilities $\hat{p}_i$ correspond to each of the k classes and sum of these probabilities is one: $\sum_{i=1}^k \hat{p}_i = 1$. Formally, this probability vector can be defined as an element of the $(k - 1)$ -dimensional probability simplex:

$$\Delta^{k-1} = \left\{ (p_1, p_2, \dots, p_k) \in [0, 1]^k \left| \sum_{i=1}^k p_i = 1 \right. \right\}.$$

True classes can be represented as $Y = i$, where $i \in 1, 2, \dots, k$. Alternatively, they can be expressed using a one-hot encoding, where $\mathbf{Y}$ is a k -dimensional vector with a value of 1 at the index of the true class and 0 elsewhere. For example, if $k = 3$ and the second class is the true class, then $\mathbf{Y} = [0, 1, 0]$. Using the $\arg \max$ function (i.e., $\arg \max(\mathbf{Y}) = 2$ in this case), one can recover the index of the true class.

2.2. Uncertainty of probabilistic classifier

The probabilistic classifier is a predictive model that outputs probabilities for each predicted class. The term *confidence* is also used instead of probability when referring to the top-label or predicted class probability, and is commonly used for multiclass classifiers. The aim of probability values is to indicate model uncertainties, but these can be unreliable, as previous works indicate (Niculescu-Mizil and Caruana, 2005; Guo et al., 2017). The higher the probability (confidence), the more certain the model is that the prediction is correct, and vice versa.

Uncertainty is needed in safety-critical areas, for example, medical imaging (Begoli et al., 2019) and autonomous vehicles (Koopman, 2022). Uncertainty information from classifiers helps inform decisions that align with the true class probability. For example, in autonomous driving, we aim to achieve extremely high certainty to avoid any obstacles, as even a one-in-a-million chance of failing to detect an object can have severe consequences. Additionally, for medical imaging, we would like to know when the model is unsure of its prediction and when a doctor should oversee it.

In this work, we treat the model’s predicted probabilities as estimates of predictive uncertainty. Alternative approaches to uncertainty quantification include, for example, Markov chain Monte Carlo (MCMC) methods (Neal, 2012), Bayesian methods (Bishop and Nasrabadi, 2006), ensembling (Lakshminarayanan et al., 2017), and dropout-based methods (Gal and Ghahramani, 2016). These approaches can provide additional insights into the magnitude and nature of uncertainty beyond standard probabilistic outputs. However, they are not considered in detail in this work.

2.3. Calibration of probabilistic classifier

Classifier uncertainty estimates can be misleading, as they reflect the model’s internal confidence, which may not correspond to the true empirical frequencies of the data. For example, a model may consistently assign high confidence (e.g., 90%) to a particular class, while its empirical accuracy at this confidence level is only 70%, indicating that the predicted confidence is overestimated. This implies that the model is overconfident in its predictions, which is a common issue in deep learning models (Guo et al., 2017). In such cases, the model is not well-calibrated, as its predicted probabilities do not match the observed frequencies.

A model may be poorly calibrated for several reasons. Overfitting, limited training data, class imbalance, model misspecification, and optimisation choices can all produce unreliable probabilities even when the test data follow the same distribution as the training data. When representative calibration data are available, post-hoc calibration can often mitigate this type of miscalibration. A different challenge arises under dataset shift (Moreno-Torres et al., 2012) or for out-of-distribution (OOD) inputs. Dataset shift may occur, for example, when a model

trained primarily under sunny conditions is deployed in rainy or foggy conditions. For example, an OOD input may correspond to a previously unseen object, such as a traffic sign or road debris. In these cases, the model encounters inputs that differ from those observed during training and may assign unjustifiably high confidence to unreliable predictions. Ordinary post-hoc calibration does not by itself guarantee reliability under an unknown or changing distribution. Such settings may also require shift detection, adaptation, or dedicated uncertainty estimation methods, which are outside the main scope of this thesis. Under dataset shift or for OOD inputs, the model may therefore become both less accurate and more overconfident. To distinguish reliable probability estimates from such cases, calibration is defined by the correspondence between predicted probabilities and observed outcome frequencies.

In binary classification, a model is well calibrated when predictions assigned a given probability correspond to that outcome with approximately the same frequency. For example, if it predicts the positive class with 85 per cent probability 100 times and is correct 85 out of 100 times. Formally, for a binary classifier

$$P(Y = 1 | f(X) = \hat{p}) = \hat{p},$$

for all $\hat{p} \in [0, 1]$, meaning that for all the probabilities given, the positive class prediction should match with the probability of the event.

Similarly for multiclass calibration (Kull, Perello Nieto, et al., 2019)

$$P(Y = i | f(X) = \hat{\mathbf{p}}) = \hat{p}_i,$$

for all $\hat{\mathbf{p}} \in \Delta^{k-1}$ and for all $i = 1, \dots, k$, where k is the number of classes, $\hat{\mathbf{p}}$ is a vector of predicted probabilities of size k and $\hat{p}_i$ denotes its i -th component. In multiclass calibration, all predicted probabilities for a given input x should be calibrated simultaneously, which is a strong and demanding condition that is difficult both to achieve and to evaluate. Evaluating this full multiclass calibration condition is particularly challenging because the probability space is high-dimensional and individual probability vectors are observed infrequently, leading to sparse data and unreliable estimates. Finite-sample estimation can also be unreliable in binary classification, but the problem becomes more severe as the number of classes and possible probability vectors increases.

Because full multiclass calibration is difficult to estimate and verify, weaker calibration conditions are often considered, including class- k calibration (Kull, Perello Nieto, et al., 2019; Nixon et al., 2019):

$$P(Y = k | f_k(X) = \hat{p}) = \hat{p},$$

for all $\hat{p} \in [0, 1]$, where the calibration condition is evaluated separately for each class k , and $f_k(X)$ denotes the predicted probability for class k .

A related notion is *marginal calibration* (Ananya Kumar et al., 2019), which also evaluates calibration separately for each class but may weigh the class-specific contributions differently when computing calibration error.

An alternative and widely used reduction of the multiclass problem is confidence calibration:

$$P(Y = \arg \max f(X) \mid \max f(X) = \hat{p}) = \hat{p},$$

for all $\hat{p} \in [0, 1]$, as introduced by (Guo et al., 2017). In contrast to approaches that take all class probabilities into account or assess calibration in a classwise manner, this definition considers only the model’s highest-probability prediction for each instance. Confidence calibration is also known as top-label calibration (Ananya Kumar et al., 2019).

These definitions specify what it means for different aspects of a classifier’s output to be calibrated. The following subsection introduces methods for transforming poorly calibrated probabilities to better satisfy the relevant calibration condition.

2.3.1. Calibration methods

Calibration methods aim to improve the reliability of predicted probabilities and can be broadly divided into post-hoc and training-time approaches. Post-hoc methods transform the outputs of an already trained model, whereas training-time methods modify the training procedure itself, for example, through alternative loss functions, regularisation, or softened target labels.

A principal advantage of post-hoc calibration is that the calibrator can be updated without retraining the underlying predictive model. This can be useful when the target distribution changes and representative labelled data from the new distribution are available. Nevertheless, recalibration alone may not be sufficient because distribution shift can degrade both calibration and predictive accuracy. If representative labelled data from the new target distribution are available, they can be used to assess the model and determine whether recalibration is sufficient or whether fine-tuning or complete retraining is required. Ordinary post-hoc calibration does not, by itself, detect or adapt to previously unseen shifts.

Post-hoc calibration requires a separate labelled dataset on which to fit the calibrator. In practice, the validation set used for hyperparameter tuning can be reused for this purpose (Guo et al., 2017), although a separate calibration set may be preferable when sufficient data are available. Under a known distribution shift, new representative labelled data are required to evaluate and recalibrate the model; reusing the original validation set would not account for the shift in the target distribution.

Training-time calibration is incorporated during model fitting. Updating such a method generally requires fine-tuning or retraining the model. Training-time and post-hoc calibration are complementary and may also be combined. The following subsections discuss these approaches in more detail.

2.3.2. Post-hoc calibrators

The idea of a post-hoc calibrator is to estimate the model’s true calibration map. In binary classification, the true calibration map $c_f^*(\hat{p}) = \mathbb{E}[Y \mid \hat{P} = \hat{p}]$ is a mapping

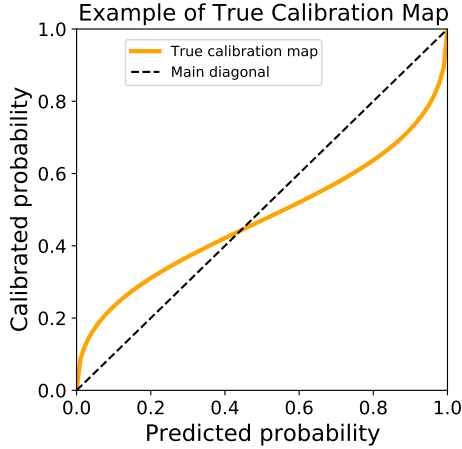


Figure 1: Example of true calibration map for a binary classifier.

that adjusts the model’s confidence to get well-calibrated predictions (Vaicenavicius et al., 2019). In practice, an estimated calibration map $\hat{c}$ is learned to map predicted probabilities $\hat{p}$ to calibrated probabilities $\hat{p}' = \hat{c}(\hat{p})$, thereby producing estimates that better align with the underlying data distribution. Figure 1 shows predicted probabilities on the x-axis and calibrated probabilities on the y-axis. The true calibration map (orange line) shows the predicted probabilities that the model should produce for it to be perfectly calibrated. If the model is already well-calibrated, the calibration map follows the main diagonal (dashed line).

Post-hoc calibration is typically performed by fitting a calibration method on a validation set using a suitable loss function, such as negative log-likelihood (NLL) or Brier score (BS) (Brier, 1950). The input to these methods is either a vector of uncalibrated probabilities $\hat{\mathbf{p}}$ in the multiclass setting or the predicted probability of the positive class $\hat{p}$ in the binary case, together with the corresponding labels $\mathbf{y}$ or y , respectively.

However, some methods instead operate on logits $\mathbf{z} \in \mathbb{R}^k$, that is, the model outputs (e.g., from a neural network) prior to applying the softmax function. The predicted probabilities are then given by $\hat{\mathbf{p}} = \sigma_{\text{sm}}(\mathbf{z})$, where σ_{sm} denotes the softmax function defined as $\sigma_{\text{sm}}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$, and $\sigma_{\text{sm}}(\mathbf{z})_i$ represents the predicted probability assigned to class i . The true calibration map is not available for datasets in practice. Thus, methods such as estimated calibration error (ECE) (section 2.4.2) are used to assess how well the model is calibrated.

The calibration is performed on a validation or calibration dataset, after training the classifier on the training data. This kind of calibration is called post-hoc calibration. The calibration model can be fine-tuned to achieve optimal calibration, for example, by adjusting its hyperparameters or initialisation. Fine-tuning could be done by splitting the validation set into two or using cross-validation to avoid overfitting. Note that cross-validation is performed using proper scoring rules, such as log-loss or the Brier score, rather than the ECE itself. Further, if the

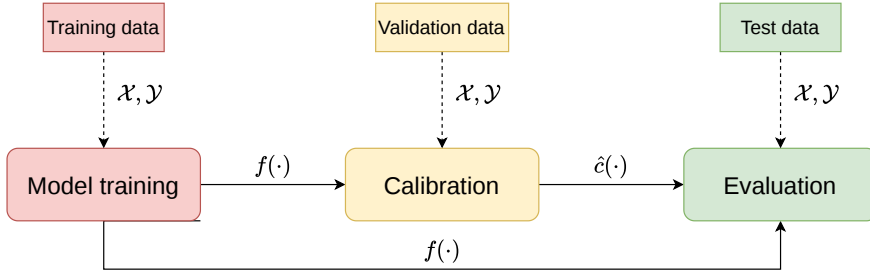


Figure 2: Calibration pipeline for post-hoc calibrators.

calibration method has many parameters, it might overfit the validation set and perform poorly on the test set. In this case, regularisation methods, such as L2 (ridge) regularisation, are used to reduce overfitting. Finally, the test data are used to evaluate the results. Figure 2 visualises this approach.

There are two types of post-hoc calibrators: those designed specifically for binary classification and those applicable to multiclass settings. Only positive class probabilities are calibrated for binary classification. The negative class can be calculated from the positive class probabilities. All class probabilities are jointly calibrated in multiclass calibration. In addition to a multiclass calibrator, a binary calibrator can also be used on a multiclass problem, applying the 1-vs-rest approach (Zadrozny and Elkan, 2002), where a separate calibrator is trained for each class. One of the k classes is taken as a positive prediction, and all other class probabilities are added together as a negative class. After calibrating each class separately, the probabilities might change, so the sum of the probabilities is no longer 1. Thus, the probabilities need to be normalised. The downside of the 1-vs-rest approach is that it cannot learn the classifier’s classwise relations. For example, some classes might be more similar and thus affect each other more than other classes, which could also affect calibration.

There are many binary and multiclass calibration methods, differing in their input representation, model complexity, structural assumptions, and data requirements. Some methods operate directly on predicted probabilities, whereas others require access to the model logits. Parametric methods are generally more data-efficient but may be unable to represent complex calibration maps, while non-parametric methods are more flexible but typically require more calibration data and stronger safeguards against overfitting. Table 1 summarises widely used calibration methods together with selected recent variants.

In the following, $\hat{p}$ denotes a predicted probability, z denotes a binary model score or logit, $\mathbf{z} \in \mathbb{R}^k$ denotes a vector of multiclass logits, σ denotes the logistic function, and σ_{sm} denotes the softmax function.

2.3.3. Binary calibration methods

This section reviews commonly used post-hoc calibration methods for binary classification, ranging from simple parametric transformations to more flexible non-parametric approaches.

Platt scaling (Platt, 2000). Platt scaling fits a two-parameter logistic transformation,

$$\hat{p}' = \sigma(az + b),$$

where $a, b \in \mathbb{R}$. The method is simple, computationally efficient, and relatively robust when the calibration set is small. Its main limitation is the restrictive sigmoid-shaped calibration map, which may not adequately represent more complex forms of miscalibration.

Histogram binning (Zadrozny and Elkan, 2001). Histogram binning partitions the predicted probabilities into m intervals and replaces each prediction with the empirical class frequency in its bin. The bins are chosen to have either equal width or to contain approximately equal numbers of observations. The method is simple and interpretable, but its estimates depend strongly on the number and placement of the bins. Too few bins introduce approximation bias, whereas too many bins produce high-variance estimates.

Isotonic regression (Zadrozny and Elkan, 2002). Isotonic regression fits a monotonically increasing, piecewise-constant calibration map using the pool-adjacent-violators algorithm. It makes fewer assumptions about the shape of the calibration map than Platt scaling and can be trained efficiently even on very large datasets. However, it may overfit when calibration data are limited, and its piecewise-constant output can produce abrupt changes between neighbouring predictions.

Bayesian binning into quantiles (BBQ) (Naeini et al., 2015). Bayesian binning into quantiles (BBQ) extends histogram binning by averaging over multiple candidate binning schemes using Bayesian model averaging. This reduces sensitivity to a single arbitrary choice of bins and provides smoothing when individual bins contain few observations. The additional flexibility comes at a greater computational cost, and the resulting map remains dependent on the considered family of binning models.

Beta calibration (Kull, Silva Filho, et al., 2017). Beta calibration is a three-parameter method defined by

$$\mu_{\text{beta}}(\hat{p}; a, b, c) = \sigma(a \ln \hat{p} - b \ln(1 - \hat{p}) + c),$$

where $a, b \geq 0$ and $c \in \mathbb{R}$. It is motivated by modelling class-conditional predicted probabilities using beta distributions. The family contains the identity map and can represent a wider range of calibration shapes than Platt scaling while retaining a small number of parameters. It is therefore particularly suitable when the calibration set is limited, although its parametric form may still be too restrictive for irregular calibration maps.

Scaling-binning calibration (Ananya Kumar et al., 2019). Scaling-binning calibration first fits a parametric scaling function and then discretises its outputs into equal-frequency bins. The calibrated value within each bin is obtained from the fitted scaling function rather than directly from the average label. This reduces the variance associated with ordinary histogram binning while retaining some of its flexibility. Nevertheless, the method still depends on the selected number of bins and on the suitability of the initial scaling model.

Spline calibration (Gupta et al., 2020). Spline calibration represents the calibration map using piecewise polynomial functions joined at a set of knots. Compared with histogram binning and isotonic regression, splines provide a smoother calibration map and can approximate complex relationships. Their performance depends on the spline degree, number and placement of knots, and regularisation strength. An overly flexible spline may overfit, whereas an insufficient number of knots may produce a biased estimate.

Proximity calibration (Xiong et al., 2024). Proximity calibration uses both prediction confidence and the proximity of an instance to the training data. It is motivated by the observation that predictions in sparsely populated regions may be more overconfident than those in dense regions. This allows the method to capture calibration differences that cannot be identified from confidence alone. Its effectiveness, however, depends on whether the selected representation and distance measure provide a meaningful notion of proximity, and computing proximity may increase inference cost.

2.3.4. Multiclass calibration methods

This section reviews post-hoc calibration methods for multiclass classification, which map a vector of logits or predicted probabilities to a calibrated probability vector. In addition to the flexibility–robustness trade-off discussed above, multiclass methods must determine whether inter-class relationships are modelled and whether the calibration map is allowed to change the predicted class.

Temperature scaling (Guo et al., 2017). Temperature scaling applies a single positive scalar T to all logits:

$$\hat{\mathbf{p}}' = \sigma_{\text{sm}}\left(\frac{\mathbf{z}}{T}\right).$$

The method is straightforward, data-efficient, and effective for many neural networks. Because all logits are scaled by the same value, their ordering is preserved, and the predicted class does not change. Its limited flexibility means that it can correct only global overconfidence or underconfidence and cannot address class-specific calibration errors.

Vector scaling (Guo et al., 2017). Vector scaling applies separate scale and intercept parameters to each class:

$$\hat{\mathbf{p}}' = \sigma_{\text{sm}}(\mathbf{v} \odot \mathbf{z} + \mathbf{b}),$$

where $\mathbf{v}, \mathbf{b} \in \mathbb{R}^k$ and $\odot$ denotes component-wise multiplication. It can correct class-specific differences that temperature scaling cannot represent. However, it requires more calibration data, does not model interactions between classes, and may change the predicted class.

Matrix scaling (Guo et al., 2017). Matrix scaling uses a full linear transformation,

$$\hat{\mathbf{p}}' = \sigma_{\text{sm}}(\mathbf{W}\mathbf{z} + \mathbf{b}),$$

where $\mathbf{W} \in \mathbb{R}^{k \times k}$ and $\mathbf{b} \in \mathbb{R}^k$. The off-diagonal entries allow the method to model interactions between classes, making it more flexible than temperature or vector scaling. This flexibility comes at the cost of $k^2 + k$ parameters, making matrix scaling particularly prone to overfitting when the number of classes is large or the calibration set is small. It may also alter the predicted class.

Mix-n-Match (J. Zhang et al., 2020). The Mix-n-Match framework introduces multiclass calibration methods that combine several simpler calibration mappings. Ensembled temperature scaling (ETS) forms an ensemble of scaling-based predictions and is designed to remain data-efficient while preserving the predicted class. Multiclass isotonic regression (IRM) incorporates non-parametric isotonic components to provide greater flexibility while maintaining ordering constraints. These ensemble methods offer a compromise between simple parametric scaling and highly flexible non-parametric calibration, but their additional components make them less transparent than temperature scaling.

Intra order-preserving functions (Rahimi et al., 2020). The Intra order-preserving (IOP) family constrains the calibration map to preserve the ordering of the class probabilities within each prediction. Consequently, the predicted class remains unchanged. The proposed neural-network parameterisation is more flexible than temperature scaling while retaining this decision-preservation property. The ordering constraint may, however, prevent the method from correcting calibration errors that would require changing the relative ranking of classes, and the neural-network formulation introduces additional optimisation and model-selection complexity.

The post-hoc calibration methods, therefore, represent different trade-offs. Platt scaling, beta calibration, and temperature scaling are attractive when calibration data are limited because they use few parameters. Beta calibration provides greater binary flexibility than Platt scaling, while temperature scaling offers a particularly simple and effective choice for neural-network classifiers. Isotonic regression and histogram-based methods make fewer assumptions about the calibration-map shape and can scale efficiently to large datasets, but their flexibility makes the choice of model complexity important. Vector and matrix scaling can correct class-specific and inter-class errors, respectively, but require progressively more data and may alter the original model decisions. No method is uniformly preferable: the appropriate choice depends on the amount of calibration data, access to logits, expected calibration-map structure, and whether preserving the predicted class is required.

Table 1: Comparison of calibration methods.

Calibration method	Setting	Number of parameters	Input
Platt scaling	Binary	2	Scores or logits
Histogram binning	Binary	Non-parametric	Probabilities
Isotonic regression	Binary	Non-parametric	Probabilities
BBQ	Binary	Non-parametric	Probabilities
Beta calibration	Binary	3	Probabilities
Scaling-binning	Binary	Varies	Varies
Spline calibration	Binary	Depends on knots	Probabilities
Proximity calibration	Binary	Depends	Probabilities and features
Temperature scaling	Multiclass	1	Logits
Vector scaling	Multiclass	$2k$	Logits
Matrix scaling	Multiclass	$k^2 + k$	Logits
Mix-n-Match (ETS)	Multiclass	4	Logits
Mix-n-Match (IRM)	Multiclass	Non-parametric	Probabilities
IOP functions	Multiclass	Depends	Probabilities

2.3.5. Training-time calibrators

In addition to applying post-hoc calibration, classifiers can be trained to produce better-calibrated probabilities directly. Training-time approaches modify the target labels, loss function, data-generation procedure, or model architecture. This literature is substantial but more heterogeneous than post-hoc calibration, since many methods were originally introduced for purposes such as regularisation, class imbalance, or uncertainty estimation and were only subsequently studied from a calibration perspective.

This thesis focuses primarily on post-hoc methods because they can be applied to an already trained model without modifying its architecture or repeating the computationally expensive training process. They also allow the predictive model and the calibration procedure to be studied separately. Nevertheless, training-time methods provide an important alternative, particularly when the training procedure can be modified.

Label smoothing (Szegedy et al., 2016). Label smoothing replaces the one-hot target distribution with a mixture of the original target and a distribution assigning non-zero probability to the other classes. This discourages the model from producing excessively confident predictions and has been shown to improve calibration in several settings (Müller et al., 2019; Maher and Kull, 2021). Its effect depends on the smoothing strength, however, and excessive smoothing may

produce underconfident predictions or remove useful information from the learned logits.

Confidence penalties (Pereyra et al., 2017). A related approach adds a penalty for low-entropy output distributions to the training objective. Unlike label smoothing, which modifies the targets, a confidence penalty acts directly on the predicted distribution and discourages highly concentrated outputs. It is simple to combine with standard classification losses, but the penalty strength must be selected carefully to avoid reducing confidence even for reliable predictions.

Focal loss (T.-Y. Lin et al., 2017). Focal loss was originally introduced to reduce the influence of easy examples in class-imbalanced classification. It modifies the cross-entropy loss by assigning less weight to examples that are already predicted with high confidence. This can restrict excessive growth of the logits and has been shown to improve neural-network calibration (Mukhoti et al., 2020). Its performance depends on the focusing parameter, and calibration improvements are not guaranteed for every model or dataset.

Calibration-aware loss functions. Calibration can also be incorporated more directly into the training objective. For example, the Brier score is a proper scoring rule that jointly reflects the accuracy and quality of probabilistic predictions. The maximum mean calibration error (MMCE) provides a differentiable kernel-based calibration measure that can be added as a regularisation term to the classification loss (Aviral Kumar et al., 2018). Such objectives target calibration more explicitly than label smoothing or focal loss, but they introduce additional hyperparameters and may be sensitive to minibatch-based estimation of the calibration term.

Mixup (H. Zhang et al., 2018). Mixup trains the model using convex combinations of pairs of inputs and their corresponding labels. The resulting soft targets and interpolated training examples encourage smoother decision boundaries and have been shown to improve predictive calibration and uncertainty estimation (Thulasidasan et al., 2019). However, the method assumes that interpolation between training examples and labels is meaningful, and its effectiveness depends on the selected mixing distribution.

Ensemble methods (Lakshminarayanan et al., 2017). Deep ensembles train several models independently and average their predictive distributions. They can improve both predictive performance and uncertainty estimation because disagreement between the models captures some variation that is absent from a single deterministic network. Their principal disadvantage is computational cost, since several models must be trained, stored, and evaluated. Moreover, an ensemble may still require post-hoc calibration.

