Investigating Deep Ensembles for the Tilted-Wave Interferometer

vorgelegt von
M. Sc.
Lara Harren (geb. Hoffmann)

an der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktorin der Naturwissenschaften
-Dr.-rer.-nat.-

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Stefan Haufe
Gutachter: Prof. Dr. Klaus-Robert Müller
Gutachter: Prof. Dr. Andreas Fischer
Gutachter: Dr. Clemens Elster

Tag der wissenschaftlichen Aussprache: 22. Juli 2022

Berlin 2022

Abstract

Optical aspheres and freeform surfaces are widely used in modern technologies and industry and form an integral part of, e.g., lenses, photo cameras or imaging optics. They can be designed to be more complex than spherical surfaces, resulting in improved functionality and light, compact optical systems. However, their further improvement is challenged by the current limitation of high-accurate optical form measurement. Further improving optical form measurements in terms of accuracy, measurement and computation time, as well as incorporating a reliable uncertainty quantification of the reconstructed topography, therefore remain topics of ongoing research for the development of new technologies with high-precision components. The tilted-wave interferometer (TWI) is a state-of-the-art measurement technique for the accurate form measurement of optical aspheres and freeform surfaces. Its evaluation procedure strongly depends on mathematical calculations including solving a high-dimensional nonlinear inverse problem. So far, the TWI analysis is based on conventional analysis methods.

The goal of this thesis is to explore the potential of deep learning for further enhancing the optical form measurement by the TWI. Data-driven machine learning methods and deep neural networks in particular have had a major impact on science and industry thanks to their ability to learn complex relationships from data. Deep learning techniques have been successfully applied and have been shown to outperform conventional methods in a variety of different applications including medical imaging, natural language processing or solving inverse problems. An advantage of deep neural networks is that they produce instant results once they are trained. This can be a benefit for time consuming applications. Also, deep learning comprises frameworks for easy-to-apply uncertainty quantifications of a solution.

Our contributions are the following. In a first proof-of-principle study, we develop a deep learning method to solve the high-dimensional inverse problem related to the form reconstruction problem of the TWI. For this purpose, a database is generated for neural network training and testing using virtual experiments. The developed method is shown to provide reliable results and a calibrated uncertainty quantification of the prediction for in-domain as well as out-of-distribution data in a simulated environment.

Next, we apply the developed deep learning approach to measurement data from a real specimen to demonstrate its practical use in a real-world environment. The results are shown to be consistent with the results of a modern TWI system from industry, and differences lie well within the estimated uncertainties of the proposed deep learning method, supporting the reliability of the developed uncertainty quantification for the deep learning approach.

Finally, a Bayesian statistical framework is developed and used to gain new insights into the role of deep ensembles that represent the current state-of-the-art for uncertainty quantification in deep learning. We propose an extension to deep ensembles that leads to an improved quantification of the epistemic part of the uncertainty in tasks of deep regression without altering the neural network training or the prediction capacity. Analytical derivations ensure easy calculations and experiments demonstrate the significant impact of the developed extension.

Zusammenfassung

Optische Asphären und Freiformflächen, wie z.B. Brillengläser, Kameraobjektive und andere Abbildungsoptiken, sind in modernen Technologien und der Industrie weit verbreitet. Durch ihre komplexen Gestaltungsmöglichkeiten helfen sie beispielsweise Abbildungsfehler zu reduzieren, was zu kompakteren und leichteren optischen Systemen führt. Jedoch ist es anspruchsvoll solche Oberflächen genau zu messen. Es wird stetig daran geforscht, die Formmessungen in Bezug auf die erzielten Genauigkeiten, Mess- und Rekonstruktionszeiten zu verbessern, sowie eine Unsicherheitsquantifizierung der rekonstruierten Topographie mit anzugeben, um neue Technologien mit hochgenauen Komponenten zu entwickeln. Das Tilted-Wave Interferometer (TWI) ist ein modernes Messgerät für die hochgenaue Formmessung von optischen Asphären und Freiformflächen. Es beinhaltet ein mathematisches Auswerteverfahren, welches bis jetzt mit konventionellen Analysemethoden bearbeitet wurde. Unter anderem muss ein hochdimensionales nichtlineares inverses Problem zur Formrekonstruktion gelöst werden.

Ziel dieser Dissertation ist, das Potential von Deep Learning (DL) für die optische Formmessung mit Hilfe des TWIs zu untersuchen. Datengetriebene Methoden und insbesondere tiefe neuronale Netze können komplexe Zusammenhänge aus Daten lernen und haben einen großen Einfluss auf die heutige Forschung und Industrie. Sie wurden bereits in verschiedensten Aufgabenfeldern erfolgreich angewendet und sind der Stand der Technik in Teilbereichen der Bildverarbeitung, natürlichen Sprachverarbeitung oder der Lösung inverser Probleme. Ein Vorteil von neuronalen Netzen ist, dass sie sofortige Ergebnisse liefern, nachdem sie trainiert wurden. Außerdem bietet DL leicht anwendbare Methoden zur Unsicherheitsbestimmung.