Training-time and post-hoc methods, therefore, address calibration at different stages. Training-time approaches can influence the learned representation and may improve both accuracy and calibration, but they require access to the original training procedure and often introduce additional optimisation choices. Post-hoc methods are generally more modular and computationally efficient, although their ability to correct miscalibration is limited by the information contained in the

trained model outputs. The two approaches are complementary and may also be combined. More comprehensive overviews of training-time calibration methods are provided in recent surveys (C. Wang, 2023; Silva Filho, Song, Perello-Nieto, et al., 2023).

2.4. Evaluation of calibration

An evaluation method is required to assess the calibration quality of predicted uncertainties. This provides an opportunity to compare uncalibrated and calibrated models and to determine which calibration method is best for the current setting. There are many ways to evaluate how well the model is calibrated, for example, visually using reliability diagrams (Murphy and Winkler, 1977; Bröcker and Smith, 2007), by estimating calibration error (Naeini et al., 2015) or by estimating overall model goodness with proper scoring rules.

Calibration evaluation differs between binary and multiclass classification, except when using proper scoring rules. In binary classification, calibration is typically assessed for the positive class. In contrast, in the multiclass setting, calibration is commonly evaluated either in a confidence-based or classwise manner; these approaches are described in more detail in the following paragraphs. A confidence-based evaluation can also be applied in binary classification; however, it is less commonly used.

2.4.1. Proper scoring rules

Proper scoring rules can be used, for example, to evaluate a model’s performance or as loss functions during training. A scoring rule is proper if it is minimised when the predicted probabilities match the true data-generating distribution, such that lower scores correspond to better-performing models.

Proper scoring rules use predictions $\hat{\mathbf{p}}$ and true labels $\mathbf{y}$ to assess how close the model’s predictions are to the true labels. The closer the prediction is, the lower the score.

The most widely used proper scoring rules in machine learning are negative log-likelihood (NLL) and Brier score (BS) (Brier, 1950). NLL, also known as log-loss or cross-entropy, measures deviance from true class in logarithmic space:

$$\mathcal{L}_{\text{NLL}}(\hat{\mathbf{p}}, \mathbf{y}) = - \sum_{i=1}^k y_i \log \hat{p}_i,$$

where $\mathbf{y}$ is a one-hot encoded label vector.

BS measures the quadratic deviation of prediction vector $\hat{\mathbf{p}}$ from true label $\mathbf{y}$,

$$\mathcal{L}_{\text{BS}}(\hat{\mathbf{p}}, \mathbf{y}) = \sum_{i=1}^k (\hat{p}_i - y_i)^2,$$

where k is number of classes.

The proper scoring rules can be decomposed into calibration and refinement loss (Kull and Flach, 2015). Thus, to achieve a low loss under a proper scoring rule, the model must produce well-calibrated probabilities. However, a near-zero loss on a test dataset does not guarantee well-calibrated predictions for future instances, for example, due to dataset shift.

Many post-hoc calibration methods also use proper scoring rules as a loss measure. For example, log-loss is used in Platt scaling, temperature scaling, and beta calibration, while the Brier score is used as a loss measure in isotonic regression.

Proper scoring rules are suitable for both binary and multiclass calibration evaluation. However, using only a proper scoring rule as a calibration error measure might give a false indication: even if the model has a better score, it does not necessarily mean it is better calibrated, because the refinement loss might decrease without the calibration loss decreasing.

2.4.2. Estimating calibration error

Calibration error (CE) quantifies the discrepancy between predicted confidence and true outcome frequencies:

$$\text{CE}^{(\alpha)} = \mathbb{E} \left[\left| c^*(\hat{P}) - \hat{P} \right|^\alpha \right],$$

where the true calibration map is defined as

$$c^*(\hat{p}) = \mathbb{E}[Y \mid \hat{P} = \hat{p}],$$

and the expectation is taken with respect to the test distribution (Ananya Kumar et al., 2019). The parameter $\alpha \geq 1$ controls the emphasis placed on larger deviations, with $\alpha = 1$ corresponding to absolute error and $\alpha = 2$ to squared error. The map c^* is determined by the relationship between the model’s predicted probabilities and the corresponding outcomes in the target data distribution. It describes how each predicted probability would need to be adjusted to obtain well-calibrated probabilities. Because this distribution depends on both the model and the data, the calibration map may change when the data distribution changes. For synthetic data, the true calibration map may be known by construction. For real-world data, however, it is unknown and must be estimated from a finite validation or test sample. The resulting empirical calibration map depends on the selected function class, fitting procedure, and available observations. Under suitable regularity conditions, a statistically consistent and sufficiently flexible estimator may converge to the true calibration map as the sample size increases. In finite samples, however, reliability depends on the dimensionality of the prediction space, the complexity of the estimator, the density of observations in different probability regions, and class imbalance. Sparse regions of the probability space may therefore remain difficult to estimate accurately even when the overall dataset is large. There is consequently

no universal minimum number of observations required for reliable calibration estimation. A restrictive estimator may have low variance but introduce systematic bias, whereas a highly flexible estimator may reduce approximation bias but overfit limited calibration data. Since the true calibration map is generally unavailable for real-world data, calibration error can only be estimated.

Calibration error is most commonly estimated using estimated calibration error (ECE), also known as expected calibration error (Bröcker, 2012; Naeini et al., 2015). ECE is a binning-based method similar to histogram binning (Zadrozny and Elkan, 2001). Bin vector $\mathbf{B} = [B_1, \dots, B_m, B_{m+1}]$ provides bin boundaries, so that $0 = B_1 < B_2 < \dots < B_m < B_{m+1} = 1 + \varepsilon$, resulting in m bins like $[B_i, B_{i+1})$. A small value ε is added to 1 to ensure that the upper bound is included in the last bin. The probability space is partitioned into m bins using either equal-width or equal-size binning: (1) equal-width binning means that each bin spans an equal interval of the probability space; (2) equal-size binning means that the bins are constructed such that each bin contains an equal number of instances.

$$\text{ECE}^{(\alpha)} = \frac{1}{n} \sum_{j=1}^m n_j |\bar{y}_j - \bar{p}_j|^\alpha,$$

where n_j denotes the number of instances in the j -th bin and m is the total number of bins. The parameter $\alpha \geq 1$ controls the emphasis on larger deviations between $\bar{y}_j$ and $\bar{p}_j$: $\alpha = 1$ corresponds to the absolute error, while $\alpha = 2$ corresponds to the squared error. Average label is $\bar{y}_j = \frac{1}{n_j} \sum_{i: \hat{p}_i \in [B_j, B_{j+1})} y_i$ and average prediction is $\bar{p}_j = \frac{1}{n_j} \sum_{i: \hat{p}_i \in [B_j, B_{j+1})} \hat{p}_i$.

ECE calculations differ slightly between binary and multiclass classification. First, in binary classification, the predicted probability $\hat{p}$ of the positive class is typically used. Second, in the multiclass setting, the most commonly used variant of ECE is confidence-ECE (Guo et al., 2017), which evaluates calibration based on the predicted probability of the most likely class, $\max(\hat{\mathbf{p}})$. Confidence-ECE can also be applied in the binary setting. In the multiclass setting, classwise-ECE (Kull, Perello Nieto, et al., 2019; Nixon et al., 2019) can also be used in a manner analogous to binary classification. Specifically, for each class i , ECE is computed separately and then averaged across all classes, which can be written as

$$\text{ECE}_{\text{cw}}^{(\alpha)} = \frac{1}{n} \sum_{i=1}^k \sum_{j=1}^m n_{i,j} |\bar{y}_{i,j} - \bar{p}_{i,j}|^\alpha.$$

A key limitation of ECE is the choice of the number of bins. It has been shown that different choices can lead to substantially different estimates (Roelofs et al., 2022), and there is no clear guidance on how to select an appropriate number of bins for a given model (Naeini et al., 2015). In addition, ECE is a biased estimate (Ferro and Fricker, 2012), so it should be debiased (Ananya Kumar et al., 2019). Issues with binning or inaccurate calibration-error estimation can lead to decision errors.

One potential addition to binning-based ECE evaluation is ECE_{sweep} (Roelofs et al., 2022). ECE_{sweep} is an extension of a binning-based method that provides a workaround for bin selection. The method incrementally increases the number of equal-sized bins until the monotonicity constraint is violated. It then reverts to the previous step, ensuring that the resulting bins satisfy the monotonicity constraint.

2.4.3. Other evaluation measures

Estimating calibration error with binning-based methods has several flaws. The biggest is choosing the right number of bins for a certain dataset. The following methods for estimating calibration error are kernel-based. A key drawback of these methods is their higher computational cost.

Kernel density estimation (KDE) is a non-parametric binning-free evaluation measure (J. Zhang et al., 2020). KDE outperforms debiased binning-based ECE across datasets with up to 1000 instances, especially on smaller datasets.

Kernel calibration error (KCE) is a non-parametric binning-free evaluation measure for multiclass evaluation (Widmann et al., 2019). KCE can be defined in an unbiased manner and interpreted as a bounded test statistic.

Kolmogorov-Smirnov calibration error (Gupta et al., 2020) is a binning-free calibration evaluation measure that compares classwise cumulative distributions and is inspired by the Kolmogorov-Smirnov statistical test (Smirnov et al., 1939). Kolmogorov-Smirnov error measures classwise-ECE and can be used to visualise CE similar to reliability diagrams (Gupta et al., 2020).

2.4.4. Visualisation and interpretability

Another way to evaluate calibration error is through visualisation. Visualisation provides more fine-grained information than a single scalar metric and can help identify regions where the model is poorly calibrated.

In this context, interpretability refers to the extent to which a calibration representation helps explain how the predicted probabilities are modified. An interpretable representation may reveal where the largest adjustments occur, whether predictions tend to be under- or overconfident, and whether calibration changes the relative probabilities of different classes. Visualisation supports this interpretation by showing aspects of the calibration map that are hidden by a single aggregate error measure.

Reliability diagram (Murphy and Winkler, 1977) is the most commonly used visualisation method for calibration (Figure 3). The reliability diagram is a bar graph (Figure 3a), where each blue bar shows the average label $\bar{y}$, and these bars should align with the diagonal line to have a well-calibrated model. The average prediction $\bar{p}$ is shown with red lines over blue bars (Figure 3c) or alternatively with red bars overlapping blue bars (Guo et al., 2017). These red lines or bars show the calibration gap, showing how much the model, on average, is under- or overconfident. The average label $\bar{y}$ and average prediction $\bar{p}$ are computed

in the same way as for ECE (Section 2.4.2). In addition to binary calibration error, confidence-ECE and classwise-ECE can also be visualised using reliability diagrams. For confidence-ECE, the maximum predicted probability for each sample is used, whereas for classwise-ECE, reliability diagrams are constructed for each class separately. Figure 4 has equal-size binning, meaning that each bin has an equal number of elements, but the probability space for each bin is different. Some bins might be very narrow in densely populated regions. For example, Figure 4a has three last bins on a very narrow space, and it is very difficult to see how well these bins are calibrated. Thus, a reliability diagram with a zoomed section, as in 4b, is helpful for visualising this situation. If the datasets are large and the bins are even narrower, log-scaling can help visualise them.

Reliability diagrams provide useful information about how many predictions fall within different probability intervals. However, classical reliability diagrams lack some useful information. Firstly, where is the bin centre location (Silva Filho, Song, Perelló Nieto, et al., 2023) (Figure 3b), meaning the average prediction $\bar{p}_k$ for k -th bin. Secondly, in equal-width binning, some bins may contain only a few samples. The solution to this is adding a frequency histogram (Silva Filho, Song, Perelló Nieto, et al., 2023) under the figure to show how many instances are falling into each bin (Figure 3c).

Reliability diagrams provide a marginal interpretation of calibration by showing how empirical frequencies vary with predicted probabilities. However, the values are averaged within bins and may therefore conceal variation among predictions in the same interval. In multiclass settings, confidence reliability diagrams further reduce each probability vector to its largest component, while classwise diagrams examine each class separately and do not directly display interactions between classes. These diagrams are therefore informative but provide only a partial interpretation of a multiclass calibration map.

In addition to this type of bar graph, errors in reliability diagrams can also be shown relative to a horizontal line, rather than a diagonal line, for improved readability (Bröcker and Smith, 2007; Vaicenavicius et al., 2019). Figure 5 shows deviation from calibrated results, so a well-calibrated model has all points close to the middle line. One possible extension to reliability diagrams is to add consistency bars (Vaicenavicius et al., 2019). The consistency bar shows how much confidence one should have in the deviation. The bigger the bar, the more the actual deviation might vary from the one shown.

Interpreting calibration becomes increasingly difficult as the number of classes grows because a multiclass calibration map transforms one probability vector into another. Reliability diagrams provide classwise or confidence-based projections of this transformation, but they do not show the complete relationship among the class probabilities. For a three-class problem, the full probability vector can instead be visualised on the two-dimensional probability simplex. Formally, the probability simplex is defined as $\Delta^{k-1} = \{(p_1, p_2, \dots, p_k) \in [0, 1]^k \mid \sum_{i=1}^k p_i = 1\}$.

Figure 6 illustrates a calibration map for a three-class problem. Each vertex of

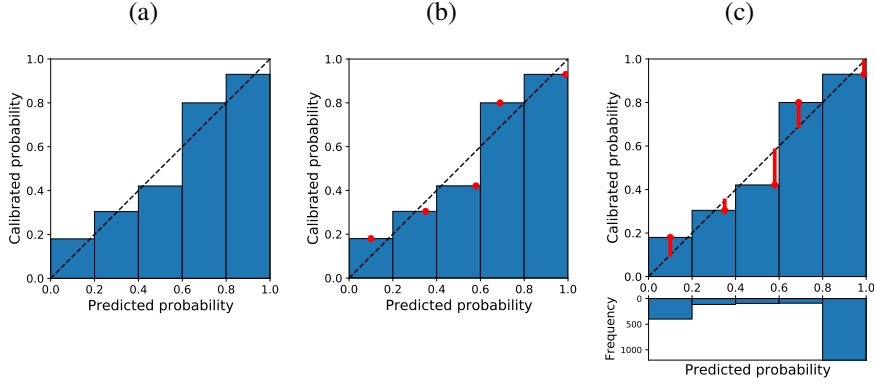


Figure 3: Example of reliability diagram variants: (a) equal-width binning showing only average labels $\bar{y}_k$, (b) equal-width binning with red dots indicating bin centres ($\bar{p}_k$) and (c) equal-width binning with an added histogram of sample frequencies and vertical lines indicating calibration gaps. Adapted from Publication II.

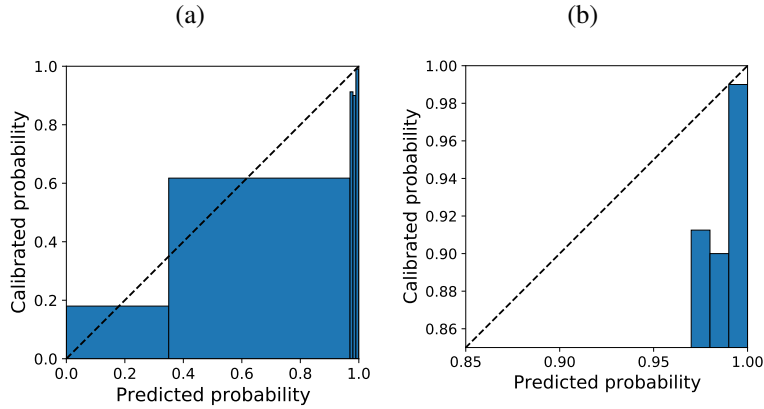


Figure 4: Example of a reliability diagram with equal-size binning, each bin has the same number of elements. The figure on the right (b) is zoomed in to show the last 3 bins.

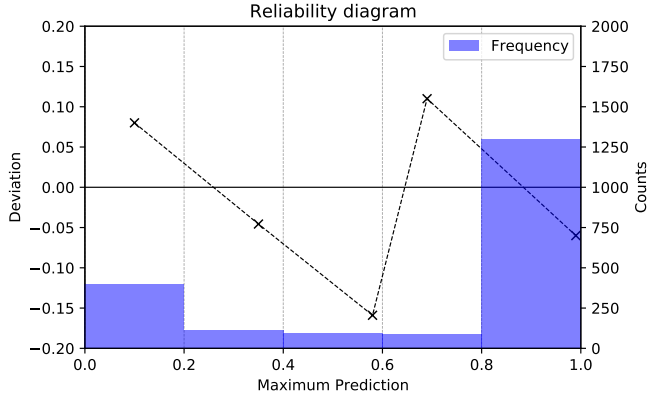


Figure 5: Example of reliability diagram with deviations instead of showing average accuracy and average probability for each bin (Vaicenavicius et al., 2019). The blue bars indicate the sample frequencies in each bin.

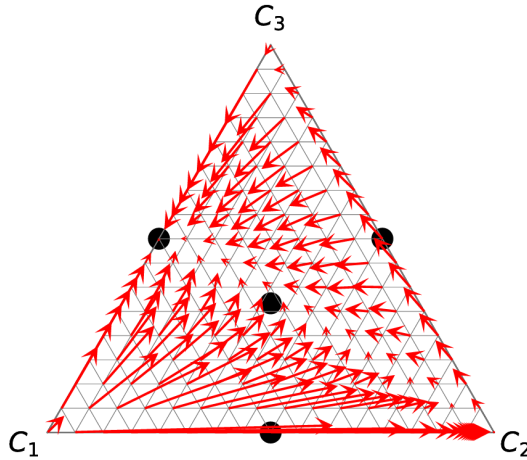


Figure 6: Example of a probability simplex for visualising 3-class calibration. The arrows mark the change of probabilities after calibration. Adapted from Publication I.

the triangle corresponds to assigning probability 1 to a particular class, while points on an edge assign probability 0 to the remaining class. The centre corresponds to equal probabilities for all three classes. The arrows connect the original and calibrated probability vectors, thereby showing both the direction and magnitude of the calibration adjustment.

The simplex provides a more complete interpretation than separate classwise diagrams for three-class problems because it preserves the relationship among all class probabilities. However, this representation does not extend directly beyond three classes. For larger multiclass problems, interpretation must therefore rely on complementary representations, such as classwise reliability diagrams, lower-

dimensional projections, or parameter matrices. Each representation reveals a different aspect of the calibration map, but none alone fully characterises its behaviour.

3. MULTICLASS CALIBRATION METHOD - DIRICHLET CALIBRATION (PUBLICATION I)

3.1. Introduction

This chapter introduces the first contribution of the thesis, including **RQ1**: *Can we improve multiclass calibration for deep learning models?*

The most popular and used multiclass calibration method for deep neural networks is temperature scaling (Guo et al., 2017). It has a single parameter and generally alleviates the issue of overconfidence effectively.

This chapter aims to go beyond temperature scaling by highlighting its potential shortcomings and proposing a new, versatile multiclass post-hoc calibration method. The proposed method, called Dirichlet calibration, generalises beta calibration (Kull, Silva Filho, et al., 2017) method to the multiclass setting. We begin by describing Dirichlet calibration method for neural networks, followed by a discussion of its interpretability. We then compare it with temperature scaling and matrix scaling, and subsequently present its implementation. Next, we propose a specialised regularisation method, off-diagonal and intercept regularisation (ODIR), to mitigate overfitting in settings with a large number of classes. Finally, we evaluate the performance of Dirichlet calibration across various experiments and compare it with other popular post-hoc calibration methods.

3.2. Motivation

The background chapter reviewed a range of post-hoc calibration methods; however, at the time our method was proposed, temperature scaling remained the dominant approach for deep neural networks. Matrix scaling is a powerful calibration method but is prone to overfitting and operates in logit space, as does temperature scaling. While temperature scaling can be applied in multiclass settings, its single parameter limits its flexibility and prevents it from achieving proper classwise calibration, as illustrated in Figure 7. The figure compares confidence reliability diagrams (Figures 7a and 7b) before and after calibration, indicating near-perfect calibration of confidence. However, when examining individual classes (e.g., class 2 in Figures 7c and 7d), Dirichlet calibration demonstrates superior performance compared to temperature scaling. These observations highlight the need for more flexible calibration methods that can simultaneously achieve classwise calibration based on the model’s probabilistic outputs.

3.3. Methodology

Dirichlet calibration is a post-hoc calibration method for multiclass probabilistic classifiers. Its motivation follows from the structure of multiclass predictions: each prediction is a probability vector whose components are non-negative and sum to

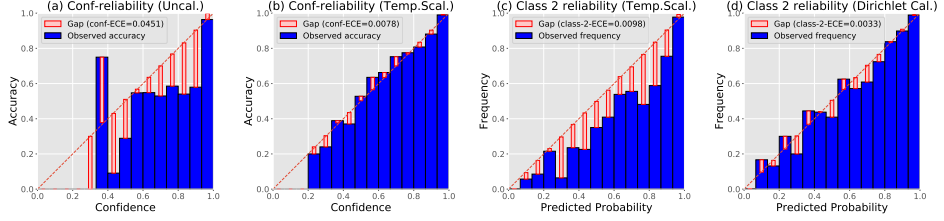


Figure 7: Reliability diagrams for WideNet32 (Zagoruyko and Komodakis, 2016) on CIFAR-10. Subfigure (a) shows the confidence reliability diagram prior to calibration, while (b) presents the same after temperature scaling. Panels (c) and (d) display classwise reliability diagrams for class 2 after temperature scaling and Dirichlet calibration, respectively. Adapted from Publication I.

1. Such vectors lie on the probability simplex, for which the Dirichlet distribution provides a natural probabilistic model.

Assuming that the predicted probability vectors follow class-conditional Dirichlet distributions, Bayes’ rule leads to posterior class probabilities that can be expressed as a linear transformation of the logarithms of the predicted probabilities followed by a softmax transformation. This produces a flexible calibration map capable of correcting both class-specific biases and interactions among predicted class probabilities.

Dirichlet calibration can also be understood as a multiclass generalisation of beta calibration. Beta calibration models the predicted probability in binary classification using separate class-conditional beta distributions for the positive and negative classes. Since the beta distribution is a special case of the Dirichlet distribution for two classes, extending the same construction to probability vectors with more than two components leads naturally to Dirichlet calibration.

We introduce two parametrisations: a canonical parametrisation for improved interpretability and a linear parametrisation for easier implementation. Both parametrisations yield equivalent calibration maps. The parametrisations are then described mathematically.

The linear parametrisation is given by

$$\hat{\mu}_{\text{DirLin}}(\hat{\mathbf{p}}; \mathbf{W}, \mathbf{b}) = \sigma_{\text{sm}}(\mathbf{W} \ln(\hat{\mathbf{p}}) + \mathbf{b}),$$

where $\mathbf{W} \in \mathbb{R}^{k \times k}$ is a weight matrix, $\mathbf{b} \in \mathbb{R}^k$ is a bias vector, and $\ln(\hat{\mathbf{p}})$ denotes the element-wise natural logarithm of the probability vector $\hat{\mathbf{p}}$. The softmax function σ_{sm} maps the transformed logits back to the probability simplex, where $\sigma_{\text{sm}}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$.

The canonical parametrisation is given by

$$\hat{\mu}_{\text{Dir}}(\hat{\mathbf{p}}; \mathbf{A}, \mathbf{c}) = \sigma_{\text{sm}}\left(\mathbf{A} \ln\left(\frac{\hat{\mathbf{p}}}{1/k}\right) + \ln \mathbf{c}\right),$$

where the matrix $\mathbf{A} \in [0, \infty)^{k \times k}$ has non-negative entries, and each column contains at least one zero. The term $\frac{\hat{\mathbf{p}}}{1/k}$ denotes component-wise division, and $\mathbf{c} \in \Delta^{k-1}$ is a

probability vector of length k . The linear parametrisation can be readily converted into the canonical form, which is used in the following paragraph for improved interpretability.

The linear and canonical parametrisations differ in form but define the same family of calibration maps, as stated in Theorem 1.

Theorem 1 (Equivalence of linear and canonical parametrisations). *The parametric families $\hat{\mu}_{\text{DirLin}}(\hat{\mathbf{p}}; \mathbf{W}, \mathbf{b})$ and $\hat{\mu}_{\text{Dir}}(\hat{\mathbf{p}}; \mathbf{A}, \mathbf{c})$ are equal, i.e. they contain exactly the same calibration maps.*

Proof. The proof is given in Publication I. ■

3.3.1. Interpretability

The canonical parameterisation of Dirichlet calibration improves interpretability in two main ways. First, the intercept parameter $\mathbf{c}$ has a direct probabilistic interpretation: it represents the calibrated output distribution when the input prediction is uniform. Second, the entries of the coefficient matrix $\mathbf{A}$ are non-negative and identifiable relative to a zero-valued reference entry in each column. An entry A_{ij} describes the relative strength with which evidence for input class j contributes to the calibrated logit of output class i .

The matrix $\mathbf{A}$ can therefore be used to compare class-specific interactions. Its entries should not, however, be interpreted as direct marginal changes in the calibrated class probabilities. The final effect of an individual coefficient depends on the complete input probability vector and on the other parameters of the calibration map. Consequently, although the canonical parameterisation provides insight into relationships between classes, interpreting the complete behaviour of a multiclass calibration map remains challenging.

Figure 8 illustrates the interpretation of a Dirichlet calibration map for a convolutional neural network trained on the SVHN (Netzer et al., 2011) dataset, which contains 10 classes corresponding to house-number digits. The left panel, Figure 8a, shows the coefficient matrix of the canonical parameterisation. The diagonal entries describe class-specific effects, whereas the off-diagonal entries describe interactions between different input and output classes. For example, the relatively large value of 0.63 in entry (3, 9) indicates that evidence for input class 8 contributes comparatively strongly to the calibrated logit associated with class 2.

This coefficient does not by itself imply a fixed increase in the calibrated probability of class 2. Rather, its effect depends on the full uncalibrated probability vector and on the remaining coefficients of the calibration map. In the examples considered here, the resulting calibration map shifts selected class-specific decision boundaries and produces 26 fewer misclassifications. The corresponding changes in the confusion matrix are shown in Figure 8b. This example demonstrates how selected coefficients can provide limited insight into class-specific decisions, even

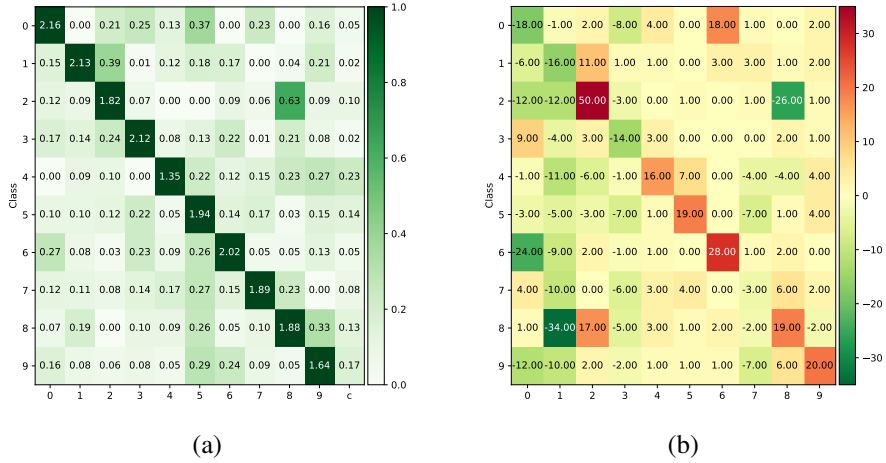


Figure 8: Interpretation of a Dirichlet calibration mapping: (a) the coefficient matrix in the canonical parameterisation for SVHN_convnet; and (b) the resulting changes in the confusion matrix after applying the calibration map. The matrix entries describe relative contributions to the calibrated logits and should not be interpreted as direct marginal changes in the calibrated probabilities. Adapted from Publication I.

though no individual coefficient fully characterises the behaviour of the calibration map.

More generally, interpreting the complete behaviour of a multiclass calibration map requires combining several complementary representations. The coefficient matrix reveals relationships between input and output classes, whereas classwise reliability diagrams show how calibration changes the predicted probabilities for individual classes. Together, these views provide a more complete interpretation than either representation alone.

3.3.2. Comparison to other methods

Dirichlet calibration is closely related to beta calibration and scaling-based calibration methods. In the binary case, Dirichlet calibration reduces to beta calibration. The beta-calibration results, therefore, illustrate the behaviour of the underlying modelling principle in the binary setting, while the Dirichlet-calibration results demonstrate its extension to multiclass predictions. The principal difference between Dirichlet calibration and scaling methods, such as temperature, vector, and matrix scaling, concerns the quantities to which their parameterisations are applied. Dirichlet calibration operates on probability vectors, whereas scaling methods operate on logits, that is, the raw model outputs before application of the softmax function. Despite this difference, the Dirichlet calibration family contains all temperature-scaling mappings; see Publication I for further details.

The implementation of Dirichlet calibration is nevertheless closely related to

that of matrix scaling, as discussed in Section 3.3.3. Matrix scaling applies a linear transformation directly to the logits. However, its parameters do not generally admit the same interpretation as those of the canonical parameterisation of Dirichlet calibration. An element W_{ij} describes the contribution of input logit z_j to the transformed logit of class i , but it does not correspond directly to a change in the calibrated probability of that class because the final softmax transformation depends jointly on all transformed logits. In addition, the conventional softmax parameterisation is not fully identifiable without further constraints, since different parameter values can represent the same calibrated probabilities. Individual coefficients may therefore provide some insight into interactions between class logits, but their interpretation depends on the remaining parameters of the model. By contrast, the canonical parameterisation of Dirichlet calibration operates directly on the input probabilities and imposes constraints that provide an identifiable representation relative to a reference entry.

The choice of calibration method also determines whether the model’s original classification decisions can change. Temperature scaling adjusts prediction confidence while preserving the class with the highest predicted probability. This property may be desirable in applications where calibration should not modify the model’s original decisions. By contrast, more flexible methods, including vector scaling, matrix scaling, and Dirichlet calibration, can change the predicted class when correcting class-specific or inter-class calibration errors. This additional flexibility may improve calibration or classification performance, but it also allows the calibration map to act as a decision-correction mechanism rather than solely as a confidence adjustment.

3.3.3. Implementation

Dirichlet calibration can be straightforwardly implemented using neural networks. First, the logarithm of the predicted probabilities is taken, and the transformed inputs are then passed through a fully connected layer. Finally, a softmax function is applied to map the outputs back to the probability space. Figure 9 illustrates Dirichlet calibration implementation for neural networks. This implementation is closely related to that of matrix scaling (Guo et al., 2017). The main difference is that matrix scaling operates on logits, whereas Dirichlet calibration uses the logarithms of probabilities (i.e., pseudo-logits). The transformation from logits to probabilities and back to logits is lossy, as it removes one degree of freedom. This occurs because probabilities encode only relative differences between logits, not their absolute values; thus, logits can only be recovered up to an additive constant. Therefore, matrix scaling is preferable when logits are available, as also supported by the experimental results.

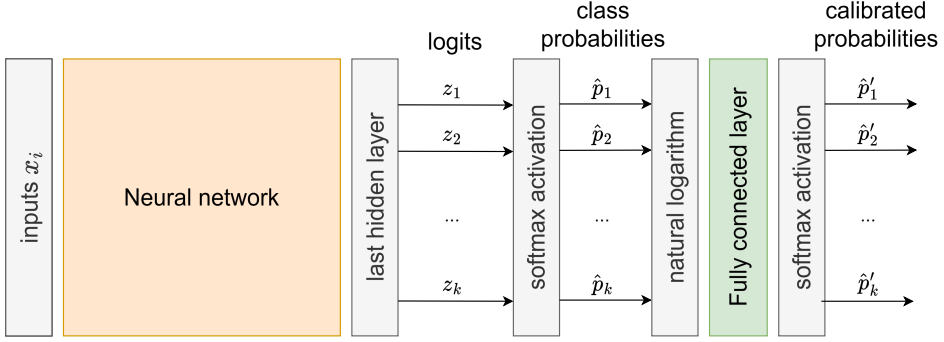


Figure 9: Implementation example of Dirichlet calibration method using neural networks.

3.3.4. Off-diagonal and intercept regularisation

Matrix scaling and Dirichlet calibration contain many parameters and are therefore prone to overfitting. Standard ridge regression, or L2 regularisation, can mitigate this problem, but it was insufficient for neural network calibration in our experiments. We therefore introduce off-diagonal and intercept regularisation (ODIR), which regularises the off-diagonal and intercept parameters separately:

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^n \text{logloss}(\hat{\mu}_{\text{DirLin}}(\hat{\mathbf{p}}(x_i); \mathbf{W}, \mathbf{b}), y_i) + \lambda \cdot \left(\frac{1}{k(k-1)} \sum_{i \neq j} w_{ij}^2 \right) + \mu \cdot \left(\frac{1}{k} \sum_j b_j^2 \right)$$

where w_{ij} denotes an entry of the weight matrix $\mathbf{W}$, b_j denotes an entry of the intercept vector $\mathbf{b}$, and λ and μ are tunable regularisation hyperparameters.

The diagonal entries of $\mathbf{W}$ primarily control class-specific calibration, whereas the off-diagonal entries capture interactions between classes. ODIR therefore penalises the off-diagonal entries using λ and the intercept terms using μ , while leaving the diagonal entries unregularised. By contrast, standard L2 regularisation penalises all parameters jointly. Because the diagonal entries are typically larger, they receive a stronger absolute penalty, while the smaller off-diagonal entries remain relatively less constrained.

In the experiments (Table 5), the most frequently selected value of λ was 0.25, with selected values ranging from 0.005 to 5000. For μ , the most frequently selected value was 0.001, with values ranging from 10^{-5} to 10^6 . Since both penalty terms are normalised by the number of parameters they contain, the magnitudes of λ and μ primarily reflect the degree of regularisation required by the two parameter groups. Their difference is therefore more indicative of the relative tendency of the off-diagonal and intercept parameters to overfit than of any particular class or the overall difficulty of the dataset.

The current formulation treats all classes equally. When errors involving a particular class carry substantially greater risk, this asymmetry is generally better addressed through a cost-sensitive downstream decision rule. Class-specific loss weights or regularisation terms could also be introduced, but doing so would make the calibration objective application-dependent and could conflate probability calibration with decision-cost optimisation.

As discussed previously, Dirichlet calibration and matrix scaling have similar implementations. ODIR can therefore also be applied to matrix scaling, improving its effectiveness despite earlier findings that matrix scaling is particularly prone to overfitting on datasets with many classes (Guo et al., 2017).

3.4. Experiments

Experiments were conducted on both traditional datasets and neural network models. The experiments with traditional datasets were carried out by the second author of Publication I (Kull, Perello Nieto, et al., 2019).

The non-neural experiments comprise 21 UCI datasets (J. Wang, 2023) and 11 non-neural classifier configurations (e.g., KNN, decision trees, and Naive Bayes), resulting in 231 experimental settings. The optimal post-hoc calibrator was selected from eight methods: two binning approaches (equal-width (EW) and equal-size (ES)), isotonic regression (Isot), beta calibration (Beta), temperature scaling (TempS), vector scaling (VecS), and Dirichlet calibration with L2 (DirL2) and ODIR regularisation (DirODIR).

Binary calibration methods were implemented using a one-vs-rest approach, fitting a separate calibration map for each class, followed by normalisation. The calibration methods were trained using 3-fold cross-validation, with NLL as the loss function, and the final calibrated predictions were obtained by averaging the outputs of the three folds, following the approach of Platt (2000). The hyperparameters of calibration methods were selected using the training folds within the 3-fold cross-validation. The number of folds was chosen to be three due to the limited number of data instances.

For evaluation, classwise-ECE and log-loss (NLL) were used as the primary metrics, as reported in Tables 2 and 3, respectively. Dirichlet calibration with ODIR regularisation achieved the lowest classwise-ECE, closely followed by vector scaling. The best log-loss was obtained by Dirichlet calibration with L2 regularisation, followed by Dirichlet calibration with ODIR regularisation.

Neural network experiments were conducted on three image classification datasets: CIFAR-10 (Krizhevsky, 2012), CIFAR-100 (Krizhevsky, 2012), and SVHN (Netzer et al., 2011), using six convolutional neural network architectures (ResNet110 (He et al., 2015), ResNet110-SD (Huang, Sun, et al., 2016), ResNet152-SD (Huang, Sun, et al., 2016), DenseNet40 (Huang, Liu, et al., 2016), WideNet32 (Zagoruyko and Komodakis, 2016), and LeNet-5 (LeCun et al., 1998)) as well as three pretrained convolutional models (Cheni, 2017). In total, six models

were evaluated on the CIFAR-10/100 datasets, and two on the SVHN dataset.

In these experiments, the proposed method was compared against the best-performing approaches from Guo et al. (Guo et al., 2017), namely temperature scaling, vector scaling, and matrix scaling. To mitigate overfitting, ODIR regularisation was applied to matrix scaling, in addition to Dirichlet calibration. The hyperparameter tuning and averaging procedures in the neural network experiments were performed similarly to those in the non-neural experiments, except that 5-fold cross-validation was used due to the larger amount of available data. Both Dirichlet calibration with L2 regularisation and ODIR regularisation were evaluated.

For evaluation, log-loss (NLL) and classwise-ECE were used as the primary metrics, as reported in Tables 4 and 5, respectively.

In the experiments, the proposed ODIR regularisation improved the performance of both matrix scaling and Dirichlet calibration compared to temperature scaling. Vector scaling achieved the lowest classwise-ECE overall, followed by matrix scaling and Dirichlet calibration with ODIR regularisation. On CIFAR-10, Dirichlet calibration with L2 regularisation achieved the lowest classwise-ECE,

Table 2: Classwise-ECE rankings of non-neural models, computed over 21 UCI datasets. Adapted from Publication I.

	Uncal	DirL2	DirODIR	Beta	TempS	VecS	Isot	ES	EW
adas	6.7	3.3	2.9	5.4	6.3	3.4	4.0	6.4	6.5
forest	7.6	5.6	3.4	3.6	5.0	2.1	2.7	8.3	6.7
knn	4.1	4.6	3.1	4.2	5.5	2.5	5.4	7.9	7.6
lda	6.3	4.1	3.4	3.4	6.8	3.8	3.0	7.9	6.3
logistic	8.0	3.1	3.6	3.8	6.1	3.4	3.1	7.8	5.9
mlp	5.5	4.1	3.2	3.2	6.3	3.8	2.7	8.6	7.6
nbayes	7.4	2.8	1.9	5.0	7.8	3.4	3.8	7.0	6.0
qda	6.5	3.6	2.6	4.0	7.0	4.0	2.5	8.3	6.3
svc-linear	7.1	3.7	3.8	4.0	4.8	3.2	3.2	8.5	6.7
svc-rbf	6.0	4.0	4.0	4.2	5.0	3.7	3.3	8.6	6.2
tree	2.2	6.3	4.0	3.7	4.2	3.4	5.0	8.5	7.7
avg rank	6.12	4.12	3.28	4.05	5.89	3.34	3.54	7.99	6.67

Table 3: Log-loss (NLL) rankings of non-neural models, computed over 21 UCI datasets. Adapted from Publication I.

	Uncal	DirL2	DirODIR	Beta	TempS	VecS	Isot	ES	EW
adas	8.4	2.1	2.0	4.7	7.8	5.1	4.9	4.8	5.1
forest	7.0	5.7	4.4	3.1	3.9	3.4	5.8	6.8	4.8
knn	8.2	2.8	4.9	6.4	7.2	5.4	2.7	4.4	3.0
lda	7.4	1.9	3.0	3.5	5.6	4.0	7.2	7.1	5.2
logistic	7.6	1.7	2.5	3.2	5.3	4.0	8.1	7.6	5.0
mlp	4.5	2.7	4.5	3.1	3.5	4.0	8.0	8.6	6.2
nbayes	8.6	1.4	2.5	4.7	6.2	4.4	5.8	4.9	6.5
qda	7.5	2.2	3.0	3.8	5.4	4.5	6.7	7.3	4.6
svc-linear	6.7	1.9	2.3	3.7	4.4	3.7	8.1	8.1	6.0
svc-rbf	7.6	4.0	3.4	3.7	5.7	2.9	6.0	5.4	6.3
tree	6.2	3.6	5.7	6.2	7.4	6.7	2.0	4.0	3.2
avg rank	7.24	2.73	3.48	4.21	5.66	4.38	5.93	6.28	5.10

followed by vector scaling and methods using ODIR. Dirichlet calibration with ODIR regularisation consistently outperforms its L2-regularised counterpart on the CIFAR-100 dataset, highlighting the need for more effective regularisation in settings with a large number of classes.

The lowest log-loss was achieved by matrix scaling with ODIR regularisation over all datasets, followed closely by Dirichlet calibration with ODIR. The performance difference between matrix scaling and Dirichlet calibration was marginal. However, since matrix scaling operates directly on logits and does not rely on a lossy transformation, it consistently achieved slightly better results.

Table 4: Comparison of calibration methods using classwise-ECE scores and rankings across deep neural network experiments. Adapted from Publication I.

	Uncal	general-purpose calibrators			calibrators using logits	
		TempS	Dir-L2	Dir-ODIR	VecS	MS-ODIR
c10_convnet	0.104 ₆	0.044 ₄	0.043 ₂	0.045 ₅	0.043₁	0.044 ₃
c10_densenet40	0.114 ₆	0.040 ₅	0.034₁	0.037 ₄	0.036 ₂	0.037 ₃
c10_lenet5	0.198 ₆	0.171 ₅	0.052₁	0.059 ₄	0.057 ₂	0.059 ₃
c10_resnet110	0.098 ₆	0.043 ₅	0.032₁	0.039 ₄	0.037 ₃	0.036 ₂
c10_resnet110_SD	0.086 ₆	0.031 ₄	0.031 ₅	0.029 ₃	0.027 ₂	0.027₁
c10_resnet_wide32	0.095 ₆	0.048 ₅	0.032 ₃	0.029 ₂	0.032 ₄	0.029₁
c100_convnet	0.424 ₆	0.227₁	0.402 ₅	0.240 ₃	0.241 ₄	0.240 ₂
c100_densenet40	0.470 ₆	0.187 ₂	0.330 ₅	0.186₁	0.189 ₃	0.191 ₄
c100_lenet5	0.473 ₆	0.385 ₅	0.219 ₄	0.213 ₂	0.203₁	0.214 ₃
c100_resnet110	0.416 ₆	0.201 ₃	0.359 ₅	0.186₁	0.194 ₂	0.203 ₄
c100_resnet110_SD	0.375 ₆	0.203 ₄	0.373 ₅	0.189 ₃	0.170₁	0.186 ₂
c100_resnet_wide32	0.420 ₆	0.186 ₄	0.333 ₅	0.180 ₂	0.171₁	0.180 ₃
SVHN_convnet	0.159 ₆	0.038 ₄	0.043 ₅	0.026 ₂	0.025₁	0.027 ₃
SVHN_resnet152_SD	0.019 ₂	0.018₁	0.022 ₆	0.020 ₃	0.021 ₅	0.021 ₄
Average rank	5.71	3.71	3.79	2.79	2.29	2.71

Table 5: Comparison of calibration methods using log-loss (NLL) scores and rankings across deep neural network experiments. Adapted from Publication I.

Model	Uncal	general-purpose calibrators			calibrators using logits	
		TempS	Dir-L2	Dir-ODIR	VecS	MS-ODIR
c10_convnet	0.391 ₆	0.195₁	0.197 ₄	0.195 ₂	0.197 ₅	0.196 ₃
c10_densenet40	0.428 ₆	0.225 ₅	0.220₁	0.224 ₄	0.223 ₃	0.222 ₂
c10_lenet5	0.823 ₆	0.800 ₅	0.744 ₂	0.744 ₃	0.747 ₄	0.743₁
c10_resnet110	0.358 ₆	0.209 ₅	0.203₁	0.205 ₃	0.206 ₄	0.204 ₂
c10_resnet110_SD	0.303 ₆	0.178 ₅	0.177 ₄	0.176 ₃	0.175 ₂	0.175₁
c10_resnet_wide32	0.382 ₆	0.191 ₅	0.185 ₄	0.182 ₂	0.183 ₃	0.182₁
c100_convnet	1.641 ₆	0.942₁	1.189 ₅	0.961 ₂	0.964 ₄	0.961 ₃
c100_densenet40	2.017 ₆	1.057 ₂	1.253 ₅	1.059 ₄	1.058 ₃	1.051₁
c100_lenet5	2.784 ₆	2.650 ₅	2.595 ₄	2.490 ₂	2.516 ₃	2.487₁
c100_resnet110	1.694 ₆	1.092 ₃	1.212 ₅	1.096 ₄	1.089 ₂	1.074₁
c100_resnet110_SD	1.353 ₆	0.942 ₃	1.198 ₅	0.945 ₄	0.923₁	0.927 ₂
c100_resnet_wide32	1.802 ₆	0.945 ₃	1.087 ₅	0.953 ₄	0.937 ₂	0.933₁
SVHN_convnet	0.205 ₆	0.151 ₅	0.142 ₃	0.138 ₂	0.144 ₄	0.138₁
SVHN_resnet152_SD	0.085 ₆	0.079₁	0.085 ₅	0.080 ₂	0.081 ₄	0.081 ₃
Average rank	6.0	3.5	3.79	2.93	3.14	1.64

3.5. Summary and limitations

In conclusion, we proposed a novel multiclass post-hoc calibration method, Dirichlet calibration, together with the regularisation method ODIR. We examined the interpretability of Dirichlet calibration map and showed that it can provide some insight into potential issues in uncalibrated model outputs. Furthermore, we evaluated performance across a diverse range of datasets and models in both non-neural and neural settings. The results indicate that Dirichlet calibration and matrix scaling with ODIR regularisation achieve strong performance on convolutional neural networks.

The main limitation is the lack of evaluation across different types of deep neural networks and datasets, as the current experiments focus primarily on convolutional architectures. Additionally, it would be valuable to investigate how different calibration methods behave under dataset shift and how effective they are for detecting out-of-distribution samples. Finally, hyperparameter tuning for ODIR regularisation introduces additional computational cost, as two parameters, one for off-diagonal terms and one for intercepts, must be selected. Even when considering only 10 values per hyperparameter, this results in 100 combinations, each requiring five additional fits due to 5-fold cross-validation. This makes the calibration process significantly slower compared to other methods.

4. RETHINKING EVALUATION OF CALIBRATION: FIT-ON-TEST (PUBLICATION II)

4.1. Introduction

This chapter provides an overview of the second contribution of the thesis, addressing **RQ2**: *Why calibration evaluation is a difficult problem to solve and how to improve evaluation methods?*

Prior work has shown that deep neural networks can be poorly calibrated (Guo et al., 2017). To assess how well a model is calibrated, scoring metrics such as ECE and visualisation tools such as reliability diagrams are commonly used. However, existing calibration measures have several limitations, making evaluation results sensitive to method and hyperparameter choices.

We begin with an overview of the issues and limitations of current calibration evaluation methods, followed by a discussion of a key challenge: the absence of a ground-truth calibration map. This work brings together different evaluation approaches and requirements, and provides a comparison of evaluation methods across synthetic, pseudo-real, and real datasets. We introduce a fit-on-test approach for calibration evaluation and assess existing evaluation methods across three tasks. Furthermore, we propose new calibration evaluation methods based on this approach. Finally, we introduce a pseudo-real dataset derived from the CIFAR-10 dataset to better understand how calibration evaluation methods perform in calibration assessment, calibration map prediction, and calibrator ranking.

4.2. Motivation

ECE (Naeini et al., 2015) is one of the most widely used methods for evaluating calibration. It partitions instances into bins based on predicted confidence and compares the average confidence in each bin to the corresponding average accuracy.

A key issue with binning-based methods is selecting the number of bins. In practice, this choice is often arbitrary, with common values being 10, 15, or 20. However, it has been shown that different choices can significantly affect results and even alter method rankings (Guo et al., 2017; Naeini et al., 2015). When some bins contain only a few samples, the resulting calibration error estimates become imprecise and exhibit high variance. This issue can be partially mitigated by equal-size binning, where each bin contains the same number of samples. On the other hand, using too few bins introduces a smoothing effect, which may obscure important calibration characteristics.

To address the problem of selecting the number of bins, the method $\text{ECE}_{\text{sweep}}$ (Roelofs et al., 2022) has been proposed. This approach increases the number of bins until the estimated error stops increasing monotonically across bins. However, this introduces an implicit bias by assuming a monotonically increasing calibration map, which may not hold in practice. Another limitation of ECE is its inability to

accurately capture the true calibration map. The binning procedure often fails to follow the underlying calibration curve, as illustrated in Figure 12a. Despite these limitations, ECE remains a widely used method for calibration evaluation. This is largely due to its simplicity and ease of interpretation. When combined with reliability diagrams, it provides a useful qualitative overview of model calibration.

In the next section, we discuss a fundamental challenge in calibration evaluation: the absence of a ground-truth calibration map.

4.3. Fit-on-test

The standard practice in machine learning is to split the data into training, validation, and test sets. The training set is used to fit the model, while the validation set is used for model selection, including hyperparameter tuning and architecture selection. The test set is then used to obtain an unbiased estimate of model performance. To ensure a fair evaluation, it must remain separate from the training and model-selection process, thereby preventing data leakage and overfitting to the test data.

The fit-on-test perspective for calibration evaluation arises from the absence of a known ground-truth calibration map, as illustrated in Figure 10. Since this map is unknown, it is typically estimated or approximated before the model’s calibration error can be assessed. Within this framework, an evaluation method first estimates the calibration map from the test data and then measures its deviation from the identity function. The instance-wise calibration error is defined as $\widehat{\text{CE}}(\hat{p}) = \frac{1}{n} \sum_{i=1}^n |\hat{c}(\hat{p}_i) - \hat{p}_i|^\alpha$, where n is the number of instances and $\hat{c}$ denotes the estimated calibration map.