Die Beiträge dieser Dissertation lassen sich wie folgt zusammenfassen. In ersten Untersuchungen wird eine DL-Methode zur Lösung des hochdimensionalen inversen Problems zur Formrekonstruktion des TWIs entwickelt. Dafür wird eine Datenbank für das Trainieren und Testen tiefer neuronaler Netze mit Hilfe von virtuellen Experimenten generiert. Die Unsicherheitsquantifizierung wird auf Basis von Deep Ensembles, welche als aktueller Stand der Technik zur Quantifizierung der Unsicherheit in DL gesehen werden können, in den DL-Ansatz mit eingebaut. Es wird gezeigt, dass die entwickelte Methode zuverlässige Ergebnise und eine kalibrierte Unsicherheitsquantifizierung der Vorhersage auf simulierten Daten liefert.

Als Nächstes wird die entwickelte DL-Methode auf Messdaten eines realen Prüflings angewendet, um zu zeigen, dass sie auch in der Praxis verwendet werden kann. Ein modernes TWI-Messsystem aus der Industrie wird als Vergleichsmethode herangezogen, um die Ergebnisse bewerten zu können. Die Rekonstruktionsergebnisse der beiden Methoden sind konsistent und die Unterschiede liegen innerhalb der geschätzten Unsicherheiten.

Schließlich werden Deep Ensembles von einem Bayes-statistischem Blickwinkel untersucht, um neue Einblicke in die Methode und ihre Unsicherheitsquantifizierung zu gewinnen. Darauf basierend wird eine Erweiterung vorgeschlagen, welche zu einer verbesserten Quantifizierung der epistemischen Unsicherheit für Regressionsprobleme führt, ohne dass die Netze neu trainiert werden müssen oder sich ihre Vorhersagegenauigkeit ändert. Analytische Herleitungen sorgen für einfache Berechnungen und Experimente belegen den signifikanten Einfluss der Erweiterung.

Acknowledgment

This thesis was realized with the help of many people, whom I would like to thank.

First of all, I thank Dr. Clemens Elster for his immense support and great supervision throughout my doctoral research. In addition to answering patiently any question, his ideas and broad expertise lead to many fruitful discussions and contributed significantly to the development of this thesis.

I thank Prof. Klaus-Robert Müller and his research group from TU Berlin for teaching me the basics of machine learning and thus inspiring me to continue researching in this field. Agreeing early on to be my supervisor, his support and interest have greatly motivated my work.

Furthermore, I thank Prof. Andreas Fischer for taking the time to be a reviewer of this thesis and for his interest in my work.

My gratitude also belongs to my colleagues at PTB who were always eager to help and open for discussions. In particular, I would like to thank Dr. Ines Fortmeier for her tireless assistance and advice. I could always count on her expert knowledge, especially in optics and form measurements. I thank Manuel Stavridis for providing SimOptDevice and for his help in adapting relevant interfaces. Thank you Selma, Narbota, Tobias, Hugalf, Franko, Manuel, Jörg, Josua, Katy, Gerd, Manuel, Michael, Gregor and Fabian, I very much appreciated working with all of you.

In addition, I thank Dr. Manuel Marshall, Dr. Michael Schulz, Lukas Wessels and Bastian Harren for proofreading parts of this thesis.

Finally, I would like to express my gratitude and love to my family and friends for their support during my PhD.

Table of Contents

List of Figures

List of Tables

Abbreviations

AI	artificial intelligence
BFS	best fit sphere
CCD	charge-coupled device
CGHs	computer-generated holograms
CMM	coordinate measuring machines
CNN	convolutional neural network
DE	deep ensembles
DL	deep learning
DNN	deep neural network
ELBO	evidence lower bound
GUM	guide to the expression of uncertainty in measurement
i.i.d.	independent and identically distributed
IFK	Fredholm integral equation of the first kind
ISO	international organization for standardization
ITO	institute for technical optics of the Stuttgart university
KL	Kullback-Leibler
MAE	mean absolute error
MAP	maximum a posterior
ML	machine learning
MLA	micro lens array
MLE	maximum likelihood estimate
MSE	mean squared error
OPDs	optical path length differences
PMD	phase measuring deflectometry

PTB	Physikalisch-Technische Bundesanstalt
ReLU	rectified linear unit
RGB image	image that has a red, a green and a blue channel
RMSD	root mean squared deviation
RMSE	root mean squared error
SI	international system of units
TWI	tilted-wave interferometer

Symbols

x	input to the neural network
y	target value (ground truth)
d_x	input dimension
d_y	output dimension
D	training data set
N	number of data samples in the training set
θ	model parameters
Θ	parameter space
λ	regularization parameter
$\mathcal{L}$	loss function
$L(\theta)$	likelihood function for model parameters θ
$l(\theta)$	log-likelihood function for model parameters θ
f_θ	neural network with parameters θ
$f_{DE}(x)$	ensemble prediction for inverse problem related to TWI for input x
u	ensemble uncertainty for inverse problem related to TWI
K	number of ensemble members
C	number of input channels to the neural network
T_0	design topography
T	topography of the specimen
ΔT	difference topography
L_0	OPDs corresponding to the design topography
L	OPDs corresponding to the specimen
ΔL	difference of optical path length differences
R_1, R_2	wavefront manipulators
F	numerical forward model related to the TWI evaluation procedure
Z	Zernike polynomial function
$\mathbf{I}$	identity matrix
$q(\theta)$	approximated posterior distribution of θ
$\hat{\theta}$	estimate for the MAP of θ
$\gamma\mathbf{I}$	estimated covariance matrix for model parameters θ
$\eta(x)$	output of regression function for input x
$\hat{\eta}(x)$	output of next-to-last layer of neural network for input x