We refer to this approach as the *fit-on-test* view because the calibration map is fitted using the same dataset on which the calibration discrepancy is evaluated. This terminology is intended to highlight an unconventional aspect of calibration evaluation, but it does not imply that the predictive model itself is tuned or improved using the test data. Rather, the test labels are used only to estimate a diagnostic quantity describing the calibration of the evaluated model.

The fit-on-test approach consists of three main steps: (1) estimating the calibration map on the test set, (2) computing instance-wise calibration errors, and (3) aggregating these errors into an overall calibration error estimate. An overview is shown in Figure 10.

The resulting calibration error also admits an intuitive interpretation as the amount of adjustment required to calibrate the model. More precisely, it reflects how far the original predicted probabilities must be moved to match the estimated calibrated probabilities. A larger deviation from the identity map therefore corresponds to a greater required calibration adjustment. This interpretation does not define a separate estimation procedure, since the required adjustment can only be quantified after the calibration map has been estimated; rather, it provides an intuitive view of the resulting calibration discrepancy.

The reliability of the resulting estimate depends on the choice of function class

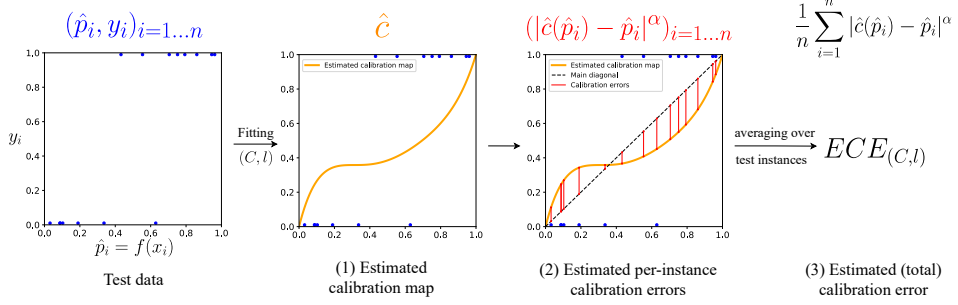


Figure 10: Pipeline of fit-on-test estimation of calibration error. Adapted from Publication II.

used to represent the calibration map. A class that is too restrictive may be unable to capture the true calibration relationship and may therefore introduce systematic bias. Conversely, an excessively flexible class may overfit the test data, particularly when the sample size is limited, leading to an unstable calibration-error estimate. The selected function class should therefore reflect both the expected structure of the calibration map and the amount of available data. Quantifying the bias caused by misspecification of the function class remains an important direction for future research.

One alternative interpretation is to estimate the calibration map on a validation set and then evaluate its deviation from the identity function on a separate test set. This is a valid procedure, but it estimates a slightly different quantity. The resulting discrepancy depends not only on the model’s calibration on the test distribution, but also on how well the calibration-map estimator fitted on the validation set generalises to that distribution. In contrast, the fit-on-test formulation estimates the calibration map directly on the distribution for which the calibration discrepancy is being measured.

In practice, the difference between these procedures may be small when the validation and test distributions are sufficiently similar, and the calibration-map estimator generalises well. Nevertheless, fitting on validation data includes the generalisation error of the estimated calibration map, whereas fitting on test data aims to estimate the calibration discrepancy of the test distribution directly.

Fitting on test data is difficult to avoid in calibration evaluation because the test set does not provide access to the true calibration map required to compute calibration error directly. Consequently, existing methods generally estimate some form of calibration relationship from the same data on which the error is evaluated. At first glance, ECE appears to avoid the fit-on-test paradigm because it operates on bins rather than explicitly fitting a calibration function. However, each bin implicitly estimates the calibrated probability as the average accuracy within that bin, which is then compared with the average confidence. Thus, ECE can also be interpreted as a fit-on-test method. In Publication II, we show that ECE corresponds to a fit-on-test estimator with slope-1 bin tops, where each bin is

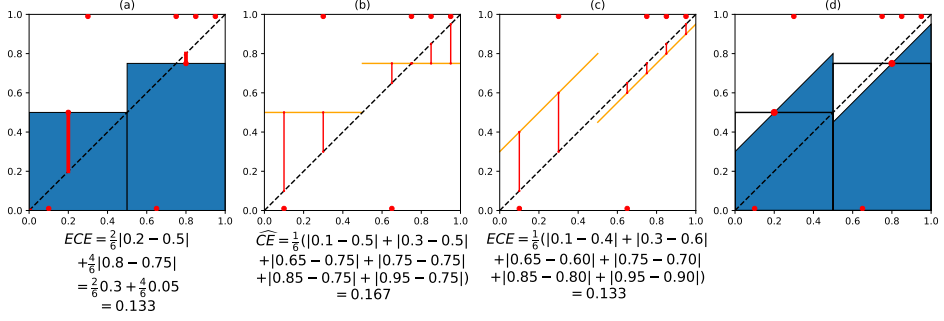


Figure 11: A synthetic example demonstrating ECE as a fit-on-test estimator of calibration error. (a) Reliability diagram for a test set of six instances, yielding $ECE = 0.133$; (b) piecewise-constant estimator with $CE = 0.167$; (c) piecewise slope-1 estimator recovering the same $ECE = 0.133$ as in (a) while visualising per-instance errors; and (d) the proposed *reliability diagram with diagonal filling*, combining (a) and (c). Adapted from Publication II.

represented by a rising linear segment with slope one, that is, a 45° line. Instead of measuring the distance between average accuracy, represented by flat bin tops, and average confidence, one can reinterpret the bin tops as having slope 1 and compute deviations at the instance level. Figure 11 shows that this formulation yields the same score as the original ECE.

Having established that ECE can be interpreted as a fit-on-test evaluation method, it can be improved by selecting the number of bins through cross-validation, as illustrated in Figure 12a. The number of bins is selected using proper scoring rules, such as log-loss or the Brier score, rather than estimated calibration error. This reduces the risk of choosing a binning scheme that is either too coarse to capture the calibration structure or too flexible for the available sample size. Moreover, to the best of our knowledge, most existing approaches estimate at least part of the calibration relationship from the test data. For example, methods based on binning use the test labels to estimate empirical class frequencies within each bin, then compare them with the corresponding predicted probabilities.

Building on the equivalence between ECE with slope-1 bin tops and the original ECE, we propose a modified reliability diagram with diagonal filling. This representation makes deviations from the diagonal easier to assess visually and facilitates comparison between the estimated and true calibration maps when the latter is available. Figure 12a illustrates reliability diagrams with diagonal filling, together with the outlines of the corresponding flat-top bins for reference.

4.4. Piecewise linear method for estimating the calibration map

ECE exhibits several limitations (Figure 12a) that hinder its ability to accurately approximate the true calibration mapping: (1) it produces a discontinuous mapping,

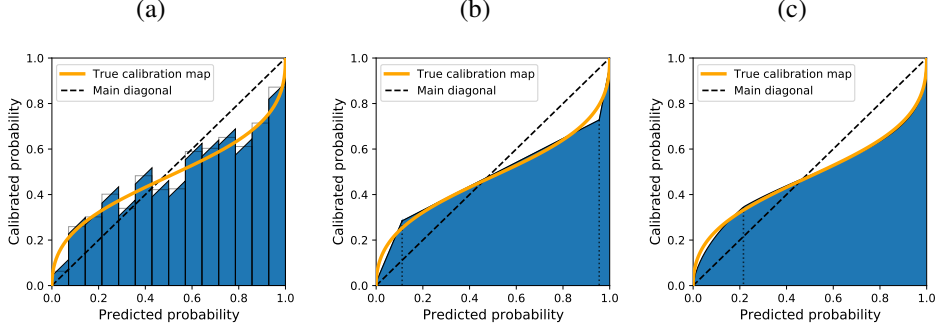


Figure 12: Different reliability diagrams on synthetic data, with the number of bins or segments selected via cross-validation. Subfigure (a) shows a reliability diagram with diagonal filling using 14 bins; (b) a piecewise linear reliability diagram with 3 segments; and (c) a piecewise linear reliability diagram in logit–logit space with 2 segments. Adapted from Publication II.

introducing artificial “jumps” that are unlikely to be present in true calibration maps, and (2) slope-1 bin tops cannot represent calibration maps with different slopes. We propose that the desired family of calibration functions should satisfy the following criteria: (1) continuity, (2) inclusion of the identity function, (3) sufficient flexibility to approximate arbitrary calibration curves, and (4) a limited number of parameters to reduce the risk of overfitting. Since ECE already satisfies some of these criteria, we build upon it by first allowing bin tops with varying slopes, rather than restricting them to slope 1, and then enforcing continuity by connecting adjacent bin tops. This results in a piecewise linear family of calibration functions.

Our proposed method, piecewise linear (PL) (Figure 12b), is sufficiently simple to limit overfitting while remaining flexible enough to approximate a wide range of calibration-map shapes. The PL method consists of b linear segments connected at their endpoints, forming a piecewise linear function. In contrast to ECE, this representation avoids flat bin tops and can therefore approximate the underlying calibration map more accurately. Each linear segment can be interpreted as a bin characterised by its slope and height. Since adjacent segments are connected, the function can be parameterised using $b + 1$ height parameters, denoted by $\mathbf{H}$, corresponding to the values at the segment boundaries. In addition, $b - 1$ parameters are required to define the internal bin boundaries $\mathbf{B}$, since the first and last boundaries are fixed at 0 and 1, respectively. In total, the model requires $2b$ parameters, where b is the number of bins, or linear segments.

The number of bins b remains a hyperparameter of the PL method. To select an appropriate value, we use cross-validation to balance flexibility against overfitting. Figures 12a and 12b compare the standard ECE and the proposed PL methods on the same dataset, with the number of bins selected by cross-validation in both

cases. The results indicate that PL produces a smoother and more natural curve that follows the underlying calibration map more closely.

The number of bins could, in principle, be learned jointly with the bin boundaries and heights. However, b is a discrete structural parameter, whereas the boundaries and heights are continuous parameters, resulting in a mixed discrete–continuous and generally non-smooth optimisation problem. Such an approach would also require a complexity penalty or model-selection criterion to prevent the selection of an unnecessarily large number of bins. This could be addressed, for example, through an explicit model-selection criterion or a differentiable relaxation that suppresses unnecessary candidate segments during optimisation. Developing and evaluating such approaches remains a direction for future work.

The piecewise linear reliability diagram can be used analogously to a standard reliability diagram derived from ECE for assessing model calibration. Furthermore, the piecewise linear function obtained by PL can serve directly as a calibration map $\hat{c}$ for computing CE.

PL is implemented using a neural-network formulation to enable flexible and efficient training. The architecture consists of two components: a binning layer and an interpolation layer.

The main advantage of the neural-network formulation is that it enables the partition and the piecewise linear map to be optimised jointly using the same objective. In a fixed-tree approach, the gating structure would typically be predetermined or learned separately from the calibration values. The neural-network formulation instead provides a modular and differentiable implementation in which the loss function, input representation, output constraints, bin-width restrictions, and gating mechanism can be modified while retaining end-to-end gradient-based optimisation. Similar extensions may be possible in a tree-based framework, but would generally require a different optimisation procedure.

The binning layer takes the predicted probability $\hat{p}$ as input and computes $b + 1$ gating values corresponding to the bin boundaries. Each neuron in this layer is associated with a boundary B_i and evaluates whether $\hat{p}$ lies to the left or right of that boundary, using $L_i = \hat{p} - B_i$ and $R_i = B_i - \hat{p}$. The binning layer is parameterised by b real-valued parameters $\mathbf{Z} = (z_1, z_2, \dots, z_b)$, which define the bin boundaries. These parameters are transformed using a softmax function σ_{sm} to obtain bin widths, which are then cumulatively summed to produce ordered bin boundaries. The interpolation layer contains $b + 1$ parameters representing the boundary heights, denoted by $\mathbf{H}$, which are passed through a logistic function to constrain their values.

Given the gating information from the binning layer, the calibrated probability $\hat{p}'$ is computed by interpolating between the two neighbouring heights H_k and H_{k+1} :

$$\hat{c} = \sum_{k=1}^b g(L_{k+1}, R_k) \left(H_k \frac{L_{k+1}}{L_{k+1} + R_k} + H_{k+1} \frac{R_k}{L_{k+1} + R_k} \right),$$

where the gating function is defined as $g(L_{k+1}, R_k) = \mathbb{I}[(L_{k+1} > 0) \wedge (R_k > 0)]$.

This hard gating function identifies the exact interval containing $\hat{p}$ and therefore preserves the intended piecewise linear structure. A possible alternative would be to use smooth, locally supported basis functions that assign differentiable weights to neighbouring intervals. Such a formulation could improve the stability of joint optimisation of the bin locations and boundary heights. However, the resulting map would only approximate a piecewise linear function near the learned boundaries. The choice therefore involves a trade-off between exact piecewise linearity and potentially smoother end-to-end optimisation, and smooth gating remains a possible direction for future work.

We further propose PL3 (Figure 12c), a piecewise linear calibration method defined in logit–logit space. The method is designed to retain the simplicity and favourable properties of temperature scaling while allowing a more flexible calibration map. Temperature scaling corresponds to a single linear transformation in logit space, whereas PL3 generalises this idea by using multiple learned linear segments. This produces a more flexible transformation in probability space while keeping the model relatively constrained.

PL3 is also related to beta calibration, although the two methods impose different structural assumptions. Beta calibration derives its shape from a parametric transformation involving the logarithms of the predicted probability and its complement, whereas PL3 represents the calibration map using a small number of linear segments in logit–logit space. Beta calibration can be interpreted as producing a mapping with two characteristic pieces, while PL3 allows an arbitrary number of segments and can therefore approximate a broader range of local calibration patterns. Figure 13 illustrates these differences by comparing temperature scaling, beta calibration, and PL3.

A detailed description of the implementation of PL and PL3 is provided in Publication II.

4.5. Experiments and results

The previous chapters introduced the challenges posed by the absence of ground truth and the fit-on-test paradigm. One possible approach is to use synthetic datasets generated from a known calibration map, for which the true CE is available. However, such calibration maps may not reflect the relationships encountered in real-world data. To bridge this gap, we use a pseudo-real dataset for calibration evaluation, which forms the basis of the main results; Publication II also reports results on fully synthetic and real datasets.

In this thesis, *pseudo-real data* refers to generated data designed to preserve realistic characteristics while providing substantially more observations than a conventional real-world dataset. The term does not imply that the generated observations are statistically identical to real data. Rather, the large sample size enables the calibration map to be estimated with much greater precision, providing

(a)

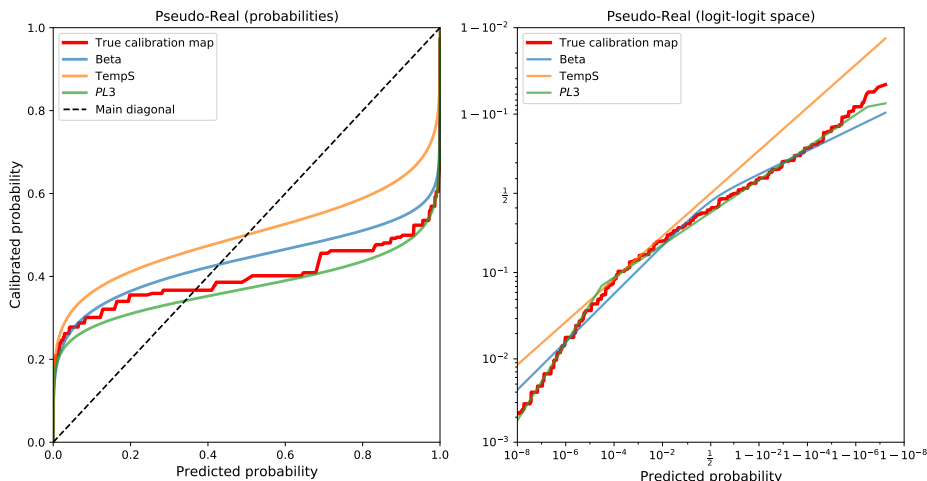


Figure 13: Motivation for PL3. Results are compared in probability space (left column) and logit–logit space (right column) for the ResNet110 (He et al., 2015) model on the CIFAR-5m dataset for the “cats vs. rest” task. Adapted from Publication II.

a close reference approximation against which calibration-error estimators can be compared.

For the pseudo-real evaluation, we use the CIFAR-5m dataset (Nakkiran et al., 2020), which contains more than five million generated images resembling those in CIFAR-10 (Krizhevsky, 2012). Although the images are generated by a generative model, they retain many characteristics of the original dataset, and models trained on CIFAR-5m achieve performance comparable to models trained on CIFAR-10. The combination of realistic image structure and a very large sample size makes the dataset suitable for large-scale calibration analysis.

For our experiments, we train three convolutional neural networks: ResNet110 (He et al., 2015), WideNet32 (Zagoruyko and Komodakis, 2016), and DenseNet40 (Huang, Liu, et al., 2016). Each model is trained on a subset of 45,000 instances sampled from CIFAR-5m, following the training procedures described in the respective original works.

To approximate the true calibration map, we fit an isotonic regression model using one million data instances, while reserving the remaining data for evaluation. Alternative methods for estimating the calibration map were also tested and achieved comparable performance. Although CIFAR-5M is a 10-class dataset, we focus on binary classification settings, as this simplifies visualisation and comparison of calibration results. Specifically, we consider three one-vs-rest tasks (cars vs. rest, cats vs. rest, and dogs vs. rest), as well as a top-label prediction task based on model confidence.

In the experiments, we consider three types of evaluation tasks. First, calibration map estimation, which assesses how closely a method approximates the underlying calibration map; in other words, how accurately it reconstructs the reliability diagram. This is particularly relevant when model predictions are used for downstream decision-making by humans or automated systems. Second, calibration error estimation, which evaluates how accurately a method estimates the overall calibration error. This provides a summary measure of the model’s reliability. Third, calibrator ranking, which aims to determine which calibration method performs best. This task is essential when selecting among competing calibration methods, as it requires reliable comparative evaluation.

The task of evaluating calibration map estimation is studied using two complementary approaches: (1) post-hoc calibration method evaluation and (2) calibration evaluation method assessment.

To compare post-hoc calibration methods, we use three convolutional models, ResNet110 (He et al., 2015), WideNet32 (Zagoruyko and Komodakis, 2016), and DenseNet40 (Huang, Liu, et al., 2016), trained on 45k samples from CIFAR-5m and calibrated on an additional 5k samples. The training setup follows the procedure described by Guo et al. (2017). We consider a range of widely used post-hoc calibration methods, including temperature scaling (TempS), vector scaling (VecS), matrix scaling with off-diagonal and intercept regularisation (MSODIR), Dirichlet calibration with ODIR regularisation (DirODIR), spline calibration (Spline), and order-invariant intra order-preserving functions (IOP). In addition, we include binary calibration methods such as Platt scaling (Platt), isotonic regression (Isotonic), beta calibration (Beta), scaling-binning (ScaleBin), ECE-based binning (ES_{sweep}) with equal-size binning (ES), and the proposed piecewise linear methods (PL and PL3).

The results show that, although the model predictions are initially uncalibrated, PL3 provides the closest approximation to the underlying calibration map, making it the most effective post-hoc calibration method (see Table 6). This evaluation identifies which methods best correct poorly calibrated model outputs.

To assess the quality of calibration evaluation methods, we consider the outputs of the six best-performing post-hoc calibration methods (Beta, VecS, Platt, PL3, ScaleBin, and Isotonic). This setting focuses on cases where predictions are already

Table 6: Evaluation of post-hoc calibrators based on $|\hat{c} - c^*| (\times 10^3)$ in approximating the true calibration map for DenseNet40, ResNet110, and WideNet32 models on CIFAR-5m. Each calibrator is trained on 5000 instances, with results averaged over the three classifiers. Adopted from Publication II.

	Platt	Beta	Isotonic	ScaleBin	TempS	VecS	dirODIR	MSODIR	Spline	IOP	ES_{sweep}	PL3	PL
cars vs rest	3.6 ₂	4.3 ₇	4.0 ₅	4.1 ₆	7.3 ₁₂	3.8 ₄	3.8 ₃	4.3 ₈	6.7 ₁₀	8.2 ₁₃	7.1 ₁₁	2.9₁	4.9 ₉
cats vs rest	6.8 ₂	15.0 ₁₂	9.0 ₄	9.5 ₅	14.0 ₁₁	10.4 ₆	13.3 ₉	12.9 ₈	8.8 ₃	13.5 ₁₀	19.3 ₁₃	5.3₁	12.3 ₇
dogs vs rest	4.5 ₂	14.4 ₁₂	6.8 ₄	6.7 ₃	11.7 ₁₁	8.6 ₅	11.6 ₁₀	11.5 ₉	8.7 ₆	11.0 ₈	25.5 ₁₃	3.5₁	10.6 ₇
confidence	13.1 ₄	17.4 ₇	11.6 ₃	8.7 ₂	22.7 ₈	34.2 ₁₁	38.1 ₁₃	37.5 ₁₂	15.4 ₆	24.8 ₁₀	23.9 ₉	8.5₁	15.4 ₅

close to being calibrated. The task then measures the discrepancy between the estimated calibration map and the reference calibration map.

We compare a range of calibration evaluation methods, including post-hoc calibrators and the proposed PL-based methods. As shown in Table 7, PL achieves the best performance, followed closely by PL3, with beta calibration ranking third.

Table 7: Evaluation of calibration estimators for reliability diagrams based on $|\hat{c} - c^*| (\times 10^3)$ on CIFAR-5m. Unlike Table 6, this comparison is performed on predictions that have already undergone post-hoc calibration. Adopted from Publication II.

		ES ₁₅	ES _{sweep}	ES _{CV}	PL3	PL	Platt	Beta	Isotonic	Spline	IOP
	All	9.47 ₉	8.62 ₇	8.52 ₆	7.27 ₂	6.27₁	8.39 ₅	7.6 ₃	9.2 ₈	9.55 ₁₀	7.81 ₄
Initial model	ResNet110	9.43 ₈	9.04 ₆	9.22 ₇	7.33 ₂	6.7₁	8.9 ₅	7.97 ₃	9.66 ₉	10.15 ₁₀	8.64 ₄
	DenseNet40	9.98 ₁₀	8.59 ₇	8.37 ₆	7.08 ₃	5.98₁	7.93 ₅	7.35 ₄	9.15 ₈	9.42 ₉	6.92 ₂
	WideNet32	9.0 ₉	8.22 ₆	7.97 ₅	7.39 ₂	6.14₁	8.33 ₇	7.49 ₃	8.8 ₈	9.08 ₁₀	7.86 ₄
Data size	1000	13.34 ₁₀	11.44 ₆	11.49 ₇	10.23 ₄	8.02₁	10.88 ₅	9.63 ₃	12.99 ₉	12.08 ₈	8.17 ₂
	3000	8.46 ₈	7.64 ₆	8.19 ₇	6.64 ₂	6.22₁	7.44 ₅	6.8 ₃	8.65 ₉	9.24 ₁₀	7.43 ₄
	10000	6.61 ₆	6.77 ₇	5.88 ₃	4.93 ₂	4.59₁	6.85 ₈	6.38 ₅	5.97 ₄	7.34 ₉	7.81 ₁₀
Task	cars vs rest	4.07 ₇	4.61 ₉	3.82 ₄	3.58 ₃	2.74₁	4.21 ₈	3.13 ₂	3.9 ₆	4.84 ₁₀	3.82 ₅
	cats vs rest	10.45 ₁₀	10.15 ₉	9.24 ₅	7.44 ₂	6.64₁	9.82 ₇	9.02 ₄	9.6 ₆	10.11 ₈	8.61 ₃
	dogs vs rest	9.17 ₉	8.75 ₇	8.67 ₆	7.33 ₃	5.8₁	7.81 ₅	7.41 ₄	8.98 ₈	9.3 ₁₀	6.63 ₂
	confidence	14.19 ₉	10.96 ₄	12.35 ₇	10.72 ₂	9.91₁	11.71 ₅	10.84 ₃	14.34 ₁₀	13.95 ₈	12.16 ₆

The second task focuses on estimating the ECE score as accurately as possible, i.e., quantifying the deviation of the calibration map from the diagonal. The results show that ECE with 15 equal-size bins performs best, followed closely by PL3 and beta calibration (Table 8). ECE is debiased, which may give it an advantage, although it is less effective at estimating the calibration map.

The third task evaluates the ability to rank six calibrators, with the goal of identifying the method that produces the most accurate ranking. The isotonic fit-on-test evaluator achieves the best performance in this task (Table 8).

Table 8: Evaluation of calibration estimators for true calibration error, measured by $|ECE - CE| (\times 10^3)$, together with the ranking of six calibrators according to rankcorrel(ECE) on CIFAR-5m. Adapted from Publication II.

Metric	Data	ES ₁₅	ES _{sweep}	ES _{CV}	PL3	PL	Platt	beta	isotonic	Spline	IOP
$ ECE - CE $	one-vs-rest	2.3₁	3.24 ₅	3.66 ₈	2.59 ₂	2.87 ₃	3.63 ₇	2.9 ₄	3.8 ₉	4.16 ₁₀	3.48 ₆
	confidence	4.54 ₂	5.14 ₄	6.37 ₇	4.73 ₃	5.24 ₅	5.83 ₆	4.39₁	7.2 ₉	7.39 ₁₀	6.96 ₈
rankcorrel(ECE)	one-vs-rest	0.457 ₃	0.195 ₇	0.443 ₄	0.36 ₆	0.513 ₂	0.01 ₉	0.128 ₈	0.552₁	0.398 ₅	-0.043 ₁₀
	confidence	0.624 ₂	0.434 ₆	0.563 ₃	0.406 ₇	0.51 ₅	0.018 ₉	0.511 ₄	0.657₁	0.392 ₈	-0.095 ₁₀

4.6. Summary and limitations

In conclusion, this work addresses the problem of evaluating calibration error in the absence of a ground-truth calibration map. Since such a map is typically unavailable in real-world datasets, all calibration evaluation methods must implicitly estimate it before computing calibration error. We formalise this perspective as the *fit-on-test* paradigm, which enables the use of standard techniques, such as cross-validation, to improve model fitting and generalisation. We further introduce a pseudo-real dataset approach that enables evaluation in a more realistic setting than purely synthetic data. Using a large-scale dataset with millions of instances, we approximate the calibration map by fitting on one million samples and evaluating on the remaining data. In addition, we propose a new family of calibration evaluation methods based on piecewise linear functions (PL and PL3). These methods extend ECE by introducing continuity and flexible slopes, allowing for more accurate approximation of calibration maps. Empirically, PL3 achieves the best performance for post-hoc calibration, while PL performs best in approximating the calibration map for already well-calibrated predictions. For calibration error estimation, ECE with 15 equal-size bins performs best, followed by PL3 and beta calibration. For ranking calibration methods, isotonic calibration provides the most accurate results.

The main limitation of this work is that the evaluation is conducted on a single pseudo-real dataset. While this provides useful insights for deep convolutional networks, it does not necessarily generalise to other model classes or datasets. Although this setting offers a strong baseline, further investigation of pseudo-real datasets is needed. A second limitation is the focus on binary classification, which simplifies visualisation and interpretation but does not capture the full complexity of multiclass calibration, including classwise and multiclass calibration error. Finally, calibration performance is known to be sensitive to dataset shift and out-of-distribution data. Further research is required to better understand and address these challenges in practical deployment settings.

5. COMBINING UNCERTAINTY INFORMATION OF SEMANTIC SEGMENTATION FOR AUTONOMOUS PERCEPTION SYSTEMS (PUBLICATION III)

This chapter provides an overview of the final contribution of the thesis, addressing **RQ3**: *What uncertainty information supports reliable decision-making in an autonomous perception system?*

The preceding chapters examined how predicted probabilities can be calibrated and how calibration can be evaluated. This chapter applies these ideas to spatial uncertainty estimates in autonomous driving. In this setting, the output of a perception system is not useful only as a segmentation map; it must also be transformed into uncertainty representations that support planning and risk-aware decision-making. These derived uncertainty quantities may themselves be uncalibrated, creating a direct connection to the methodological problems studied in the previous chapters.

The primary goal of a perception system is to reason about its surroundings, for example, the environment of an autonomous vehicle (AV). Such a system may detect objects either to avoid them or to use them for navigation, such as lane markings; this can be described as an object-based view. Alternatively, the autonomous system may segment the surrounding environment to determine the category, such as object, road, or sidewalk, to which each point or grid cell in the perception area belongs; this is referred to as a grid-based view.

This work proposes a framework that combines grid-based and object-based representations to account for multiple sources of uncertainty required for reliable and safe navigation. To this end, we employ the bird’s-eye-view (BEV) semantic segmentation model *FIERY* (Hu et al., 2021), which produces a grid-based representation with instance-level information. The proposed framework, using *FIERY* as a representative example, is described in detail in the following sections. Finally, a downstream planning task is used to evaluate whether incorporating additional uncertainty information improves the safety and reliability of trajectory planning.

5.1. Motivation

Self-driving vehicles are becoming increasingly prevalent; however, their safety and reliability are not guaranteed. A commonly reported metric for evaluating such systems is the number of kilometres, or miles, driven without interventions or collisions. Collisions may arise from several interacting factors, including perception errors, inaccurate localisation, failures to predict the behaviour of other road users, unsuitable trajectory planning, violations of traffic rules, inappropriate speed selection, and unusual road or environmental conditions.

The contribution of this chapter focuses specifically on perception uncertainty and its use in risk-aware planning. Reliable information about where the road is occupied is a prerequisite for subsequent decision-making, but it does not by

itself address the complete autonomous-driving problem. Instead, the proposed framework should be understood as one component within a broader system that also includes localisation, behaviour prediction, planning, and control.

Within the perception component, missed object detections are particularly important because an undetected road user may not be considered by the planner. Even when an object is detected, the system may need to infer not only its location but also its likely behaviour or intention.

To reduce collision risk, the system must therefore identify when its predictions about road conditions or surrounding objects are uncertain or imprecise. The grid-based view, which consists of cells containing class predictions, such as road, vehicle, or pedestrian, is effective for representing spatial occupancy and object locations; however, it lacks explicit information about instance-level grouping across cells.

While nearby segmented grid cells could be aggregated for object detection, this is challenging because neighbouring cells are strongly correlated, as illustrated in Figure 14. Since adjacent cells share information about the presence of an object, it is difficult for a planner to obtain a reliable object-level uncertainty estimate for a region.

Conversely, in an object-based view, the perception system detects individual objects and must decide which detections to retain as true positives. This typically requires a detection threshold, below which detections are discarded. Such discarded detections may nevertheless contain informative signals, potentially leading to missed objects.

Planners currently tend to use either no uncertainty information or only limited uncertainty information (Reda et al., 2024). The proposed approach therefore has the potential to improve the safety and reliability of trajectory planning. However, as demonstrated in our experiments, the resulting uncertainty estimates are often not well calibrated and consequently require additional calibration.

These limitations motivate combining object-based and grid-based representations. In addition, we incorporate information about potentially undetected objects through a region occupancy query (ROQ). This allows the planner to query the probability that a given region, such as the road ahead or a crosswalk, is unoccupied, thereby supporting safer and more informed trajectory planning.

5.2. Uncertainty-aware perception system

This chapter proposes a framework that provides the necessary components for an uncertainty-aware perception system, highlighting the importance of integrating multiple sources of uncertainty to ensure safe and reliable operation.

An uncertainty-aware perception model should incorporate both an object-based view, enriched with uncertainty information about each object’s shape, size, location, and trajectory, and a grid-based view, represented as occupancy maps indicating the likelihood that a given grid cell is occupied. In addition, the

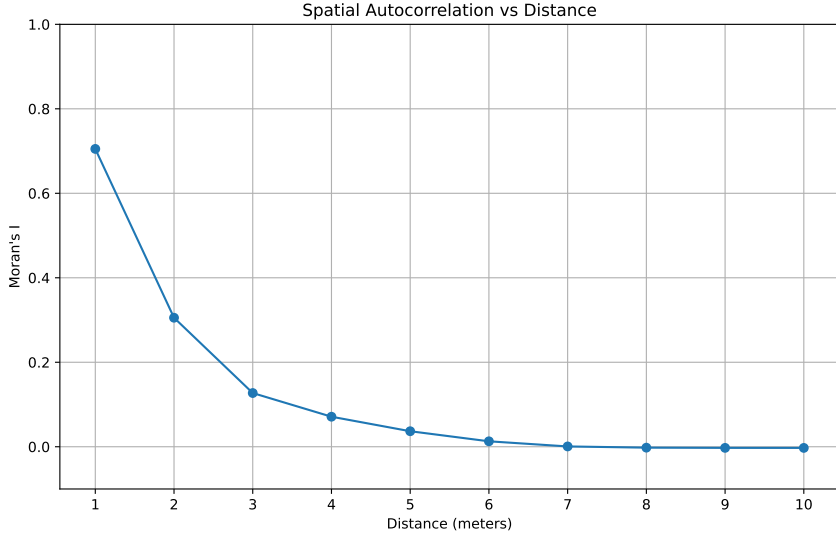


Figure 14: Dependence of grid-cell correlation on Euclidean distance in metres. For each value of D , correlations are evaluated using pairs of cells with distances greater than $D - 1$ and at most D . Results are derived from BEV semantic-segmentation outputs of the FIERY model. Adapted from Publication III.

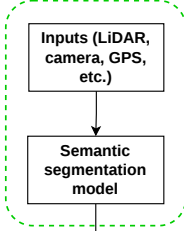
framework enables the planner to query the likelihood of occupancy within specific regions of interest, which we define as a region occupancy query (ROQ).

More precisely, the proposed uncertainty-aware perception model outputs the following:

1. A list of detected objects (object-based view) with associated uncertainty information:
 - *Presence uncertainty*: the likelihood that an object exists at a given location;
 - *Localisation uncertainty*: a probability distribution over possible object locations;
 - *Shape uncertainty*: a probability distribution over possible object shapes;
 - *Trajectory uncertainty*: a probability distribution over possible future trajectories.
2. An occupancy map (grid-based view) representing the probability that the grid cell is occupied.
3. Region occupancy query (ROQ): a query mechanism that provides uncertainty estimates over a specified region (e.g., multiple grid cells within an area of interest).

The proposed framework is illustrated in Figure 15, which provides an overview of the uncertainty-aware perception pipeline.

Preliminaries



Our proposed uncertainty framework

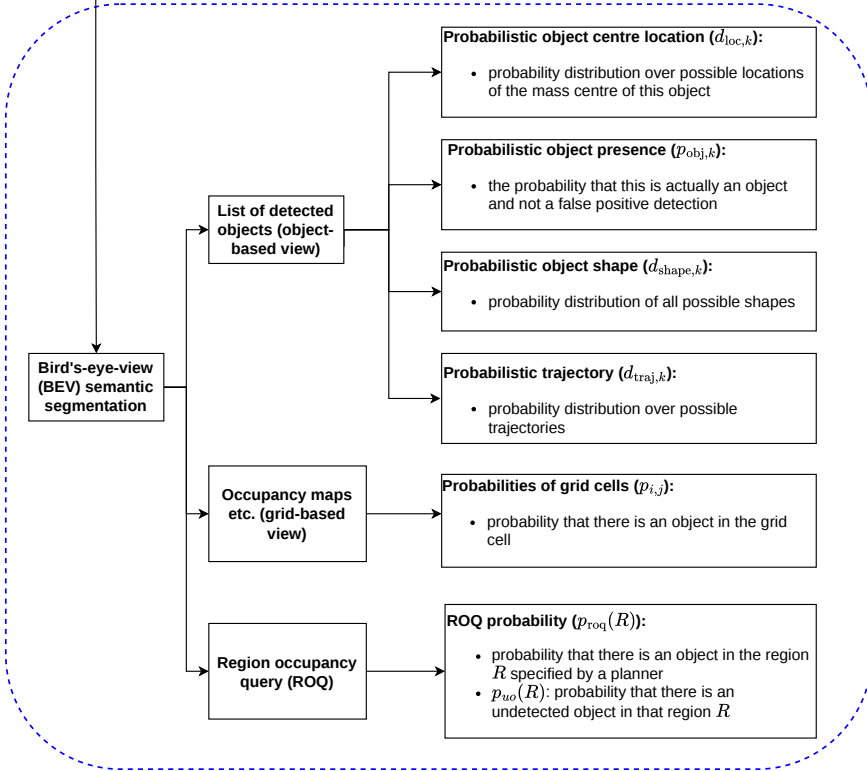


Figure 15: Overview of the proposed uncertainty framework, illustrating the uncertainties extracted from BEV semantic segmentation. The blue dashed outline highlights the scope of this work. Adapted from Publication III.

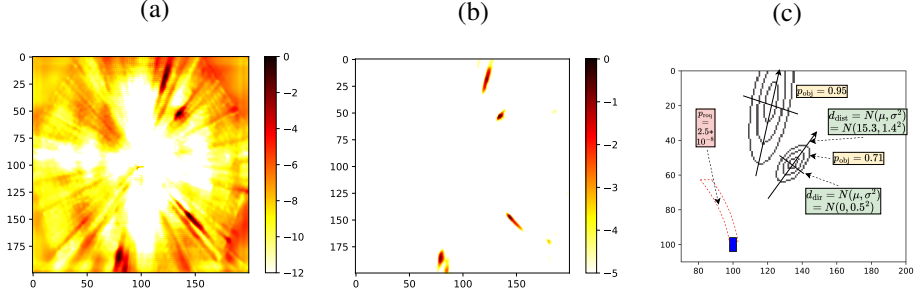


Figure 16: Bird’s-eye-view (BEV) perception uncertainty based on the FIERY model. Subfigures (a) and (b) show FIERY semantic segmentation in log-scale (natural logarithm) for improved visibility, clipped at -12 and -5, respectively. Subfigure (c) illustrates the proposed additional information on object locations, represented as ellipsoids, along with an example region occupancy query (ROQ) region indicated by the dashed red outline. The blue rectangle at (100, 100) denotes the ego-vehicle. Note that subfigure (c) is zoomed in. Adapted from Publication III.

Figure 16 presents semantic segmentation outputs with and without thresholding, alongside the proposed uncertainty representations, including location and presence uncertainty, as well as uncertainty related to undetected objects based on ROQ.

The BEV semantic segmentation model produces uncertainty estimates at the pixel level, indicating whether each pixel belongs to an object (Figures 16a and 16b). The output is represented as a 200×200 grid, where each cell corresponds to a $0.5 \text{ m} \times 0.5 \text{ m}$ area. The values in Figures 16a and 16b are log-scaled for improved visibility.

Figure 16a is clipped at -12 , highlighting the prevalence of low-probability regions alongside localised areas of higher probability. Figure 16b is clipped at -5 , effectively applying a detection threshold, as low-probability regions cannot be reliably interpreted as detected objects. However, pixel-wise predictions alone are insufficient, as it is necessary to determine which pixels belong to the same object in order to derive meaningful uncertainty estimates for object location, shape, and presence (Figure 16c).

Figure 16c illustrates the output of the proposed framework, including presence and localisation uncertainty (represented as ellipsoids), as well as the probability of missed detections derived from the proposed ROQ, shown as a dashed red region. These uncertainty representations are essential for enabling a comprehensive, uncertainty-aware planning or decision-making system.

5.3. Methods: extracting uncertainty estimates from the BEV semantic segmentation

This section describes how to obtain the proposed uncertainty estimates for both grid-based and object-based representations, incorporating additional region occupancy queries (ROQ). In this example, we employ the BEV semantic segmentation model FIERY, which processes six camera images to produce a 200×200 grid representing the surroundings of the ego vehicle. The model generates a top-down (bird’s-eye-view) representation and outputs binary semantic predictions in the form of occupancy probabilities, indicating the likelihood of a vehicle being present in each grid cell. However, the proposed approach is not limited to the FIERY model; any BEV semantic segmentation model that outputs an occupancy probability matrix can be used. Formally, we assume an output of the form $\hat{P} = (\hat{p}_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$, where $\hat{p}_{ij} \in [0, 1]$ denotes the probability of object presence in the grid cell with coordinates (i, j) .

The following subsections describe our approach to extracting uncertainty estimates, specifically for object centre location, object presence, and the ROQ probability. Shape and trajectory uncertainties are omitted from this chapter to ensure a concise presentation; the reader is referred to Publication III for further details and methods addressing these aspects.

5.3.1. Probabilistic object centre location

The probabilistic object centre location representation, denoted by d_{loc} , provides a compact approximation of the spatial distribution of occupancy evidence associated with detected objects. The overall approach is to fit a two-dimensional Gaussian Mixture Model (GMM) to locations sampled from the semantic-segmentation output. A GMM is chosen because the spatial distribution of segmentation probabilities often exhibits an approximately ellipsoidal structure.

First, we estimate the number of detected objects, which determines the number of mixture components n_{nodes} . For this purpose, we use the semantic-segmentation output $(\hat{p}_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$ and retain grid cells whose probabilities exceed a threshold p_{thresh} . These high-probability cells are clustered to identify candidate object centres using the same clustering approach as the FIERY model (Hu et al., 2021) employs for instance segmentation.

Next, we perform probability-based stratified sampling to construct the dataset used for fitting the GMM. Each grid cell at coordinates (i, j) is included according to its predicted occupancy probability. The replication count is defined as

$$c_{ij} = \text{round}(m_{\text{thresh}} \cdot \hat{p}_{ij}),$$

where $m_{\text{thresh}} = 1/p_{\text{thresh}}$. Thus, a grid cell whose probability equals the threshold is included once. Lowering p_{thresh} increases the number of sampled locations and may also increase the estimated number of objects.

Finally, a two-dimensional GMM with n_{nodes} components is fitted to the sampled locations. The resulting mixture approximates the spatial distribution of location conditional on the predicted occupancy mass.

More precisely, the representation is an equally weighted mixture

$$d_{\text{loc}}(\mathbf{x}) = \frac{1}{K} \sum_{k=1}^K \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k),$$

where $K = n_{\text{nodes}}$, and μ_k and Σ_k denote the mean and covariance matrix of component k . The mixture weights are fixed to $1/K$ and are not estimated from the data. The pixel-level probabilities affect the component means and covariances through the sampling procedure, but differences in total occupancy mass between components are not retained as separate mixture weights.

The component means of the GMM represent centres of occupancy mass, while the covariance matrices describe the spatial dispersion of the associated evidence. This dispersion may reflect uncertainty in object-centre location, as well as in object extent, shape, orientation, and localisation. These effects are not separately identifiable from a single occupancy map and are therefore captured jointly by the fitted covariance.

This representation differs from the Gaussian mixture probability hypothesis density (PHD) filter (Vo and Ma, 2006), whose intensity function, defined within the random-finite-set framework, integrates over a region to give the expected number of targets in that region. In contrast, d_{loc} is normalised to unit mass, and the estimated object count is represented separately by K . It also differs from Markov random field occupancy models (Rachlin et al., 2005), which explicitly model statistical dependencies between neighbouring occupancy states. The covariance matrices used here instead describe the spatial dispersion of sampled occupancy evidence rather than dependence between cell states.

Furthermore, the quality of the representation depends on the estimated number of components. If too few components are selected, occupancy evidence from multiple objects may be merged into a single Gaussian. This may be especially problematic when the objects have different trajectories or become distinguishable only after the occlusion period resolves. Models that explicitly represent uncertainty in the number of objects, such as random-finite-set or probability-hypothesis-density formulations, provide a possible direction for future work.

Algorithm 1 summarises the procedure for generating the spatial location representation.

5.3.2. Probabilistic object presence

The probabilistic object presence p_{obj} is computed based on the object centre location distribution d_{loc} for each detected object (n_{nodes}). The distribution d_{loc} represents the likelihood of the object centre being located in different grid cells, thereby spreading the probability mass across multiple cells.

Algorithm 1: Probabilistic object centre location. Adapted from Publication III.

Input: $(\hat{p}_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$;
 p_{thresh} ; /* Threshold for potential cluster centres */
 $m_{\text{thresh}} = \frac{1}{p_{\text{thresh}}}$;
Calculation:
 Estimate the number of potential cluster centres n_{nodes} using p_{thresh} .
 Set $c_{ij} = \text{round}(m_{\text{thresh}} \hat{p}_{ij})$ and construct a spatial sample by replicating each grid-cell location (i, j) according to c_{ij} .
 Fit an equally weighted GMM with n_{nodes} components to the sampled locations.
Return: d_{loc} with n_{nodes} components.

Since an object typically occupies multiple grid cells, a reliable estimate of its presence also requires incorporating the expected object size, denoted by S_{exp} . To this end, the probabilistic object presence p_{obj} is computed by aggregating the probability mass over grid cells that lie within the region covering 99% of the Gaussian distribution, denoted as $d_{\text{loc},99\%}$. The aggregated value is then normalised based on the expected object size S_{exp} .

The choice of the 99% probability mass reflects a trade-off between capturing the majority of the object's spatial uncertainty and excluding low-probability regions that are more likely to correspond to noise. This ensures that most plausible object locations are considered while limiting the influence of the Gaussian distribution's long tails.

Algorithm 2: Probabilistic object presence. Adapted from Publication III.

Input: $(\hat{p}_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$;
 d_{loc} ; /* Location distribution */
 S_{exp} ; /* expected size of a typical object */
Calculation:
 Calculate $p_{\text{obj}} = \frac{\sum_{i,j \in d_{\text{loc},99\%}} \hat{p}_{i,j}}{S_{\text{exp}}}$; /* where $d_{\text{loc},99\%}$ means 99% of Gaussian probability mass. */
 Clip p_{obj} to the range $[0, 1]$.
Return: p_{obj}

Summing marginal probabilities across spatially dependent grid cells does not account for their joint dependence structure and may therefore yield values exceeding one. Moreover, calibrated pixel-level probabilities do not necessarily remain calibrated after aggregation to the object level. Although the Gaussian mixture representation captures spatial structure, it does not resolve this issue. Consequently, the resulting object-level probability may require separate calibration.

Due to the spatial dependence between neighbouring grid cells (see Figure 14) and potential inaccuracies in the assumed object size S_{exp} , the resulting probability

estimate may exceed valid bounds. Therefore, in practice, the final object presence probability p_{obj} is clipped to the interval $[0, 1]$.

Algorithm 2 provides an overview of the procedure for estimating probabilistic object presence.

5.3.3. Probabilities of region occupancy queries

The probabilistic region occupancy query (ROQ) p_{roq} represents the likelihood that a region R is occupied, i.e., that there is an undetected object present within it. The formulation accounts for both detected and potentially undetected objects, and is defined as follows:

$$p_{\text{roq}}(R) = 1 - (1 - p_{\text{uo}}(R)) \prod_{k=1}^{n_{\text{obj}}} (1 - p_{\text{obj},k} \cdot p_{\text{overlap},k}(R)).$$

The key idea is to estimate the probability that no objects occupy the region and to derive the probability of occupancy as its complement. This probability is composed of contributions from both detected and undetected objects. For detected objects, we use the object presence probability $p_{\text{obj},k}$, obtained as described in the previous section, and combine it with an overlap factor $p_{\text{overlap},k}(R)$, which quantifies the proportion of the object distribution overlapping with the region R .

To account for undetected objects, we estimate $p_{\text{uo}}(R)$ using the probabilistic semantic segmentation output. Specifically, we exclude grid cells that are likely associated with detected objects by removing those overlapping with the 99% probability mass of their location distributions $d_{\text{loc},99\%}$. This defines the residual region $R \setminus d_{\text{loc},99\%}$. The probability of undetected objects is then computed as

$$p_{\text{uo}}(R) = \frac{\sum_{(i,j) \in R \setminus d_{\text{loc},99\%}} \hat{p}_{ij}}{S_{\text{exp}}},$$

where $\hat{p}_{ij}$ denotes the occupancy probability of grid cell (i, j) and S_{exp} is the expected object size. The summation used to obtain $p_{\text{uo}}(R)$ provides an approximate region-level aggregation and may require clipping and separate calibration.

Finally, the detected- and undetected-object contributions are combined to estimate the probability that the region is empty, from which $p_{\text{roq}}(R)$ is obtained. This definition of $p_{\text{roq}}(R)$ assumes independent combination of detected- and undetected-object contributions. Dependencies between objects, overlapping location distributions, or shared pixel-level evidence may violate this assumption. However, this formulation provides a unified region-level uncertainty estimate that can be enriched with additional information, such as object shape and trajectory uncertainty, to support safer and more informed planning.

5.4. Experiments

This section describes the model and dataset used for extracting uncertainty estimates. Additionally, we demonstrate that the uncertainties are not well-calibrated

out of the box, using estimated calibration error (ECE) and reliability diagrams as evaluation metrics. Finally, we apply calibration methods and analyse how calibration affects a downstream planning task.

The uncertainty estimates are obtained using the BEV semantic segmentation model FIERY (Hu et al., 2021), retrained for the pedestrian class. The original model was trained to detect vehicles; however, due to its strong performance and the limited variability in the vehicle class, it proved difficult to reliably evaluate uncertainty quantification. Due to the increased difficulty of pedestrian classification, the model is retrained on pedestrians, using the same training procedure as described in the original FIERY setup (Hu et al., 2021).

The model is trained on the NuScenes (Caesar et al., 2020) dataset, which contains diverse driving scenarios collected under varying conditions in Boston and Singapore. For evaluation, we use standalone validation sequences, each approximately 20 seconds (40 frames) in duration.