1 Introduction

1.1 Motivation

Metrology is indispensable for modern industry. The ever-growing technological progress requires similar advancements for accurate measurement capabilities. The tilted-wave interferometer (TWI) first studied in Liesener (2007) and Garbusi et al. (2008) is a state-of-the-art optical measurement system for the accurate form measurement of optical aspheres and freeform surfaces. Optical surfaces, especially non-spherical optics such as aspheres and freeform surfaces, are essential for modern optical systems, e.g. glasses, photo cameras or imaging optics (Braunecker et al., 2008; Rolland et al., 2021). Aspheres and freeform surfaces can be designed to be more complex than spherical surfaces since they have a higher degree of freedom. Advantages include aberration correction and more compact elements, leading to lighter and simpler to mount optical systems (Fang et al., 2013a). However, the high accuracy measurement of optical aspheres and freeform surfaces is challenging (Fortmeier and Schulz, 2022). Further improving the form measurement in terms of accuracy and computing time, as well as reliably quantifying its uncertainty, remain urgent open issues for the development of new technologies with high-precision components.

The TWI illuminates a surface under test with multiple differently tilted wavefronts. Thus, the TWI is applicable to a wide range of specimens with varying local slopes (Pruss et al., 2017). Reconstructing the surface form of the specimen from measurement data involves solving an inverse problem, whose treatment requires a forward model (Baer et al., 2014c). Due to the complexity of the TWI, the forward model is not available analytically, but provided by numerical simulations of the experiment. All simulations in this work related to the TWI are performed with the simulation toolbox SimOptDevice (Stavridis et al., 2018; Schachtschneider et al., 2019) developed at the Physikalisch-Technische Bundesanstalt (PTB). The virtual TWI, i.e. the numerical forward model, provides simulated data for a specified topography. The inverse problem of reconstructing the surface is solved by determining the difference of the actual topography to a given design topography. The latter corresponds to the assumed surface form of the specimen and can be expected to provide a good first approximation of the

topography. Conventional evaluation procedures of the TWI use an iterative reconstruction method based on local linearizations to solve the nonlinear inverse problem (Fortmeier et al., 2014). Accurate form measurement of optical surfaces and, in particular, the evaluation procedure of the TWI are topics of ongoing research (Gronle et al., 2022; Fortmeier and Schulz, 2022; Scholz et al., 2022).

Deep learning is a subfield of machine learning based on neural networks (LeCun et al., 2015). Machine learning methods are having a major impact on science and industry because they are able to learn complex relationships from given data (Duda et al., 2001; Balas et al., 2019). With an increasing amount of data and computing power available, especially deep neural networks have become very popular to solve complex tasks due to their deep and nonlinear structure. Examples of deep learning applications range from agriculture (Santos et al., 2019), natural language processing (Young et al., 2018) and autonomous driving (Kuutti et al., 2020) to computational imaging (Barbastathis et al., 2019), edge computing (Wang et al., 2020) and healthcare (Esteva et al., 2019). In optics, deep neural networks have been successfully applied to a variety of applications such as adaptive optics (Vdovin, 1995), phase extraction (Kando et al., 2019), wavefront sensing (Nishizaki et al., 2019) and image denoising (Zhang et al., 2017). Deep learning has also been effectively used to solve inverse problems, e.g. for inverse problems having non-unique solutions (Bishop, 1994), solving linear inverse problems (Bai et al., 2020), in imaging (Lucas et al., 2018), and magnetic resonance image reconstruction (Liang et al., 2020). This work investigates the potential of deep learning for optical form measurements solving the inverse problem associated with the TWI. Potential benefits are shorter calculation times, independence on local linearizations which might lead to an improved overall accuracy, and easy-to-apply uncertainty quantifications.

One issue when applying deep neural networks is their black-box character. A deep neural network does not indicate how trustworthy a certain prediction is. Even though a trained model might achieve good performance on the training and test data, there are various examples of trained neural networks showing an unreliable prediction behaviour. For instance, an autonomous driving car failed to correctly classify a pedestrian, resulting in an accident in 2018 in Arizona (NTSB, 2019). Subtle perturbations of the input by adversarial attacks can cause a well-trained neural network to completely change its prediction (Akhtar and Mian, 2018). Lapuschkin et al. (2019) showed that a trained deep neural network adopted a "Clever Hans" decision strategy to classify aeroplanes by looking at the image borders, where an artificially introduced padding pattern existed with uniform blue background. These examples show the necessity to deal with the black-box character when applying deep learning.