In all experiments, we compare the following setups:

- Uncalibrated (uncal): raw grid-cell occupancy probabilities produced by the model.
- Pixel-wise calibrated (pw-cal): the grid-based semantic segmentation is calibrated at the pixel level, and object-level probabilities are derived directly from the calibrated outputs.
- Object-wise calibrated (obj-cal): uncertainty estimates are calibrated at the object level (e.g., location, presence), while the underlying semantic segmentation remains uncalibrated.

This comparison highlights the importance of calibration and illustrates the impact of calibration at different levels.

5.4.1. Results on probabilistic grid-based view

In our experiments, the grid-based representation is obtained from BEV semantic segmentation, which produces occupancy probabilities $\hat{P} = (\hat{p}_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$. For consistency with the semantic segmentation literature, each grid cell is referred to as a pixel.

The dataset is split into 20% for validation (approximately 41 million pixels) and 80% for testing (approximately 164 million pixels). Calibration is performed using isotonic regression (Zadrozny and Elkan, 2002), applied independently at the pixel level. Consequently, a single calibration mapping is learned and applied uniformly to all pixels at test time. Isotonic regression is chosen due to its efficiency and suitability for large-scale datasets.

Figure 17 presents a comparison between uncalibrated and pixel-wise calibrated reliability diagrams. The calibrated model exhibits improved alignment with the diagonal, indicating better calibration. This improvement is also reflected in the evaluation metrics: the estimated calibration error (ECE) decreases from 0.00038 (uncalibrated) to 0.000032, representing an approximately tenfold reduction, while

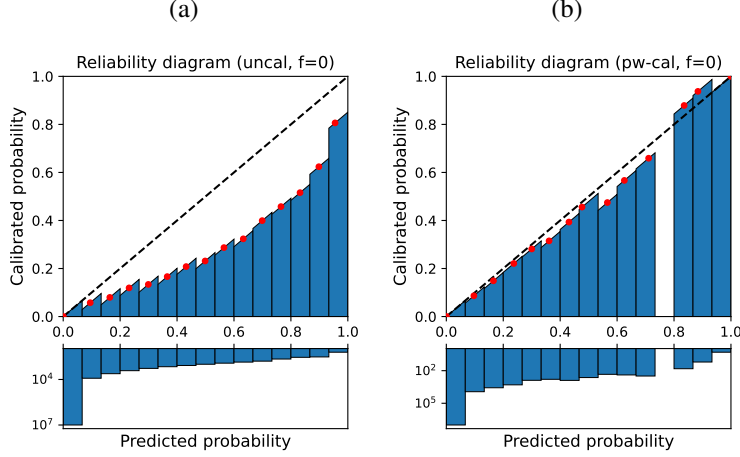


Figure 17: Reliability diagrams illustrating pixel-wise semantic segmentation calibration. Subfigure (a) corresponds to the uncalibrated model, and subfigure (b) shows pixel-wise calibration with equal-width binning into 15 bins. Adapted from Publication III.

the negative log-likelihood (NLL) decreases from 0.00356 to 0.00329.

5.4.2. Results on detected and undetected objects

In this subsection, we describe the calibration approach for object centre location, object presence, and undetected object uncertainty within a region.

The **object centre location** distribution d_{loc} is extracted using the method described in Algorithm 1. A detection threshold of $p_{\text{thresh}} = 0.01$ is used, with $m_{\text{thresh}} = 100$ controlling the sampling process for fitting GMMs. The dataset includes 30 validation sequences with 2,153 detections and 120 test sequences with 10,099 detections. The location uncertainty is initially modelled using a two-dimensional Gaussian distribution. For calibration, we decompose the distribution into two independent one-dimensional components, distance error and directional error, and apply quantile remapping (Kuleshov et al., 2018) to each component. Figure 18 presents a quantile–quantile plot for regression calibration (Kuleshov et al., 2018). For perfectly calibrated predictions, the curve should align with the diagonal, analogous to reliability diagrams. Object-wise calibration successfully improves both directional and distance error calibration. In contrast, pixel-wise calibration produces results similar to the uncalibrated case, indicating that pixel-level calibration does not effectively transfer to object-level uncertainty.

The **object presence** probabilities p_{obj} are derived from the location distribution d_{loc} , as outlined in Algorithm 2. Instead of modelling a probabilistic object shape, we assume a fixed expected size for pedestrians, $S_{\text{exp}} = 5$, as reliable shape uncertainty estimates were not available. In total, the dataset comprises 30 validation

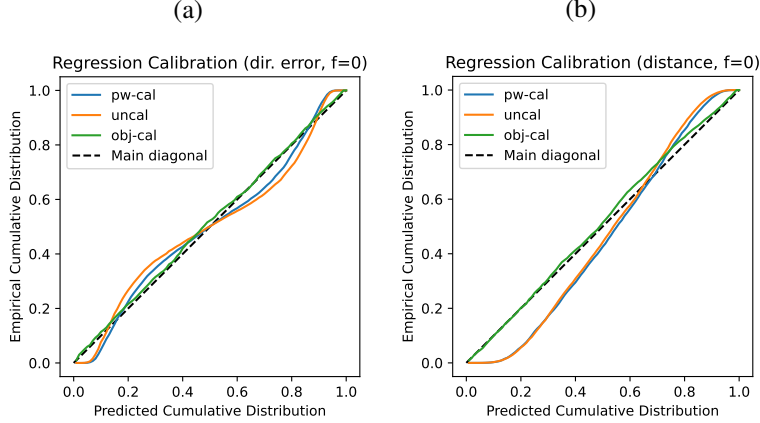


Figure 18: Regression calibration plots (Kuleshov et al., 2018) illustrating object location probabilities. Subfigure (a) corresponds to direction error and subfigure (b) to distance. Results are shown for uncalibrated (uncal), pixel-wise calibrated (pw-cal), and object-wise calibrated (obj-cal) settings. The black dashed line denotes perfect calibration. Adapted from Publication III.

sequences containing 3,066 objects and 120 test sequences containing 14,214 objects. A predicted detection is labelled as a true positive if the corresponding ground-truth object lies within the 99% probability mass of the predicted object centre location. This yields binary ground-truth labels in $\{0, 1\}$ paired with predicted object presence probabilities p_{obj} , which are used as input to the calibration method. For calibration, we employ the beta calibration method (Kull, Silva Filho, et al., 2017), as it is well-suited for limited data scenarios. Figure 19 shows that object-wise calibration achieves the best calibration performance, reducing the estimated calibration error (ECE) from 0.185 to 0.016 compared to the uncalibrated results.

The **probability of undetected objects** in a region, p_{uo} , is estimated similarly to p_{obj} , with the key difference that it is evaluated over a predefined safety-critical region R . In our experiments, R is defined as a $10 \times 20\text{m}$ area in front of the ego vehicle. To avoid overlap with detected objects, the region is refined by excluding areas corresponding to the 99% probability mass of their Gaussian location distributions, resulting in a residual region used to estimate the probability of undetected objects. Beta calibration (Kull, Silva Filho, et al., 2017) is applied to calibrate p_{uo} . The results indicate that object-wise calibration achieves the best performance, reducing the estimated calibration error (ECE) from 0.011 (uncalibrated) to 0.005. However, these results should be interpreted with caution, as the test set contains only 29 undetected instances across 4,091 frames. A larger dataset would be required to draw more definitive conclusions.

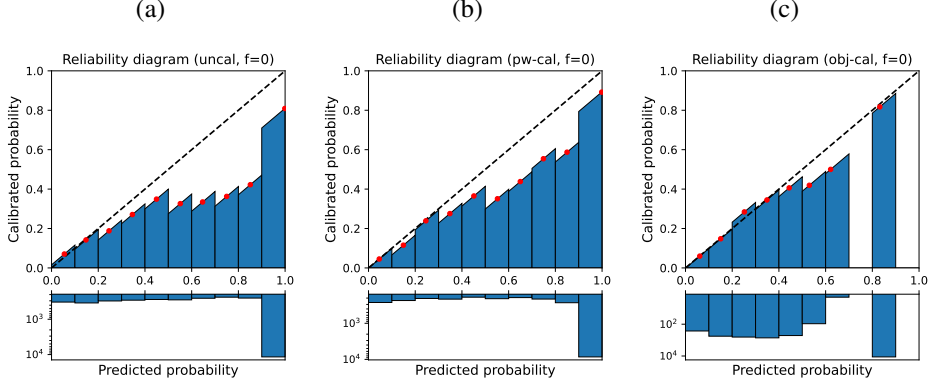


Figure 19: Reliability diagrams illustrating object presence probabilities. Sub-figures (a), (b), and (c) correspond to uncalibrated, pixel-wise calibrated, and object-wise calibrated cases, respectively. The histograms beneath the diagrams display the number of elements in each bin, with a logarithmic y-axis for better comparison. Adapted from Publication III.

5.4.3. Downstream task

A downstream task is used to evaluate the benefit of incorporating additional uncertainty information. Specifically, we consider a simplified trajectory planning task. We predefine 25 candidate trajectories and evaluate their safety by checking for potential collisions over four future timesteps (corresponding to a total horizon of 2 seconds). For each trajectory, collision likelihood is estimated using three different setups:

1. Uncalibrated semantic segmentation: raw BEV occupancy probabilities are summed along each trajectory.
2. Pixel-wise calibrated semantic segmentation: calibrated grid-cell probabilities are used to estimate collision likelihood.
3. region occupancy query (ROQ) and detection probabilities: object-wise calibrated location and presence uncertainties are combined with the probability of undetected objects within the region.

For each setup, ego-vehicle-width corridors are constructed along each trajectory, and the corresponding regions are used to evaluate collision probabilities.

To assess trajectory-ranking performance, we use the area under the ROC curve (AUC) (Fawcett, 2006). The quality of the predicted trajectory probabilities is evaluated using the Brier score (BS) and negative log-likelihood (NLL). The results in Table 9 show that all metrics improve when pixel-wise calibrated probabilities are used, with further improvements obtained from object-level calibration and region-occupancy information.

To provide a practical interpretation of these results, we also examined the num-

ber of potential pedestrian collisions associated with the highest-ranked trajectory. Across 4019 sequences and 25 candidate trajectories per sequence, the full set of candidates contained 5019 potential collisions. The trajectories selected using uncalibrated, pixel-wise calibrated, and object-calibrated ROQ estimates resulted in 51, 48, and 52 potential collisions, respectively. In some sequences, none of the candidate trajectories may be collision-free, meaning that the potential collision could only be avoided by stopping rather than by selecting an alternative trajectory from the candidate set. These values are similar, indicating that the improvements in NLL, BS, and AUC do not yet translate into a clear reduction in collisions in this simplified experiment. Pixel-wise calibration avoided three conflicts relative to the uncalibrated setup, whereas the ROQ-based method produced one additional conflict. Further experiments with more varied and realistic planning scenarios are therefore required to determine whether the improved probability estimates yield practically meaningful safety benefits.

Thus, although the object-calibrated ROQ method provides the best probabilistic and ranking performance, the collision-count analysis suggests that the practical differences between the methods remain inconclusive in the current setting.

5.5. Summary and limitations

In conclusion, this work proposes a framework for a more reliable and safe autonomous perception system by combining grid-based and object-based probabilistic representations. The object-based component consists of a set of detected objects, each associated with uncertainty estimates for object centre location, presence, shape, and trajectory. In addition, a region occupancy query (ROQ) is introduced to estimate the likelihood of collision within a specified region. The proposed approach demonstrates how these uncertainty estimates can be obtained from a bird’s-eye-view semantic segmentation model, namely FIERY, trained on pedestrian data. We show that the extracted probabilities are not well calibrated out of the box, but can be effectively calibrated using simple post-hoc calibration methods. Furthermore, the results from a downstream planning task indicate that incorporating calibrated uncertainty estimates leads to more reliable and safer trajectory planning.

The main limitation of this work is that the framework is evaluated only in a

Table 9: Evaluation of trajectory estimation performance for three approaches: uncalibrated (uncal), pixel-wise calibrated (pw-cal), and object-level calibration with region occupancy query probabilities (ROQ (obj-cal)). Best results are indicated in bold. Adapted from Publication III.

	NLL	BS	AUC
uncal	0.605926	0.034923	0.895200
pw-cal	0.536661	0.033925	0.903931
ROQ (obj-cal)	0.269738	0.032463	0.904931

simplified planning scenario, which does not fully capture the complexity of real-world applications. A more comprehensive evaluation would require comparison with existing uncertainty-aware planning approaches, integration with alternative perception systems, and direct assessment of the value of calibrated uncertainties in more realistic trajectory-planning tasks. Secondly, the modelling of probabilistic object shape remains largely hypothetical because of limitations in the available data and model capabilities; in practice, a fixed pedestrian size is assumed. Similarly, probabilistic trajectory estimation depends on the predictive capabilities of the model and is not evaluated against alternative approaches based on motion dynamics, such as velocity and direction distributions. Thirdly, several components of the framework rely on approximations. In particular, probability aggregation and object-presence estimation use simplifying assumptions, including independence and fixed object size, that may not hold in practice. Finally, the framework is validated using a single dataset and model, and its performance may vary across datasets and architectures. At the time of the study, suitable datasets combining surround-camera data with the required annotations were limited. Evaluation on additional datasets and perception models is therefore needed to assess the generality of the conclusions. The calibration analysis is also limited to ECE and simple post-hoc calibration methods. ECE was selected because it is widely recognised and provides a familiar baseline for readers in the autonomous-driving domain. The primary aim of Publication III was to introduce the uncertainty representation and demonstrate that the resulting quantities were not adequately calibrated, rather than to compare calibration-error estimators comprehensively. Nevertheless, evaluating the same uncertainties using PL3 and other calibration-error estimators would help determine how sensitive the conclusions are to the selected evaluation method and remains an important direction for future work. In addition, the framework is not optimised for real-time performance, which may limit its applicability in real-world autonomous-driving systems. Further evaluation across datasets, models, calibration-error estimators, and realistic planning scenarios is therefore necessary before deploying the framework more broadly.

6. CONCLUSION

In conclusion, machine learning is increasingly used to support critical decision-making, underscoring the need to develop trustworthy and reliable models. Such models should provide probabilistic predictions that quantify uncertainty, and these probabilities must be well-calibrated to accurately reflect true outcome frequencies. This thesis focuses on improving classifier calibration by introducing a novel calibration method and a new evaluation paradigm. Furthermore, it demonstrates how incorporating richer uncertainty information can lead to more reliable and trustworthy systems. This chapter provides a broader overview of the implications and limitations of the work, as well as directions for future research.

The thesis is situated at the intersection of probabilistic classification, post-hoc calibration, calibration evaluation, and uncertainty-aware decision-making. The background chapters present the relevant literature together with the technical concepts required for the contributions, while the contribution chapters extend this literature in three connected stages. First, the thesis develops calibration maps for correcting multiclass probabilities. Second, it studies how such maps and the resulting calibration errors can be estimated reliably. Third, it applies these ideas to uncertainty representations used in an autonomous-driving perception and planning system. This progression connects the methodological development of calibration methods with their evaluation and practical use.

RQ1: *Can we improve multiclass calibration for deep learning models?* In **Publication I**, we propose a natively multiclass calibration method, Dirichlet calibration, which operates on probabilistic inputs and provides some degree of interpretability. While temperature scaling is a widely used baseline for neural network calibration, we show that it fails to adequately correct classwise miscalibration, highlighting the need for more expressive methods. However, more flexible methods are prone to overfitting, and standard L2 regularisation is often insufficient. To address this, we introduced a novel regularisation technique, ODIR, which regularises off-diagonal and intercept terms while leaving diagonal terms unregularised. Experimental results demonstrate that Dirichlet calibration with ODIR regularisation performs particularly well on datasets with a large number of classes (e.g., 100 or more). Furthermore, matrix scaling with ODIR regularisation achieves slightly better performance than Dirichlet calibration in neural network settings. This improvement can be attributed to the use of logits as inputs in matrix scaling, whereas Dirichlet calibration operates on log-transformed probabilities, which may result in information loss. The main limitation of this work is that it does not evaluate performance under distribution shift, and neural network experiments are restricted to convolutional neural networks.

RQ2: *Why is calibration evaluation a difficult problem to solve, and how can we improve evaluation methods?* In **Publication II**, we show that accurate calibration evaluation requires access to the true calibration map, which is generally unavailable in real-world datasets. Consequently, evaluation methods implicitly

estimate the calibration map before computing calibration error, making them inherently *fit-on-test* procedures. This perspective allows calibration evaluation to be framed as a model-fitting problem, enabling the use of techniques such as cross-validation to improve estimation. We further demonstrate that commonly used metrics such as ECE also fall into the fit-on-test category, despite their binning-based formulation. To address limitations of standard ECE, including fixed binning and flat bin-top assumptions, we introduce a piecewise linear (PL) calibration evaluation method capable of approximating a wider range of calibration maps. Additionally, we propose a variant operating in logit-logit space (PL3), inspired by temperature scaling and beta calibration. To evaluate calibration methods more rigorously, we employ synthetic datasets with known calibration maps, as well as pseudo-real datasets where the calibration map is approximated using a large number of samples. For this purpose, we use the CIFAR-5m dataset (Nakkiran et al., 2020), which contains five million CIFAR-like images. This setup enables evaluation of calibration map estimation, calibration error estimation, and method ranking. Experimental results indicate that PL and PL3 methods are effective for estimating calibration maps, while debiased ECE and isotonic regression perform better for calibration error estimation and ranking, followed by PL-based methods. The main limitations of this work are that it relies on a single pseudo-real dataset and focuses primarily on binary classification.

RQ3: *What uncertainty information supports reliable decision-making in an autonomous perception system?* In **Publication III**, we propose a framework for incorporating uncertainty into autonomous perception systems. The framework represents detected objects with probabilistic attributes, including presence, location, shape, and trajectory. In addition, we introduce region occupancy query (ROQ), which estimates the probability of occupancy within a specified region by accounting for both detected and potentially undetected objects. This is particularly important, as detection systems typically discard low-probability predictions, potentially missing relevant objects. We demonstrate how these uncertainty estimates can be extracted from a probabilistic bird’s-eye-view semantic segmentation model, such as FIERY, and provide guidance for applying the approach to similar models. The extracted probabilities are not well-calibrated by default; however, we show that they can be effectively calibrated using simple post-hoc methods. Finally, we evaluate the framework on a downstream planning task, showing that incorporating both calibrated object-level and region-level uncertainty improves decision-making. This highlights the importance of combining multiple sources of uncertainty for safe and reliable autonomous systems. The work integrates insights from calibration methods, calibration evaluation, and uncertainty modelling, and applies them to the domain of autonomous driving. The main limitations of this work are that the evaluation is limited to a simplified planning scenario and a single model. While the framework provides a principled approach to uncertainty representation, further validation across diverse models, datasets, and real-world scenarios is required.

Taken together, the three contributions address complementary parts of the same uncertainty pipeline. A calibration method is useful only if its effect can be evaluated reliably, while calibrated probabilities become practically meaningful only when they are represented and incorporated into downstream decisions. The thesis, therefore, connects probability correction, calibration evaluation, and application-level uncertainty use.

6.1. Trends and opportunities for future research

The publications in this thesis primarily focus on vision classifiers, their uncertainty estimates, and the estimation and correction of calibration errors. However, the field has recently shifted from convolutional neural networks towards vision transformers and large-scale foundation models. Despite this transition, the core challenges remain unchanged: obtaining well-calibrated predictions and understanding when model outputs can be trusted.

Traditional calibration methods typically apply a single global calibration mapping to all data points. However, this approach is often insufficient in practice. Recent work therefore explores data-dependent calibration strategies, such as proximity-based calibration (Xiong et al., 2024), which partitions data based on a similarity metric, and sample-wise approaches, such as CalAttn (Liang et al., 2025). These methods exploit additional information from the data and model to improve calibration under varying conditions. Furthermore, proximity calibration is complemented by proximity-informed expected calibration error (PIECE), which incorporates proximity information into calibration error evaluation.

Beyond convolutional models, calibration methods are increasingly applied to vision transformers. Recent studies (Xiong et al., 2024; Liang et al., 2025) evaluate calibration performance across transformer-based architectures, demonstrating that calibration challenges persist across model families.

Another important research direction involves foundation models, including large language models (LLMs) and vision-language models (VLMs). LLMs are known to exhibit overconfidence and hallucinations, particularly in low-knowledge regimes (Kadavath et al., 2022). When access to token probabilities or logits is limited, alternative uncertainty estimation techniques have been proposed, such as verbalised confidence (S. Lin et al., 2022) and semantic entropy (Kuhn et al., 2023). When internal model representations are available, standard post-hoc calibration methods can be applied, alongside more advanced techniques such as adaptive temperature scaling (ATS) (Xie et al., 2024), which leverages hidden state information to compute token-specific calibration parameters.

For VLMs, calibration can often be formulated as a classification problem, enabling the use of standard post-hoc calibration techniques (Tu et al., 2024). In settings where probability outputs are not directly accessible, alternative approaches such as calibrated test-time prompt tuning (C-TPT) (Yoon et al., 2024) can be employed.

Despite these developments, the contributions presented in Chapters 3–5 highlight several limitations and opportunities for future work. A key limitation in calibration research is the lack of datasets with known ground-truth calibration maps. While synthetic datasets provide controlled settings for evaluation, they may not accurately reflect real-world calibration behaviour. Pseudo-real datasets offer a compromise, but remain limited in scope. Therefore, developing comprehensive benchmarks that combine synthetic, pseudo-real, and real-world datasets would enable more robust evaluation and comparison of calibration methods. Such benchmarks would support practitioners in selecting methods appropriate for specific models and applications.

A second direction concerns improving the piecewise linear (PL) calibration method. While PL is highly effective at approximating calibration maps, it performs comparatively worse in calibration error estimation and ranking tasks. Understanding this discrepancy is an important open problem. Potential directions include incorporating debiasing techniques, similar to those used in ECE, or conducting broader evaluations across diverse datasets to better characterise its behaviour.

A third direction involves studying robustness under distribution shift. It remains unclear how well the proposed calibration method (i.e., Dirichlet calibration) maintains performance under distribution shifts, and whether the proposed calibration evaluation metrics (PL, PL3) remain reliable in such settings. Addressing these questions would improve the reliability of calibration methods in real-world deployment scenarios.

In addition, Chapter 5 introduces a framework for uncertainty-aware perception in autonomous systems. The current evaluation is limited to a single dataset, a single model, and a simplified downstream task. Future work could extend this framework by integrating it into more advanced planning systems or developing new planning approaches that explicitly incorporate uncertainty estimates. Evaluating the framework across diverse models and datasets would provide further insight into its generalisability. Moreover, comparisons with state-of-the-art planning methods, including those that do not explicitly model uncertainty, would help quantify its practical benefits.

Overall, the contributions of this thesis advance the development of calibration methods, calibration evaluation techniques, and uncertainty-aware perception frameworks. By improving both the estimation and utilisation of uncertainty, this work contributes to the broader goal of building more reliable, trustworthy, and interpretable machine learning systems for decision-making.

BIBLIOGRAPHY

- Vaicenavicius, Juozas, Widmann, David, Andersson, Carl, Lindsten, Fredrik, Roll, Jacob, and Schön, Thomas (2019). “Evaluating model calibration in classification”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 3459–3467.
- Zagoruyko, Sergey and Komodakis, Nikos (2016). “Wide Residual Networks”. In: *CoRR* abs/1605.07146. arXiv: 1605.07146.
- He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian (2015). “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385. arXiv: 1512.03385.
- Kuleshov, Volodymyr, Fenner, Nathan, and Ermon, Stefano (2018). “Accurate uncertainties for deep learning using calibrated regression”. In: *International conference on machine learning*. PMLR, pp. 2796–2804.
- Kull, Meelis, Perello Nieto, Miquel, **Kängsepp, Markus**, Silva Filho, Telmo, Song, Hao, and Flach, Peter (2019). “Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration”. In: *Advances in Neural Information Processing Systems* 32, pp. 12295–12305.
- Kängsepp, Markus**, Valk, Kaspar, and Kull, Meelis (2025). “On the usefulness of the fit-on-test view on evaluating calibration of classifiers”. In: *Machine Learning* 114.4, p. 105. DOI: 10.1007/s10994-024-06652-6.
- Kängsepp, Markus** and Kull, Meelis (2026). “Calibrating Perception Uncertainty for Autonomous Driving”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 34.05, pp. 621–645. DOI: 10.1142/S0218488526500212.
- Vasser, Madis, **Kängsepp, Markus**, Magomedkerimov, Murad, Kilvits, Kälver, Stafinjak, Vladislav, Kivisik, Taavi, Vicente, Raul, and Aru, Jaan (2017). “VREX: an open-source toolbox for creating 3D virtual reality experiments”. In: *BMC psychology* 5.4, pp. 1–8. DOI: 10.1186/s40359-017-0173-4.
- Ingel, Anti, Shahroudi, Novin, **Kängsepp, Markus**, Tättar, Andre, Komisarenko, Viacheslav, and Kull, Meelis (2020). “Correlated daily time series and forecasting in the M4 competition”. In: *International Journal of Forecasting* 36.1, pp. 121–128. DOI: 10.1016/j.ijforecast.2019.02.018.
- Sabir, Ahmed, **Kängsepp, Markus**, and Sharma, Rajesh (2026). “The Confidence Trap: Gender Bias and Predictive Certainty in LLMs”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 40.46, pp. 39173–39181. DOI: 10.1609/aaai.v40i46.41265.
- Del, Maksym, **Kängsepp, Markus**, Domnich, Marharyta, Tampuu, Ardi, Yankovskaya, Lisa, Kull, Meelis, and Fishel, Mark (2026). *How Uncertainty Estimation Scales with Sampling in Reasoning Models*. arXiv: 2603.19118 [cs.AI].

- Tu, Weijie, Deng, Weijian, Campbell, Dylan, Gould, Stephen, and Gedeon, Tom (2024). “An empirical study into what matters for calibrating vision-language models”. In: *arXiv preprint arXiv:2402.07417*.
- Yoon, Hee Suk, Yoon, Eunseop, Tee, Joshua Tian Jin, Hasegawa-Johnson, Mark, Li, Yingzhen, and Yoo, Chang D (2024). “C-tpt: Calibrated test-time prompt tuning for vision-language models via text feature dispersion”. In: *arXiv preprint arXiv:2403.14119*.
- Xie, Johnathan, Chen, Annie S, Lee, Yoonho, Mitchell, Eric, and Finn, Chelsea (2024). “Calibrating language models with adaptive temperature scaling”. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18128–18138.
- Lin, Stephanie, Hilton, Jacob, and Evans, Owain (2022). “Teaching models to express their uncertainty in words”. In: *arXiv preprint arXiv:2205.14334*.
- Kuhn, Lorenz, Gal, Yarin, and Farquhar, Sebastian (2023). “Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation”. In: *arXiv preprint arXiv:2302.09664*.
- Kadavath, Saurav, Conerly, Tom, Askell, Amanda, Henighan, Tom, Drain, Dawn, Perez, Ethan, Schiefer, Nicholas, Hatfield-Dodds, Zac, DasSarma, Nova, Tran-Johnson, Eli, et al. (2022). “Language models (mostly) know what they know”. In: *arXiv preprint arXiv:2207.05221*.
- Liang, Wenhao, Zhang, Wei Emma, Yue, Lin, Xu, Miao, Maennel, Olaf, and Chen, Weitong (2025). “Calibration Attention: Instance-wise Temperature Scaling for Vision Transformers”. In: *arXiv preprint arXiv:2508.08547*.
- Kull, Meelis, Silva Filho, Telmo, and Flach, Peter (2017). “Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers”. In: *Artificial Intelligence and Statistics*. PMLR, pp. 623–631.
- Guo, Chuan, Pleiss, Geoff, Sun, Yu, and Weinberger, Kilian Q (2017). “On calibration of modern neural networks”. In: *International conference on machine learning*. PMLR, pp. 1321–1330.
- Koopman, Philip (2022). *How Safe Is Safe Enough?: Measuring and Predicting Autonomous Vehicle Safety*. Independently published.
- Smirnov, Nikolai V et al. (1939). “On the estimation of the discrepancy between empirical curves of distribution for two independent samples”. In: *Bull. Math. Univ. Moscou* 2.2, pp. 3–14.
- Platt, John (2000). “Probabilities for SV machines”. In: *Advances in Large Margin Classifiers*. Ed. by A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans. MIT Press, pp. 61–74.
- Wang, Cheng (2023). “Calibration in deep learning: A survey of the state-of-the-art”. In: *arXiv preprint arXiv:2308.01222*.
- Silva Filho, Telmo, Song, Hao, Perello-Nieto, Miquel, Santos-Rodriguez, Raul, Kull, Meelis, and Flach, Peter (2023). “Classifier calibration: a survey on how to assess and improve predicted class probabilities”. In: *Machine Learning* 112.9, pp. 3211–3260.

- Begoli, Edmon, Bhattacharya, Tanmoy, and Kusnezov, Dimitri (2019). “The need for uncertainty quantification in machine-assisted medical decision making”. In: *Nature Machine Intelligence* 1.1, pp. 20–23.
- Gal, Yarin and Ghahramani, Zoubin (2016). “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *international conference on machine learning*. PMLR, pp. 1050–1059.
- Der Kiureghian, Armen and Ditlevsen, Ove (2009). “Aleatory or epistemic? Does it matter?” In: *Structural safety* 31.2, pp. 105–112.
- Lakshminarayanan, Balaji, Pritzel, Alexander, and Blundell, Charles (2017). “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *Advances in neural information processing systems* 30.
- Thulasidasan, Sunil, Chennupati, Gopinath, Bilmes, Jeff A, Bhattacharya, Tanmoy, and Michalak, Sarah (2019). “On mixup training: Improved calibration and predictive uncertainty for deep neural networks”. In: *Advances in neural information processing systems* 32.
- Zhang, H, Cissé, M, Dauphin, YN, and Lopez-Paz, D (2018). “mixup: Beyond Empirical Risk Minimization. ICLR”. In: *Vancouver, BC, Canada, April*.
- Neal, Radford M (2012). *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media.
- Zadrozny, Bianca and Elkan, Charles (2001). “Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers”. In: *Icml*. Vol. 1, pp. 609–616.
- Zadrozny, Bianca and Elkan, Charles (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.
- Naeni, Mahdi Pakdaman, Cooper, Gregory, and Hauskrecht, Milos (2015). “Obtaining well calibrated probabilities using bayesian binning”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 29. 1.
- Gupta, Kartik, Rahimi, Amir, Ajanthan, Thalaisyasingam, Mensink, Thomas, Sminchisescu, Cristian, and Hartley, Richard (2020). “Calibration of neural networks using splines”. In: *arXiv preprint arXiv:2006.12800*.
- Zhang, Jize, Kailkhura, Bhavya, and Han, T Yong-Jin (2020). “Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning”. In: *International conference on machine learning*. PMLR, pp. 11117–11128.
- Widmann, David, Lindsten, Fredrik, and Zachariah, Dave (2019). “Calibration tests in multi-class classification: A unifying framework”. In: *Advances in neural information processing systems* 32.
- Kumar, Ananya, Liang, Percy S, and Ma, Tengyu (2019). “Verified uncertainty calibration”. In: *Advances in Neural Information Processing Systems* 32.
- Kumar, Aviral, Sarawagi, Sunita, and Jain, Ujjwal (July 2018). “Trainable Calibration Measures for Neural Networks from Kernel Mean Embeddings”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by

- Jennifer Dy and Andreas Krause. Vol. 80. *Proceedings of Machine Learning Research*. PMLR, pp. 2805–2814.
- Vo, B-N and Ma, W-K (2006). “The Gaussian mixture probability hypothesis density filter”. In: *IEEE Transactions on signal processing* 54.11, pp. 4091–4104.
- Rachlin, Yaron, Dolan, John M, and Khosla, Pradeep (2005). “Efficient mapping through exploitation of spatial dependencies”. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3117–3122.
- Rahimi, Amir, Shaban, Amirreza, Cheng, Ching-An, Hartley, Richard, and Boots, Byron (2020). “Intra order-preserving functions for calibration of multi-class neural networks”. In: *Advances in Neural Information Processing Systems* 33, pp. 13456–13467.
- Lin, Tsung-Yi, Goyal, Priya, Girshick, Ross, He, Kaiming, and Dollár, Piotr (2017). “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988.
- Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jon, and Wojna, Zbigniew (2016). “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.
- Kull, Meelis and Flach, Peter (2015). “Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration”. In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part I* 15. Springer, pp. 68–85.
- Roelofs, Rebecca, Cain, Nicholas, Shlens, Jonathon, and Mozer, Michael C (2022). “Mitigating bias in calibration error estimation”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 4036–4054.
- Hu, Anthony, Murez, Zak, Mohan, Nikhil, Dudas, Sofía, Hawke, Jeffrey, Badrinarayanan, Vijay, Cipolla, Roberto, and Kendall, Alex (2021). “FIERY: Future Instance Prediction in Bird’s-Eye View From Surround Monocular Cameras”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15273–15282.
- Caesar, Holger, Bankiti, Varun, Lang, Alex H, Vora, Sourabh, Liong, Venice Erin, Xu, Qiang, Krishnan, Anush, Pan, Yu, Baldan, Giancarlo, and Beijbom, Oscar (2020). “nusenes: A multimodal dataset for autonomous driving”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631.
- Reda, Mohamed, Onsy, Ahmed, Haikal, Amira Y, and Ghanbari, Ali (2024). “Path planning algorithms in the autonomous driving system: A comprehensive review”. In: *Robotics and Autonomous Systems* 174, p. 104630.
- Niculescu-Mizil, Alexandru and Caruana, Rich (2005). “Predicting good probabilities with supervised learning”. In: *Proceedings of the 22nd international conference on Machine learning*, pp. 625–632.

- Nixon, Jeremy, Dusenberry, Michael W, Zhang, Linchuan, Jerfel, Ghassen, and Tran, Dustin (2019). “Measuring Calibration in Deep Learning.” In: *CVPR workshops*. Vol. 2. 7.
- Xiong, Miao, Deng, Ailin, Koh, Pang Wei W, Wu, Jiaying, Li, Shen, Xu, Jianqing, and Hooi, Bryan (2024). “Proximity-informed calibration for deep neural networks”. In: *Advances in Neural Information Processing Systems* 36.
- Mukhoti, Jishnu, Kulharia, Viveka, Sanyal, Amartya, Golodetz, Stuart, Torr, Philip, and Dokania, Puneet (2020). “Calibrating deep neural networks using focal loss”. In: *Advances in Neural Information Processing Systems* 33, pp. 15288–15299.
- Müller, Rafael, Kornblith, Simon, and Hinton, Geoffrey E (2019). “When does label smoothing help?” In: *Advances in neural information processing systems* 32.
- Hendrycks, Dan and Dietterich, Thomas (2019). “Benchmarking neural network robustness to common corruptions and perturbations”. In: *arXiv preprint arXiv:1903.12261*.
- Ovadia, Yaniv, Fertig, Emily, Ren, Jie, Nado, Zachary, Sculley, David, Nowozin, Sebastian, Dillon, Joshua, Lakshminarayanan, Balaji, and Snoek, Jasper (2019). “Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift”. In: *Advances in neural information processing systems* 32.
- Brier, Glenn W (1950). “Verification of forecasts expressed in terms of probability”. In: *Monthly weather review* 78.1, pp. 1–3.
- Murphy, Allan H and Winkler, Robert L (1977). “Reliability of subjective probability forecasts of precipitation and temperature”. In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 26.1, pp. 41–47.
- Ferro, Christopher AT and Fricker, Thomas E (2012). “A bias-corrected decomposition of the Brier score”. In: *Quarterly Journal of the Royal Meteorological Society* 138.668, pp. 1954–1960.
- Fawcett, Tom (2006). “An introduction to ROC analysis”. In: *Pattern recognition letters* 27.8, pp. 861–874.
- Bröcker, Jochen and Smith, Leonard A. (2007). “Increasing the Reliability of Reliability Diagrams”. In: *Weather and Forecasting* 22.3, pp. 651–661. DOI: 10.1175/WAF993.1.
- Silva Filho, Telmo, Song, Hao, Perelló Nieto, Miquel, Santos-Rodriguez, Raul, Kull, Meelis, and Flach, Peter (May 2023). “Classifier calibration: a survey on how to assess and improve predicted class probabilities”. In: *Machine Learning* 112, pp. 1–50. DOI: 10.1007/s10994-023-06336-7.
- Maher, Mohamed and Kull, Meelis (2021). “Instance-based Label Smoothing For Better Calibrated Classification Networks”. In: *CoRR* abs/2110.05355.
- Moreno-Torres, Jose G, Raeder, Troy, Alaiz-Rodríguez, Rocío, Chawla, Nitesh V, and Herrera, Francisco (2012). “A unifying view on dataset shift in classification”. In: *Pattern recognition* 45.1, pp. 521–530.

- Bishop, Christopher M and Nasrabadi, Nasser M (2006). *Pattern recognition and machine learning*. Vol. 4. 4. Springer.
- Bröcker, Jochen (2012). “Estimating reliability and resolution of probability forecasts through decomposition of the empirical score”. In: *Climate dynamics* 39.3, pp. 655–667.
- Krizhevsky, Alex (May 2012). “Learning Multiple Layers of Features from Tiny Images”. In.
- Netzer, Yuval, Wang, Tao, Coates, Adam, Bissacco, Alessandro, Wu, Bo, and Ng, Andrew Y. (2011). “Reading Digits in Natural Images with Unsupervised Feature Learning”. In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*.
- Huang, Gao, Sun, Yu, Liu, Zhuang, Sedra, Daniel, and Weinberger, Kilian Q. (2016). “Deep Networks with Stochastic Depth”. In: *CoRR* abs/1603.09382. arXiv: 1603.09382.
- Huang, Gao, Liu, Zhuang, and Weinberger, Kilian Q. (2016). “Densely Connected Convolutional Networks”. In: *CoRR* abs/1608.06993. arXiv: 1608.06993.
- LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.
- LeCun, Yann (1998). “The MNIST database of handwritten digits”. In: <http://yann.lecun.com/exdb/mnist/>.
- Cheni, Aaron (2017). *Base pretrained models and datasets in pytorch*. URL: <https://github.com/aaron-xichen/pytorch-playground> (visited on 04/18/2026).
- Wang, Jingcong (2023). *UCI datasets*. DOI: 10.21227/g4y0-sw34.
- Nakkiran, Preetum, Neyshabur, Behnam, and Sedghi, Hanie (2020). “The deep bootstrap framework: Good online learners are good offline generalizers”. In: *arXiv preprint arXiv:2010.08127*.
- Pereyra, Gabriel, Tucker, George, Chorowski, Jan, Kaiser, Łukasz, and Hinton, Geoffrey (2017). “Regularizing neural networks by penalizing confident output distributions”. In: *arXiv preprint arXiv:1701.06548*.

ACKNOWLEDGEMENTS

The story of my PhD begins with my master's thesis, which was supervised by Meelis Kull. He offered me the opportunity to join a PhD program. I was reluctant to agree right away, but my friend from the master's program, Andre Tättar, was also joining, and it was now or never. Here I am, defending my PhD in November 2026. I took a few extra years to reach the finish line, but overall, I am happy with the outcome. I want to thank my supervisor, Meelis, for offering me the opportunity and supporting me throughout the journey.

I want to thank my internal reviewer, Naveed Muhammad, and external reviewers, Ulf Brefeld and Fredrik Lindsten, for their insightful feedback and comments, which helped me strengthen my thesis.

In the first year of my PhD, I also met the love of my life, Annika Kängsepp, who has been my emotional support through the hardships of the PhD and with whom I share the journey, as she started a bit later than I did.

Next, I want to thank my colleagues and friends whom I met throughout the journey. I thank my office mate from the first semester, Mikk Puustusmaa, with whom I had a great time in the office and during lunch breaks. I also thank Andre and Joonas P., the fellow PhD students who started the program with me, for sharing the struggles of the process along the way.

Our research group started small but grew each year, welcoming Mari-Liis, Novin, Viacheslav, Bhawani, Joonas, Mihkel, Rasmus, Karl-Kaspar, Marek, Heili, Halil, Laura, Julius, Urmas, and Oskar. We have had, and will continue to have, a great time together, enjoying the sauna, playing FIFA and basketball, and writing papers.

Also, I want to thank members from other research groups for chatting, solving puzzles, playing basketball, skating, and working together, in no particular order: Jesús, Mary-Ann, Ardi, Renato, Õie, Joonas A., Hendrik, Victor, Artem, Rita, Tarun, Marilin, Taavi K., Taavi L., Erki, Ali, Karl, Julius, Harry, Mert, Markus H., Madis, Dima, Tambet, Raivo and many more whom I cannot recall right away.

During my PhD, I had the chance to collaborate with many smart people. From Bristol, I worked with Peter, Miquel, Hao, and Telmo on the Dirichlet paper. From UT, I collaborated with Kaspar Valk, Novin Shahroudi, Andre Tättar, Viacheslav Komisarenko, Anti Ingel, Maksym Del, Marharyta Domnich, Ardi Tampuu, Ahmed Sabir, Lisa Yankovskaya, and Mark Fišel.

Also, I want to thank Taavi Kivisik and Madis Vasser, Jaan Aru and Raul Vicente, and Kälver Kilvits for their help with my bachelor's thesis.

Finally, I want to thank my family and friends who have been there for me, helping me relax after work and connecting me to the world outside academia.

Additionally, I want to note that rephrasing and grammatical corrections in this thesis were assisted by AI tools.

TLDR: Thank you to all who have contributed to the journey of my PhD, especially my supervisor, Meelis Kull, and my wife, Annika Kängsepp.

SISUKOKKUVÕTE

Määramatuse kalibreerimine ja hindamine sügavates närvivõrkudes

Masinõpe avaldab üha enam mõju paljudele igapäevaelu valdkondadele. Masintõlge, juturobotid, isejuhtivad sõidukid, pettuste tuvastamine, laenuotsuste tegemine ja tööliste värbamine on juba mõjutatud masinõppealgoritmidest ning see mõju kasvab eeldatavasti veelgi. Mudelid on muutunud üha suuremaks, keerukamaks ja võimekamaks. Seetõttu kasutatakse neid järjest enam ka sellistes tuvastus- ja ennustusülesannetes, kus vigade tagajärjed võivad olla tõsised, näiteks teiste liiklejate tuvastamisel ja nende tulevase asukoha ennustamisel isejuhtivates autodes. Sellest tulenevalt on kulukate vigade vältimiseks oluline osata mudelite väljundeid tõlgendada ja mõista, millal on mudel oma ennustustes ebakindel.

Masinõppemudelid võivad lisaks ennustustele anda ka nendega seotud tõenäosushinnanguid, mida saab kasutada otsuste tegemisel. Näiteks võib tõenäosuslik ilmaprognoos, kas järgmisel tunnil sajab vihma, mõjutada otsuseid seoses energiatootmisega, et kompenseerida vähenenud päikeseenergia tootmist. Need mudeli tõenäosushinnangud ehk mudeli ebakindluse hinnangud ei ole aga sageli piisavalt usaldusväärsed, mistõttu tuleb nende kvaliteeti hinnata. Tõenäosusi peetakse usaldusväärseks siis, kui need on hästi kalibreeritud. See tähendab, et ennustatud tõenäosused vastavad tegelikele esinemissagedustele. Näiteks kui mudel ennustab vihma tõenäosuseks 80%, peaks umbes 80 juhul 100-st tõepoolest vihma sadama.

Kalibreerituse hindamiseks kasutatakse tavaliselt usaldusdiagramme (*reliability diagrams*) ja mõõdikuid, nagu hinnanguiline kalibreerimisviga (*estimated calibration error*, ECE). Usaldusdiagramm võrdleb mudeli ennustatud tõenäosusi sündmuste tegeliku esinemissagedusega ehk täpsusega. See võrdlus tehakse tulphaaval, näiteks kasutades 10 tulpa, millest igaüks katab võrdse osa tõenäosusjaotusest (näiteks 0–10%, 10–20% jne). ECE on kvantitatiivne hinnang, mis mõõdab ennustatud tõenäosuste ja tegelike sageduste keskmist erinevust, arvutades selle üle diskreetseteks intervallideks jaotatud ennustuste. ECE arvutamiseks saab kasutada samu tulpi, mis on kasutusel usaldusdiagrammides. Seega pakub usaldusdiagramm intuiitivset ülevaadet mudeli kalibreeritusest ja ECE võimaldab väljendada kalibreerituse määra ühe arvulise väärtusena.

Kui mudel ei ole hästi kalibreeritud, saab selle tõenäosushinnanguid parandada kalibreerimismeetoditega, näiteks sügavates närvivõrkudes kasutatakse temperatuuri skaleerimise meetodit. Kalibreerimine on protseduur, mille eesmärk on parandada tõenäosuslike mudelite väljundite vastavust sündmuste tegelikele esinemissagedusele. Seda kasutatakse mudelite kalibreerituse parandamiseks, kohandades ennustatud tõenäosusi ilma mudeli sisemist ülesehitust muutmata.

Väitekiri koosneb kolmest omavahel seotud uurimusest, kus käsitletakse määramatuse kalibreerimisega seotud ülesandeid ja probleeme.

Esimeses uurimuses pakutakse välja uus meetod mitmeklassiliste mudelite

kalibreerimiseks koos uudse regulariseerimistehnikaga. See nn Dirichlet' kalibreerimine põhineb kaheklassilisel kalibreerimismeetodil – beetakalibreerimisel. Töös näidatakse, et Dirichlet' kalibreerimist on lihtne implementeerida ja see on ka teataval määral tõlgendatav. Lisaks pakutakse välja uudne meetod diagonaali-väliste parameetrite ja vabaliikme regulariseerimiseks (*off-diagonal and intercept regularisation*, ODIR). See meetod aitab vältida maatriksipõhiste kalibreerimis-meetodite ülesobitamist andmetele, eriti suure klasside arvuga klassifitseerijate puhul. Peale Dirichlet' kalibreerimise saab ODIR-i rakendada ka maatriksi ska-leerimisele. See kalibreerimismeetod on jäänud temperatuuri skaleerimise varju ülesobitamise tõttu. Uurimuses viiakse läbi katsed konvolutsiooniliste närvivõrku-dega, kasutades CIFAR-10, CIFAR-100 ja SVHN andmestikke. Katsetulemused näitavad, et Dirichlet' kalibreerimine ületab temperatuuri skaleerimise tulemusi. Dirichlet' kalibreerimine koos ODIR-iga on eriti tõhus suure klassiarvuga andmete puhul, näiteks 100 või enama klassi korral. Sügavate närvivõrkude katsetes osutub aga veidi paremaks maatriksi skaleerimise meetod koos ODIR-iga. Selle paremus tuleneb asjaolust, et see kasutab sisendina logiteid, aga Dirichlet' kalibreerimine kasutab logaritmitud tõenäosusi. Logaritmine võib põhjustada informatsioo-nikadu, mistõttu Dirichlet' kalibreerija jääb maatriksi skaleerimisele mõnevõrra alla.

Teises uurimuses tutvustatakse uut vaatenurka kalibreerituse hindamisele, mida nimetatakse testandmetele sobitamise (*fit-on-test*) lähenemiseks. Näidatakse, et ka-libreerituse hindamine hõlmab kalibreerimisfunktsiooni sobitamist, mis tähendab vastavuse leidmist kalibreerimata ja kalibreeritud tõenäosuste vahel, ning paku-takse välja uus kalibreerimisvea hindamise meetodi – tükatilineaarne (*piecewise linear*, PL) meetod. Testandmetele sobitamise lähenemine väidab, et tegelik kalib-reerimisteisendus (*calibration map*) ei ole teada, mistõttu sobitatakse kõigepealt teisendus testandmetel ja seejärel kasutakse seda kalibreerimisvea hindamiseks. Testandmetele sobitamise lähenemine toob esile mitu probleemi kalibreerimisvea hindamises. Esiteks sõltub testandmetel sobitatud teisenduse kvaliteet oluliselt kasutatava hindamismeetodi võimest seda kaardistust modelleerida. Teiseks puu-dub kasutajal usaldusväärne viis hinnata, kui täpne on saadud kalibreerimisvea hinnang.

Käsitledes kalibreerimisvea hindamist kui testandmetele sobitamise lähenemist, soovitatakse uurimuses paremaks teisenduse sobitamiseks standardsete masinõp-petehnikate kasutamist, näiteks ristvalideerimist ja regulariseerimist. Näidatakse, et ka tavapärased kalibreerimisvea hindamise meetodid, nagu ECE, kasutavad testandmetele sobitamise lähenemist. Võttes arvesse ECE meetodi puudujääke ja testandmetele sobitamise lähenemise eripärasid, pakutakse kalibreerimisvea hindamiseks välja PL meetod. See koosneb muudetava kaldenurgaga omavahel ühendatud lineaarsetest lõikudest, moodustades pideva kalibreerimisteisenduse. PL meetodit saab kasutada usaldusdiagrammi asemel nii kalibreerimisvea visuaal-seks hindamiseks kui ka selle arvuliseks mõõtmiseks. Lisaks pakutakse välja PL3 meetod, mille sisendiks ja väljundiks on logitid. Lõpuks kirjeldatakse uut viisi ka-

libreerimisvea hindamiseks, kasutades pseudoreaalset (*pseudo-real*) andmestikku sünteetilise või tavapärase (reaal)andmestiku asemel. Pseudoreaalne andmestik koosneb andmepunktidest, mis on generatiivse mudeli genereeritud ja originaalandmestikuga väga sarnased. Pseudoreaalne andmestik võimaldab suure hulga (nt miljoni) andmepunktide abil hinnata kalibreerimisteisendust väga täpselt. Seejärel saab seda teisendust kasutada erinevate meetodite kalibreerimisvea võimalikult täpselt võrdlemiseks, säilitades samal ajal seose reaalandmetega. Katsetulemused pseudoreaalset andmestikul näitavad esiteks, et PL3 meetod on parim kalibreerija ehk leiab kõige täpsema teisenduse kalibreerimata andmetelt kalibreeritud andmetele. Teiseks sobib PL meetod kõige paremini kalibreerimisteisenduse leidmiseks juba kalibreeritud andmete korral. Viimaks näidatakse, et parim kalibreerimisvea hindaja on ECE ja kalibreerimismeetodite järjestamiseks sobib kõige paremini isotooniline regressioon, mida tavapäraselt on kasutatud hoopis mudelite kalibreerimiseks.