Different approaches have been put forward to recover the reliability of deep learning predictions. For example, explainability methods (Adadi and Berrada, 2018; Samek et al., 2021) were introduced to explain why a certain output occurs and to uncover erroneous decision strategies of trained models. Another important pillar of trustworthy machine learning are uncertainty methods that quantify the uncertainty of a prediction (Abdar et al., 2021; Gawlikowski et al., 2021). The total uncertainty of a prediction is thereby divided into an irreducible aleatoric part that relates to the noise in the data and a reducible epistemic part that relates to the model uncertainty which can be reduced by more available training data (Gal, 2016; Hüllermeier and Waegeman, 2021). Examples of uncertainty methods comprise

Bayesian neural networks (Kononenko, 1989), ensembling techniques (Lakshminarayanan et al., 2017; Sagi and Rokach, 2018) or dropout based methods (Kingma et al., 2015; Gal and Ghahramani, 2016). Thorough testing (Tian et al., 2018; Sun et al., 2019) and a prior detection of unusual input to the neural network (Liang et al., 2018; Martin and Elster, 2021) are also of interest for a reliable model prediction. One important aspect of this work is to develop and validate the uncertainty quantification for the inverse problem from optical form measurements based on the TWI. Uncertainty quantification also plays an important role for developing traceable reference systems, which is a primary goal in metrology (BIPM et al., 2012; Fortmeier et al., 2017, 2022).

Deep ensembles (Lakshminarayanan et al., 2017) are an ensembling technique that provide the current state-of-the-art for uncertainty quantification in deep learning, scale well to high-dimensional data, and are straightforward to implement (Ovadia et al., 2019; Caldeira and Nord, 2020; Ashukha et al., 2020; Gustafsson et al., 2020). They consist of an ensemble of independently trained neural networks, where the prediction is treated as a uniformly weighted Gaussian mixture model (Lakshminarayanan et al., 2017). Deep ensembles have originally been proposed and considered as a frequentist alternative to Bayesian neural networks (Lakshminarayanan et al., 2017; Ovadia et al., 2019). They can, however, also be viewed as an approximate Bayesian inference method replacing the true posterior of the model parameters with a uniformly weighted mixture model of delta distributions around local maximum a posterior estimates (Gustafsson et al., 2020; Wilson and Izmailov, 2020). Uncertainty quantification is a current research topic in deep learning and especially the epistemic part of the uncertainty needs further investigations (Nguyen et al., 2019; Caldeira and Nord, 2020; Hüllermeier and Waegeman, 2021; Zhou et al., 2021). Viewing deep ensembles from a Bayesian perspective allows insights to be gained in this approach and enables an extension of this state-of-the-art method to improve its epistemic uncertainty quantification, which is investigated in this work.

In summary, the goal of this thesis is to investigate the potential benefit of deep learning approaches for a challenging, high-dimensional inverse problem from optical metrology. The application is based on the tilted-wave interferometer, which is part of an ongoing research area for the accurate form measurement of optical aspheres and freeform surfaces. Steps towards incorporating an uncertainty quantification to the prediction are of interest, as well as improving the measurement accuracy and calculation time. Furthermore, the measurement of optical aspheres and freeform surfaces constitutes an interesting task for deep learning applications due to its complexity and high-dimensional data. Carefully designed reference data could serve as a benchmark set for further developments and comparison studies of different machine learning algorithms. In line with current research, we focus on trustworthy deep learning and, in particular, on uncertainty quantification. Developed methods should be validated on both in-domain and out-of-distribution data to ensure their generalization capabilities. Deep ensembles are currently considered the state-of-the-art for uncertainty quantification in deep learning. Therefore, they should be investigated for further improvements and shortcomings. One way to do so is to analyze and extend them from a Bayesian perspective which is another goal of this thesis.

In a first step, we conduct a proof-of-principle study in a virtual environment generating a database and demonstrating that the area of computational optical form measurement can benefit from deep learning (Hoffmann and Elster, 2020). Second, we incorporate and validate an uncertainty quantification by deep ensembles (Hoffmann et al., 2021b). For this purpose, out-of-distribution data are generated by systematically introducing a calibration error to the simulated optical system. Then, the developed method is successfully applied to measurement data of a real aspherical specimen and compared to a state-of-the-art method from the industry (Hoffmann et al., 2021a). Finally, we derive the theory behind deep ensembles in a Bayesian framework, leading to a proposed extension for an improved uncertainty quantification for regression problems (Hoffmann and Elster, 2021).

1.2 Objectives and aims

The goal of this work is to investigate and develop deep learning methods for solving a high-dimensional inverse regression problem from computational optical form measurements. The main objectives can be summarized as follows:

- **Objective 1:** ***Generate a database for the training and testing of deep learning methods.*** One main challenge of applying deep learning methods to real-world problems is the generation of a vast dataset that correctly represents the underlying relationships. A simulation toolbox has been developed at the PTB which includes a realization of the numerical simulation of the forward model related to the inverse problem of the TWI reconstruction procedure. This virtual TWI is extended to generate a database for the development, comparison and validation of deep learning approaches.