Kolmandas uurimuses esitatakse raamistik määramatuse modelleerimiseks tajusüsteemides, keskendudes isejuhtivatele sõidukitele. Näidatakse, kuidas selles raamistikus defineeritud määramatuse hinnanguid genereerida, demonstreeritakse, et need ei ole vaikimisi hästi kalibreeritud ja vajavad lisakalibreerimist, ning pakutakse meetodeid nende kalibreerimiseks. Täpsemalt võtab raamistik arvesse tuvastatud objektide määramatusi, hõivatuse kaarte (*occupancy maps*) ja piirkondlikke hõivatuse päringuid (*region occupancy queries*). Tuvastatud objektide määramatused hõlmavad tõenäosuslikku informatsiooni objekti asukoha, olemasolu, kuju ja trajektoori kohta. Piirkondlik hõivatuse päring annab tõenäosuse, et tuvastamata jäänud objekt asub etteantud alas, näiteks jalakäijate ülekäigurajal. Raamistiku eesmärk on siduda erinevad määramatuse aspektid, et saavutada usaldusväärne ja ohutu süsteem. Töös näidatakse, kuidas raamistikus kirjeldatud määramatusi on võimalik genereerida pealtvaates esitatud semantilisest segmentatsioonist, kasutades mudelit FIERY. Samuti uuritakse, kas raamistikus esitatud määramatused on kasulikud planeerimisülesande lahendamisel. Selle katse tulemused näitavad, et kalibreeritud semantilise segmentatsiooni ennustused võimaldavad valida parema teekonna kui kalibreerimata ennustused. Kõige tõhusamaks osutub ohutu teekonna valimisel aga tõenäosuste kombinatsioon, mis hõlmab kalibreeritud objektipõhiseid määramatusi (nt asukoht ja olemasolu) ning piirkondlikke hõivatuse päringuid.

Kokkuvõttes käsitlevad need kolm uurimust mudeli tõenäosuslikke ennustusi ja nendega seotud ebakindlusi, nende kalibreerimist ning kalibreerimisvea hindamist sügavates närvivõrkudes. Töös luuakse raamistik, mis seob erinevaid määramatuse aspekte, et võimaldada ohutuma ja usaldusväärsema tajusüsteemi arendamist.

PUBLICATIONS

CURRICULUM VITAE

Personal data

Full name: Markus Kängsepp
Date of birth: 10.03.1993
Nationality: Estonia
E-mail: markus.kangsepp@ut.ee

Education

2018–2026 Ph.D. in Computer Science, University of Tartu
2016–2018 MSc in Computer Science, University of Tartu
2012–2015 BSc in Computer Science, University of Tartu

Employment

2023–2026 Scientific programmer, University of Tartu
2021–2023 Junior Research Fellow in Machine Learning, University of Tartu

Supervision

2019, BSc Thesis Jaan Erik Pihel "Modular neural networks", sup. Meelis Kull¹ and Markus Kängsepp
2020, BSc Thesis Kirill Grjaznov "Analysis of Dirichlet calibration method", Meelis Kull¹ and Markus Kängsepp
2022, MSc Thesis Martti Praks "Using Machine Learning for Default Prediction", sup. Markus Kängsepp¹, Meelis Kull and Kuldar Kõiv
2023, BSc Thesis Karl-Erik Kanal "Creating an Estonian bird image classifier using a pretrained model", sup. Markus Kängsepp¹
2024, BSc Thesis Ilmar Möls "The analysis of the effectiveness of Fiery semantic segmentation model", sup. Markus Kängsepp¹
2026, BSc Thesis Robin Henrik Neem "Visualizing confusion networks for interactive presentation of machine translations", sup. Markus Kängsepp¹
2026, BSc Thesis Oskar Männik "Application for visualising genetic relationships of Estonian beef cattle", sup. Markus Kängsepp¹

¹Supervisor in charge

Teaching

Fall 2019–2022	Introduction to Data Science (teaching assistant)
Fall 2023	Practical Data science (teaching assistant)

ELULOOKIRJELDUS

Isikuandmed

Nimi: Markus Kängsepp
Sünniaeg: 10.03.1993
Kodakondsus: Eesti
E-mail: markus.kangsepp@ut.ee

Haridus

2018–2026 Doktorikraad arvutiteaduses, Tartu Ülikool
2016–2018 Magistrikraad arvutiteaduses, Tartu Ülikool
2012–2015 Bakalaureusekraad arvutiteaduses, Tartu Ülikool

Teenistuskäik

2023–2026 Teaduslik programmeerija, Tartu Ülikool
2021–2023 Masinõppe nooremteadur, Tartu Ülikool

Juhendamine

2019, Bakalaaurusetöö Jaan Erik Pihel "Modulaarsed tehismärgivõrgud", juh. Meelis Kull¹ ja Markus Kängsepp
2020, Bakalaaurusetöö Kirill Grjaznov "Dirichlet' kalibreerimismeetodi analüüs", juh. Meelis Kull¹ ja Markus Kängsepp
2022, Magistritöö Martti Praks "Masinõppe rakendamine makseviivituse tõenäosuse hindamisel", juh. Markus Kängsepp¹, Meelis Kull ja Kuldar Kõiv
2023, Bakalaaurusetöö Karl-Erik Kanal "Eesti lindude pildiklassifitseerija koostamine eeltreenitud mudeli baasil", juh. Markus Kängsepp¹
2024, Bakalaaurusetöö Ilmar Möls "Fiery semantilise segmentatsiooni mudeli efektiivsuse hindamine", juh. Markus Kängsepp¹
2026, Bakalaaurusetöö Robin Henrik Neem "Segadusvõrkude visualiseerimine masintõlgete interaktiivseks kuvamiseks", juh. Markus Kängsepp¹
2026, Bakalaaurusetöö Oskar Männik "Eesti lihade sugulussuhete visualiseerimise rakendus", juh. Markus Kängsepp¹

Õppetöö

Sügis 2019–2022 Sissejuhatus andmeteadusesse (õppeassistent)
Sügis 2023 Praktiline andmeteadus (õppeassistent)

¹ Vastutav juhendaja

**DISSERTATIONES INFORMATICAЕ
PREVIOUSLY PUBLISHED IN
DISSERTATIONES MATHEMATICAE
UNIVERSITATIS TARTUENSIS**

19. **Helger Lipmaa.** Secure and efficient time-stamping systems. Tartu, 1999, 56 p.
22. **Kaili Müürisep.** Eesti keele arvutigrammatika: süntaks. Tartu, 2000, 107 lk.
23. **Varmo Vene.** Categorical programming with inductive and coinductive types. Tartu, 2000, 116 p.
24. **Olga Sokratova.** Ω -rings, their flat and projective acts with some applications. Tartu, 2000, 120 p.
27. **Tiina Puolakainen.** Eesti keele arvutigrammatika: morfoloogiline ühestamine. Tartu, 2001, 138 lk.
29. **Jan Villemson.** Size-efficient interval time stamps. Tartu, 2002, 82 p.
45. **Kristo Heero.** Path planning and learning strategies for mobile robots in dynamic partially unknown environments. Tartu 2006, 123 p.
49. **Härmel Nestra.** Iteratively defined transfinite trace semantics and program slicing with respect to them. Tartu 2006, 116 p.
53. **Marina Issakova.** Solving of linear equations, linear inequalities and systems of linear equations in interactive learning environment. Tartu 2007, 170 p.
55. **Kaarel Kaljurand.** Attempto controlled English as a Semantic Web language. Tartu 2007, 162 p.
56. **Mart Anton.** Mechanical modeling of IPMC actuators at large deformations. Tartu 2008, 123 p.
59. **Reimo Palm.** Numerical Comparison of Regularization Algorithms for Solving Ill-Posed Problems. Tartu 2010, 105 p.
61. **Jüri Reimand.** Functional analysis of gene lists, networks and regulatory systems. Tartu 2010, 153 p.
62. **Ahti Peder.** Superpositional Graphs and Finding the Description of Structure by Counting Method. Tartu 2010, 87 p.
64. **Vesal Vojdani.** Static Data Race Analysis of Heap-Manipulating C Programs. Tartu 2010, 137 p.
66. **Mark Fišel.** Optimizing Statistical Machine Translation via Input Modification. Tartu 2011, 104 p.
67. **Margus Niitsoo.** Black-box Oracle Separation Techniques with Applications in Time-stamping. Tartu 2011, 174 p.
71. **Siim Karus.** Maintainability of XML Transformations. Tartu 2011, 142 p.
72. **Margus Treumuth.** A Framework for Asynchronous Dialogue Systems: Concepts, Issues and Design Aspects. Tartu 2011, 95 p.
73. **Dmitri Lepp.** Solving simplification problems in the domain of exponents, monomials and polynomials in interactive learning environment T-algebra. Tartu 2011, 202 p.

74. **Meelis Kull.** Statistical enrichment analysis in algorithms for studying gene regulation. Tartu 2011, 151 p.
77. **Bingsheng Zhang.** Efficient cryptographic protocols for secure and private remote databases. Tartu 2011, 206 p.
78. **Reina Uba.** Merging business process models. Tartu 2011, 166 p.
79. **Uuno Puus.** Structural performance as a success factor in software development projects – Estonian experience. Tartu 2012, 106 p.
81. **Georg Singer.** Web search engines and complex information needs. Tartu 2012, 218 p.
83. **Dan Bogdanov.** Sharemind: programmable secure computations with practical applications. Tartu 2013, 191 p.
84. **Jevgeni Kabanov.** Towards a more productive Java EE ecosystem. Tartu 2013, 151 p.
87. **Margus Freudenthal.** Simpl: A toolkit for Domain-Specific Language development in enterprise information systems. Tartu, 2013, 151 p.
90. **Raivo Kolde.** Methods for re-using public gene expression data. Tartu, 2014, 121 p.
91. **Vladimir Sor.** Statistical Approach for Memory Leak Detection in Java Applications. Tartu, 2014, 155 p.
92. **Naved Ahmed.** Deriving Security Requirements from Business Process Models. Tartu, 2014, 171 p.
94. **Liina Kamm.** Privacy-preserving statistical analysis using secure multi-party computation. Tartu, 2015, 201 p.
100. **Abel Armas Cervantes.** Diagnosing Behavioral Differences between Business Process Models. Tartu, 2015, 193 p.
101. **Fredrik Milani.** On Sub-Processes, Process Variation and their Interplay: An Integrated Divide-and-Conquer Method for Modeling Business Processes with Variation. Tartu, 2015, 164 p.
102. **Huber Raul Flores Macario.** Service-Oriented and Evidence-aware Mobile Cloud Computing. Tartu, 2015, 163 p.
103. **Tauno Metsalu.** Statistical analysis of multivariate data in bioinformatics. Tartu, 2016, 197 p.
104. **Riivo Talviste.** Applying Secure Multi-party Computation in Practice. Tartu, 2016, 144 p.
108. **Siim Orasmaa.** Explorations of the Problem of Broad-coverage and General Domain Event Analysis: The Estonian Experience. Tartu, 2016, 186 p.
109. **Prastudy Mungkas Fauzi.** Efficient Non-interactive Zero-knowledge Protocols in the CRS Model. Tartu, 2017, 193 p.
110. **Pelle Jakovits.** Adapting Scientific Computing Algorithms to Distributed Computing Frameworks. Tartu, 2017, 168 p.
111. **Anna Leontjeva.** Using Generative Models to Combine Static and Sequential Features for Classification. Tartu, 2017, 167 p.
112. **Mozhgan Pourmoradnasseri.** Some Problems Related to Extensions of Polytopes. Tartu, 2017, 168 p.

- 113. **Jaak Randmets.** Programming Languages for Secure Multi-party Computation Application Development. Tartu, 2017, 172 p.
- 114. **Alisa Pankova.** Efficient Multiparty Computation Secure against Covert and Active Adversaries. Tartu, 2017, 316 p.
- 116. **Toomas Saarsen.** On the Structure and Use of Process Models and Their Interplay. Tartu, 2017, 123 p.
- 121. **Kristjan Korjus.** Analyzing EEG Data and Improving Data Partitioning for Machine Learning Algorithms. Tartu, 2017, 106 p.
- 122. **Eno Tõnisson.** Differences between Expected Answers and the Answers Offered by Computer Algebra Systems to School Mathematics Equations. Tartu, 2017, 195 p.

DISSERTATIONES INFORMATICAЕ UNIVERSITATIS TARTUENSIS

1. **Abdullah Makkeh.** Applications of Optimization in Some Complex Systems. Tartu 2018, 179 p.
2. **Riivo Kikas.** Analysis of Issue and Dependency Management in Open-Source Software Projects. Tartu 2018, 115 p.
3. **Ehsan Ebrahimi.** Post-Quantum Security in the Presence of Superposition Queries. Tartu 2018, 200 p.
4. **Ilya Verenich.** Explainable Predictive Monitoring of Temporal Measures of Business Processes. Tartu 2019, 151 p.
5. **Yauhen Yakimenka.** Failure Structures of Message-Passing Algorithms in Erasure Decoding and Compressed Sensing. Tartu 2019, 134 p.
6. **Irene Teinmaa.** Predictive and Prescriptive Monitoring of Business Process Outcomes. Tartu 2019, 196 p.
7. **Mohan Liyanage.** A Framework for Mobile Web of Things. Tartu 2019, 131 p.
8. **Toomas Krips.** Improving performance of secure real-number operations. Tartu 2019, 146 p.
9. **Vijayachitra Modhukur.** Profiling of DNA methylation patterns as biomarkers of human disease. Tartu 2019, 134 p.
10. **Elena Sügis.** Integration Methods for Heterogeneous Biological Data. Tartu 2019, 250 p.
11. **Tõnis Tasa.** Bioinformatics Approaches in Personalised Pharmacotherapy. Tartu 2019, 150 p.
12. **Sulev Reisberg.** Developing Computational Solutions for Personalized Medicine. Tartu 2019, 126 p.
13. **Huishi Yin.** Using a Kano-like Model to Facilitate Open Innovation in Requirements Engineering. Tartu 2019, 129 p.
14. **Faiz Ali Shah.** Extracting Information from App Reviews to Facilitate Software Development Activities. Tartu 2020, 149 p.
15. **Adriano Augusto.** Accurate and Efficient Discovery of Process Models from Event Logs. Tartu 2020, 194 p.
16. **Karim Baghery.** Reducing Trust and Improving Security in zk-SNARKs and Commitments. Tartu 2020, 245 p.
17. **Behzad Abdolmaleki.** On Succinct Non-Interactive Zero-Knowledge Protocols Under Weaker Trust Assumptions. Tartu 2020, 209 p.
18. **Janno Siim.** Non-Interactive Shuffle Arguments. Tartu 2020, 154 p.
19. **Ilya Kuzovkin.** Understanding Information Processing in Human Brain by Interpreting Machine Learning Models. Tartu 2020, 149 p.
20. **Orlenys López Pintado.** Collaborative Business Process Execution on the Blockchain: The Caterpillar System. Tartu 2020, 170 p.
21. **Ardi Tampuu.** Neural Networks for Analyzing Biological Data. Tartu 2020, 152 p.

22. **Madis Vasser.** Testing a Computational Theory of Brain Functioning with Virtual Reality. Tartu 2020, 106 p.
23. **Ljubov Jaanuska.** Haar Wavelet Method for Vibration Analysis of Beams and Parameter Quantification. Tartu 2021, 192 p.
24. **Arnis Parsovs.** Estonian Electronic Identity Card and its Security Challenges. Tartu 2021, 214 p.
25. **Kaido Lepik.** Inferring causality between transcriptome and complex traits. Tartu 2021, 224 p.
26. **Tauno Palts.** A Model for Assessing Computational Thinking Skills. Tartu 2021, 134 p.
27. **Liis Kolberg.** Developing and applying bioinformatics tools for gene expression data interpretation. Tartu 2021, 195 p.
28. **Dmytro Fishman.** Developing a data analysis pipeline for automated protein profiling in immunology. Tartu 2021, 155 p.
29. **Ivo Kubjas.** Algebraic Approaches to Problems Arising in Decentralized Systems. Tartu 2021, 120 p.
30. **Hina Anwar.** Towards Greener Software Engineering Using Software Analytics. Tartu 2021, 186 p.
31. **Veronika Plotnikova.** FIN-DM: A Data Mining Process for the Financial Services. Tartu 2021, 197 p.
32. **Manuel Camargo.** Automated Discovery of Business Process Simulation Models From Event Logs: A Hybrid Process Mining and Deep Learning Approach. Tartu 2021, 130 p.
33. **Volodymyr Leno.** Robotic Process Mining: Accelerating the Adoption of Robotic Process Automation. Tartu 2021, 119 p.
34. **Kristjan Krips.** Privacy and Coercion-Resistance in Voting. Tartu 2022, 173 p.
35. **Elizaveta Yankovskaya.** Quality Estimation through Attention. Tartu 2022, 115 p.
36. **Mubashar Iqbal.** Reference Framework for Managing Security Risks Using Blockchain. Tartu 2022, 203 p.
37. **Jakob Mass.** Process Management for Internet of Mobile Things. Tartu 2022, 151 p.
38. **Gamal Elkoumy.** Privacy-Enhancing Technologies for Business Process Mining. Tartu 2022, 135 p.
39. **Lidia Feklistova.** Learners of an Introductory Programming MOOC: Background Variables, Engagement Patterns and Performance. Tartu 2022, 151 p.
40. **Mohamed Ragab.** Bench-Ranking: A Prescriptive Analysis Approach for Large Knowledge Graphs Query Workloads. Tartu 2022, 158 p.
41. **Mohammad Anagreh.** Privacy-Preserving Parallel Computations for Graph Problems. Tartu 2023, 181 p.
42. **Rahul Goel.** Mining Social Well-being Using Mobile Data. Tartu 2023, 104 p.

43. **Anti Ingel.** Algorithms using information theory: classification in brain-computer interfaces and characterising reinforcement-learning agents. Tartu 2023, 142 p.
44. **Shakshi Sharma.** Fighting Misinformation in the Digital Age: A Comprehensive Strategy for Characterizing, Identifying, and Mitigating Misinformation on Online Social Media Platforms. Tartu 2023, 158 p.
45. **Kristiina Rahkema.** Quality Analysis of iOS Applications with Focus on Maintainability and Security Aspects. Tartu 2023, 182 p.
46. **Ivan Slobozhan.** Studying Online Social Media Engagement in CIS Countries during Protests, Mass Demonstrations and War. Tartu 2023, 81 p.
47. **Nurlan Kerimov.** Building a catalogue of molecular quantitative trait loci to interpret complex trait associations. Tartu 2023, 248 p.
48. **Pavlo Tertychnyi.** Machine Learning Methods for Anti-Money Laundering Monitoring. Tartu 2023, 117 p.
49. **Abasi-amefon Obot Affia.** A Framework and Teaching Approach for IoT Security Risk Management. Tartu 2023, 180 p.
50. **Raimond-Hendrik Tunnel.** Video Game Design and Development Bachelor's Curriculum for Estonia. Tartu 2024, 137 p.
51. **Ahto Salumets.** Bioinformatics analysis of various aspects in immunology. Tartu 2024, 198 p.
52. **Mohammed Abdulhameed Shaif Ali.** Deep Learning Methods for Cell Microscopy Image Analysis. Tartu 2024, 143 p.
53. **Pille Pullonen-Raudvere.** Foundations of Efficient and Secure Algorithm Development for Secure Multiparty Computation. Tartu 2024, 265 p.
54. **Marili Rõõm.** Multiple approaches to learners' success and factors affecting it in computer programming MOOCs. Tartu 2024, 170 p.
55. **Shivananda Rangappa Poojara.** Design and Orchestration of Scalable, Event-Driven Serverless Data Pipelines for Internet of Things (IoT) Applications. Tartu 2024, 172 p.
56. **Hassan Abdulgaleel Hassan Salim Eldeeb.** Empowering Machine Learning Pipelines with Automated Feature Engineering. Tartu 2024, 121 p.
57. **Muhammad Uzair.** Soft decision making for agri-food 4.0. Tartu 2024, 158 p.
58. **Kirill Milintsevich.** Estimation of Depression Level from Text: Symptom-Based Approach, External Knowledge, Dataset Validity. Tartu 2024, 130 p.
59. **Maksym Del.** Multilingual and Multi-Domain Representational Patterns Across Trpansformer-Based Models. Tartu 2024, 131 p.
60. **Kristo Raun.** Adaptive Out-of-order Handling in Streaming Conformance Checking. Tartu 2024, 118 p.
61. **Toivo Vajakas.** Towards integration of mobile network data into analyzing human mobility. Tartu 2024, 103 p.
62. **Katsiaryna Lashkevich.** Data-Driven Analysis and Optimization of Waiting Times in Business Processes. Tartu 2024, 169 p.
63. **Alejandra Duque-Torres.** Classifying, Constraining and Ranking Metamorphic Relations. Tartu 2025, 159 p.

64. **Mariia Bakhtina.** A Method for Information Security and Privacy Management in Smart Solutions. Tartu 2025, 199 p.
65. **Andre Tättar.** Multilingual Machine Translation for Under-Resourced Languages. Tartu 2025, 170 p.
66. **Mahmoud Shoush.** Prescriptive Process Monitoring Under Uncertainty and Resource Constraints. Tartu 2025, 178 p.
67. **Alireza Akhavi Zadegan.** A Multimodal approach for refining Mapping and Localization by Integrating Generative AI and Pedestrian-Centric Data. Tartu 2025, 147 p.
68. **Eerik Muuli.** Automating the assessment and feedback processes in IT teaching – improving creation and maintenance from the teaching staff perspective. Tartu 2025, 196 p.
69. **Kateryna Kubrak.** Towards User-Centered Prescriptive Process Monitoring Systems. Tartu 2025, 151 p.
70. **Zhigang Yin.** Computing and Sensing in a Smart Ring. Tartu 2025, 251 p.
71. **Abdul-Rasheed Olatunji Ottun.** Practical Trustworthy Artificial Intelligence with Human Oversight. Tartu 2025, 239 p.
72. **Sander Mikelsaar.** Analysis and Optimization of Iteratively Decodable Codes. Tartu 2025, 146 p.
73. **Marharyta Domnich.** Advancing Human-Centric Counterfactual Explanations in Explainable AI. Tartu 2025, 210 p.
74. **Viacheslav Komisarenko.** Aligning Training Loss to Evaluation Metrics in Deep Learning. Tartu 2026, 165 p.
75. **Heidi Taveter.** Using Programming-Process Data of Introductory Programming Courses: Finding Solver Types, Giving Feedback, and Detecting Plagiarism. Tartu 2026, 184 p.
76. **Daniel Majoral Lopez.** Deep neural networks for microscopy images. Tartu 2026, 81 p.
77. **Mahir Gulzar.** Addressing Real-world Scenarios via Motion Prediction in Autonomous Driving. Tartu 2026, 141 p.
78. **Hele-Andra Kuulmets.** Cross-Lingual Transfer Learning and Evaluation in Low-Resource Settings. Tartu 2026, 180 p.
79. **Simmo Saan.** Correctness Witnesses for Thread-Modular Program Analysis. Tartu 2026, 251 p.
80. **Tarun Khajuria.** Scene understanding in human and computer vision. Tartu 2026, 151 p.
81. **Modar Sulaiman.** From Data to Fair Decisions: On Ensuring Fairness in Machine Learning Models. Tartu 2026, 209 p.
82. **Farooq Ayoub Dar.** Autonomous Pervasive Sensing for Proactive Environmental Sustainability. Tartu 2026, 192 p.
83. **Karl Kristjan Kaup.** Neural computations underlying psychedelic insight. Tartu 2026, 256 p.
84. **Mari Seeba.** A Multi-Stakeholder Framework for Comparable and Repeatable Security Level Evaluation in Organizations. Tartu 2026, 189 p.

85. **Shan Wu.** Deep Learning Architectures for Autonomous Perception: From Sensor Auto-Calibration to Efficient Multi-Object Tracking. Tartu 2026, 175 p.
86. **Mayowa Olaide Olapade.** Human-centered Autonomous Drone Systems. Tartu 2026, 294 p.
87. **Muhammad Awais Ali.** How Will My Business Process Unfold? Predicting Case Suffixes With Start and End Timestamps. Tartu 2026, 162 p.