- **Objective 2:** ***Develop a deep learning approach for the form measurement of optical aspheres and freeform surfaces.*** Optical aspheres and freeform surfaces are widely used in industry, but their surface form is challenging to measure. A scalable deep learning approach is developed to solve the high-dimensional, nonlinear inverse problem associated with the TWI to reconstruct the surface form of the specimen from its measurement data.

- **Objective 3:** ***Extend the developed approach to incorporate an uncertainty quantification to the estimated solution.*** The main drawback of deep neural networks is their black-box character. Classic deep learning approaches do not explain why they make a certain prediction, provide only point estimates of the solution and do not highlight a potentially failed prediction which limits their applicability to real-world scenarios. Uncertainty quantification is an important pillar to deal with the black-box character and to restore the trust in the deep neural network predictions.

- **Objective 4:** ***Apply and validate the developed method on measurement data of a real specimen.*** In general, a large dataset is required for the neural network training. Here, and in many other applications, data can be generated by simulating the underlying physics. The simulated data can then be used to train, e.g., a neural network. However, the question remains if the trained neural network can generalize from the treatment of simulated data to real data. It might be unable to handle measurement data

in a real-world scenario even though it produces perfect results in a virtual environment. Furthermore, the ground truth is unknown for a real specimen and thus, it is hard to validate the estimated solution.

- **Objective 5:** ***Investigate and improve the uncertainty quantification by deep ensembles for regression problems.*** Deep ensembles mark the current state-of-the-art for uncertainty quantification in deep learning. They were originally proposed as a frequentist method, but more recently their Bayesian footing has been put forward as well. The objective here is to extend the latter point of view and explore its consequences which lead to an improved uncertainty quantification. Our focus thereby lies on regression problems, since the considered inverse problem from optical form measurement falls in that class of problems.

1.3 Thesis structure and overview

Chapter 2 (Background) introduces fundamentals of optical form measurement, deep learning and inverse problems. The application from optical form measurement considered here is based on the tilted-wave interferometer (TWI). Therefore, the TWI is placed in the wider context of metrology and its experimental setup is presented. Then, background information on deep learning is provided, focusing on aspects and approaches relevant to this work which include an introduction to artificial neural networks, possible design choices of neural network models and their training procedure, as well as frequentist and Bayesian approaches to incorporate uncertainty quantification. Furthermore, a brief introduction to inverse problems is given since the TWI-application involves solving an inverse problem.

Chapter 3 (Virtual tilted-wave interferometer) presents the virtual tilted-wave interferometer (TWI) that is needed for both the conventional evaluation procedure of the form reconstruction and the data generation process for the neural network training. The virtual TWI represents the numerical forward model of the inverse problem associated to the TWI. This work employs a realization from the simulation toolbox SimOptDevice developed at the PTB. In order to improve the numerical forward model, we include a refinement that is carried out by incorporating a phenomenological component determined by comparing predictions of the numerical model with real measurements. This chapter presents the simulation toolbox, the inverse problem of the form reconstruction and the calibration process. Also, the data generation process considering Objective 1 is explained.

Chapter 4 (Deep learning for the tilted-wave interferometer) develops a deep learning approach for the computational optical form measurement of optical aspheres and freeform surfaces based on the TWI. Additionally, we extend the proposed method to incorporate an uncertainty quantification via deep ensembles. Results are presented and validated on in- and out-of-distribution (noisy) test data. In particular, test data are used that include an increasing systematic error in the calibration of the virtual optical system. Objectives 2, 3 and partially 5 are addressed in this chapter.

Chapter 5 (Measuring a real specimen) investigates the generalization capability of the deep learning approach to measurement data. It shows how the proposed method can be

applied to measure a real specimen and how the results can afterwards be compared to other methods. More precisely, the developed approach is applied to measurement data from a real aspherical specimen. The results are then compared to the form reconstruction of a state-of-the-art method used in the industry. This chapter deals with Objectives 4 and to some extent 5.

Chapter 6 (Deep ensembles from a Bayesian perspective) discusses the deep ensembles, used in the previous chapters for uncertainty quantification, from a more general perspective. The focus lies on regression problems which naturally arise when discretizing continuous inverse problems. Deep ensembles have originally been proposed as a frequentist method but have been shown to have a Bayesian footing as well. We specify the statistical model, the prior distributions, and the form of approximation so that the uncertainty quantification by deep ensembles can be considered as an approximate Bayesian method. In addition, we explore the properties of this technique. Based on our statistical model for deep ensembles, we then propose a natural extension from the employed delta distributions to Gaussian distributions which is shown to improve the epistemic uncertainty of a new prediction. The extended method is applied to toy examples, real data, and the inverse problem of the form reconstruction related to the TWI. Objective 5 is investigated in this chapter.

1.4 List of publications

The following contributions have been published during my PhD studies. I would like to thank the co-authors for their involvement, collaboration, and for allowing to use parts of the publications in this thesis.

Peer-reviewed journals

[1] Hoffmann, L., Fortmeier, I. and Elster, C., (2021). Deep learning for tilted-wave interferometry. *tm - Technisches Messen*, vol. 89, no. 1, pp. 33-42, https://doi.org/10.1515/teme-2021-0103.

[2] Hoffmann, L., Fortmeier, I. and Elster, C., (2021). Uncertainty quantification by ensemble learning for computational optical form measurements. *Machine Learning: Science and Technology*, vol. 2, no. 3, p. 035030, https://doi.org/10.1088/2632-2153/ac0495.

[3] Hoffmann, L. and Elster, C., (2020). Deep neural networks for computational optical form measurements. *Journal of Sensors and Sensor Systems*, vol. 9, no. 2, pp. 301-307, https://doi.org/10.5194/jsss-9-301-2020.

Conferences/Workshops proceedings (short paper)

[4] Hoffmann, L., Fortmeier, I. and Elster, C., (2021). D1.3 Ensemble Learning for Computational Optical Form Measurement. *SMSI 2021-System of Units and Metrological Infrastructure*, pp. 318-319, https://doi.org/10.5162/SMSI2021/D1.3.

[5] Hoffmann, L. and Elster, C., (2020). E2.4 Deep Neural Networks for optical form measurements. *SMSI 2020-System of Units and Metrological Infrastructure*, pp. 368-369, https://doi.org/10.5162/SMSI2020/E2.4.

Conferences/Workshops talks

[6] Hoffmann, L., Fortmeier, I. and Elster, C., (2021). Uncertainty Quantification in Deep Learning for Computational Optical Form Measurement. In: *VirtMet 2021: 1st International workshop on Metrology for virtual measuring instruments.*

[7] Hoffmann, L. and Elster, C., (2021). Deep Learning for Inverse Problems – Applying Ensemble Learning for Uncertainty Quantification. In: *MSMM 2021: Mathematical and Statistical Methods for Metrology.*

[8] Martin, J., Amanova, N., Hoffmann, L., Schmähling, F. and Elster, C., (2021). Applying deep learning in metrology – an overview over some potentials and challenges. In: *MSMM 2021: Mathematical and Statistical Methods for Metrology.*

[9] Hoffmann, L., Fortmeier, I. and Elster, C., (2021). Ensemble Learning for Computational Optical Form Measurement. In: *SMSI 2021: Sensor and Measurement Science International.* → 2nd price best paper award for the short paper [4] plus the talk.

Further publications/talks

[10] Hoffmann, L., (2022). KI zur Formmessung optischer Oberflächen. In: *PTB News.*

[11] Hoffmann, L. and Elster, C., (2021). Deep Ensembles from a Bayesian Perspective. https://arxiv.org/abs/2105.13283.

[12] Hoffmann, L., (2021). Talk: Introduction to the PTB, deep learning, uncertainty uantification, and optical form measurements. In: *BIMAQ seminar: Bremen Institute for Metrology, Automation and Quality Science.*

[13] Hoffmann, L., (2021). Science slam: From mathematics to optical form measurements and deep learning. In: *SMSI 2021: Sensor and Measurement Science International*, https://www.youtube.com/watch?v=Gf4VTl97hGI.

2

Background

Deep learning techniques have been successfully applied in various domains such as computational imaging (Barbastathis et al., 2019), natural language processing (Young et al., 2018) and biomedicine (Cao et al., 2018). This work explores deep learning methods to a specific problems arising in metrology. Metrology is the science of measurement as defined in BIPM (2004). The application considered here is based on the tilted-wave interferometer (TWI) - a state-of-the-art interferometrical, optical measurement system for the accurate form reconstruction of optical aspheres and freeform surfaces. In this work, deep learning techniques are developed to solve the high-dimensional inverse problem posed by the TWI and their uncertainty is quantified.

This chapter provides background information and introduces fundamentals. It is divided into three parts. In Section 2.1, the application of optical form measurement is placed in its context and the experimental setup of the TWI is presented. In Section 2.2, basic concepts of deep learning are presented including the building blocks of deep neural networks, their training procedure, how to quantify their uncertainty, and an introduction to deep ensembles. In Section 2.3, the basics of inverse problems are briefly discussed.

2.1 Tilted-wave interferometer

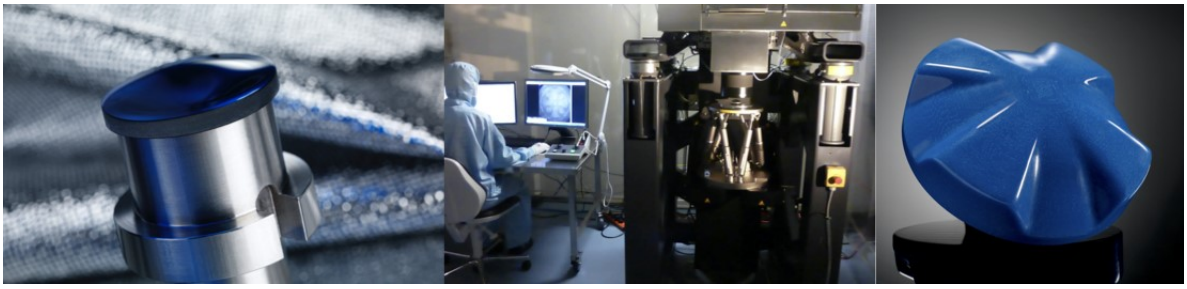

Figure 2.1: These pictures show a complex optical freeform surface (left, original source: CC UPOB e.V.), the TWI in the laboratory of the PTB (middle), and an exaggerated plastic replication of a multi-radius surface (Fortmeier et al., 2019) (right).[1]

The tilted-wave interferometer (TWI) is an interferometrical measurement system for the accurate form measurement of optical aspheres and freeform surfaces (Garbusi et al., 2011; Baer et al., 2014c), cf. Figure 2.1. The TWI measuring principle was proposed by the institute for technical optics of the Stuttgart university (ITO) in the early 2000s (Liesener, 2007; Garbusi et al., 2008) and has since then been further developed in cooperation with the PTB and the industrial partner Mahr GmbH who distributes a commercially oriented version of it (Fortmeier, 2016; Mahr GmbH, 2021; Baer, 2016; Schindler et al., 2017; Beisswanger et al., 2019).

This section introduces the application of measuring optical aspheres and freeform surfaces (2.1.1), places the TWI in the wider context of form metrology (2.1.2) and presents its experimental setup (2.1.3). More information about the simulation environment and details about the evaluation procedure are given in Chapter 3. The interested reader is also referred to Baer (2016), Fortmeier (2016) and EURAMET (2019) for more details.

2.1.1 Optical aspheres and freeform surfaces

According to Free Dictionary (2003), an *optical surface* is defined as follows:

> "An interface between two media, such as between air and glass, which is used to reflect or refract light."

In practice, optical surfaces are assumed to be sufficiently smooth as to not impede the produced optical image. This behavior of an *optically smooth surface* is described in the international organization for standardization (ISO) (ISO 10110-8:2019, 2019) as:

> An "optical surface for which the height variation of the surface texture is considerably smaller than the wavelength of light", such that "the amount of light scattered is small", and which "are usually produced by polishing or moulding".

Optical surfaces are components of optical systems. Examples of such optical systems are imaging and illumination systems (Lerner and Sasian, 2000). Their design and fabrication are challenging (Shannon, 1997). Spherical surfaces induce aberrations, such as spherical aberration, astigmatism and coma, which lead to blurred images (Hecht, 2018). Figure 2.2a illustrates spherical aberration caused by a lens with a spherical surface. The focal point of the refracted light differs for rays (green lines) passing near the center of the spherical surface (black dotted line) and for those which pass further away from it.

Spherical surfaces are uniquely defined by their radius. Aspheres and freeform surfaces have a higher degree of freedom (Thompson and Rolland, 2012). They can therefore compensate for occurring aberrations through a more complex surface form. This is illustrated in Figure 2.2b, where all refracted light rays meet in the same focal point. Furthermore, multiple spherical surfaces can be replaced by fewer non-spherical ones which lead to lighter and more compact optical systems (Schulz, 1987).

[1]Source: `https://www.ptb.de/cms/en/ptb/fachabteilungen/abt4/fb-42/ag-424.html`. Accessed: 2021-10-29.

Figure 2.2: Light rays passing through **a)** a spherical lens and **b)** an optimized aspherical lens.

The classical rotationally invariant aspheric surface is described by a conic section and a power series (ISO 10110-12:2019, 2019), i.e.

$$z(h) = \frac{h^2}{R\left(1+\sqrt{1-(1+\kappa)\left(\frac{h}{R}\right)^2}\right)} + \sum_{i=2}^{n} A_{2i} h^{2i}, \quad \text{with } h = \sqrt{x^2+y^2}, \tag{2.1}$$

where z is the so-called *sagitta*, x, y are the base coordinates, R is the radius of curvature of the base sphere, κ is the conic constant and $A_{2i}, i = 2, \ldots, n$, are the aspheric coefficients. Note that the spherical surface is a special case of the asphere definition where κ and the aspheric coefficients equal zero. Due to their improved adaptability, optical aspheres and freeform surfaces are widely used in modern optics (Braunecker et al., 2008). However, it is also more challenging to manufacture and measure them (Blalock et al., 2015).

2.1.2 Form metrology

Accurate fabrication gets increasingly important for modern technologies. However, an object can only be fabricated as accurately as it can be measured. Therefore, investigating high-accuracy form measurement systems is an ongoing research topic (EURAMET, 2014, 2019). The current measurement techniques can be grouped in different ways.

One way is to divide form metrology into *tactile* and *optical* measurement techniques (Fang et al., 2013b). Tactile techniques scan the surface under test via contact measurements where the stylus touches the surface. An advantage of tactile measurement techniques is the flexibility to measure a variety of different specimens. A disadvantage is that the surface under test can get damaged, which is highly undesirable. Optical or non-contact techniques, on the other hand, do not touch the surface during the measurement. They do not corrupt the surface of the specimen and achieve faster measuring times than tactile techniques.

Form metrology can also be divided into *pointwise* and *areal* measurement techniques. Pointwise techniques are based on sensors (tactile or optical) that scan the surface at predefined measurement points. Both, the measurement accuracy and time, depend on the number of surface points to be scanned. These measurement systems are known as coordinate measuring machines (CMM) (Flack, 2014). In contrast, areal measurement techniques (which are commonly optical) do not scan individual points on the specimen, but wider areas at a time. Examples are interferometry and phase measuring deflectometry (PMD) (Häusler et al., 2013; Huang et al., 2018). The advantages of areal measurement techniques are the comparatively short measurement time and the resulting high resolution.

The phenomen of interference occurs when two light waves superpose (Hariharan, 2010). The resulting intensity fringe patterns are called *interferograms*. The basic idea of optical

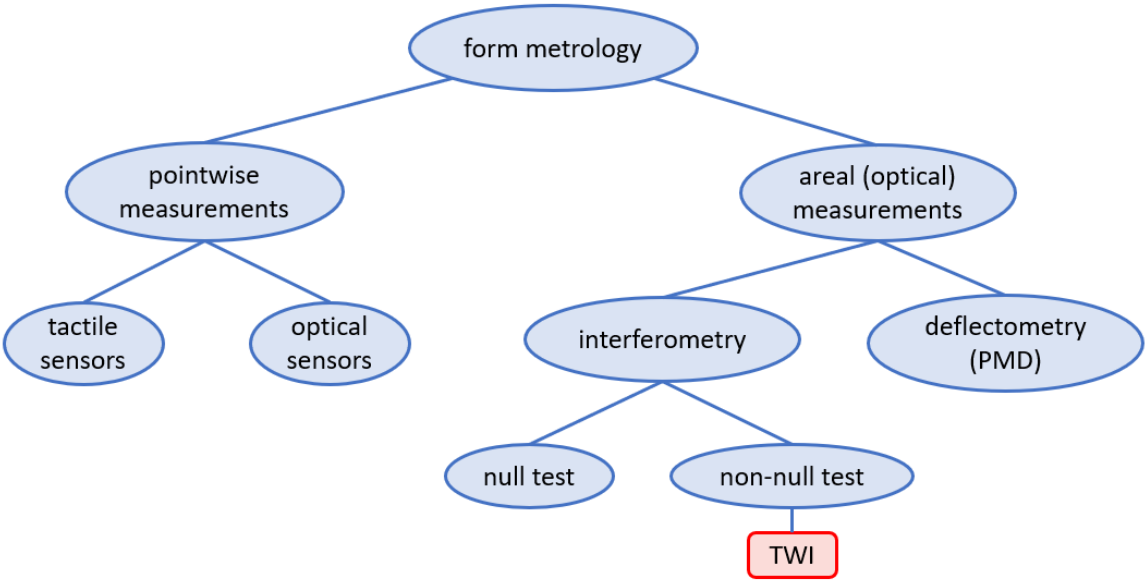

Figure 2.3: The diagram places the tilted-wave interferometer (TWI) in the wider context of form metrology.

interferometry (Malacara, 2007) for form reconstruction is to collect the interfering coherent light between a test wavefront reflected by the surface under test and the reference wavefront. There exist two possible configurations of the optical system, namely the *null test* and the *non-null test* configuration. In the null test configuration, the test wavefront reaches the test topography perpendicular to its surface. Hence, the reflected light follows the same path through the optical system as the incoming light. The null test configuration is highly desirable as errors introduced by the optical system extinguish themselves (DeVries et al., 2006). In non-null test configurations, the test wavefront does not reach the surface under test with an angle of 90 degrees and the reflected light then takes a different path through the optical system than the incoming light. This leads to a more complex ray path and some light rays might even be deviated in such a way as to leave the optical system completely causing so-called *vignetting errors*.

Spherical and planar reference wavefronts can be produced with high accuracy. Thus, spherical and planar surfaces can be measured in a null test configuration. Asphere and freeform metrology is very challenging because the surfaces can strongly deviate from spheres or planes and it is therefore not trivial to find a null test configuration. Some measurement techniques are based on finding a compensatory element to deviate the reference wavefront depending on the individual specimen under test such that a null test configuration is possible. Examples are computer-generated holograms (CGHs) (Pruss et al., 2004). Also, some techniques measure multiple subapertures in individual null test configurations and stitch together the results afterwards (Murphy et al., 2003). Other methods find a way to cope with the non-null test configuration. An example is the TWI (Garbusi et al., 2008; Baer et al., 2014b). Figure 2.3 presents an overview of the introduced techniques and places the TWI in its metrological context.

The term *computational* metrology was introduced to emphasize the trend towards using the computational power of computers, e.g. for data analysis and calculations, in addition

to the classical measurement setup (Hopp et al., 1993; Bergmann et al., 2016). It marks methods that depend on the computing power to process the conducted raw data. In summary, the TWI is an interferometrical system for the computational optical form measurement of optical aspheres and freeform surfaces using a non-null test configuration. Its detailed setup is described below.

2.1.3 Experimental setup

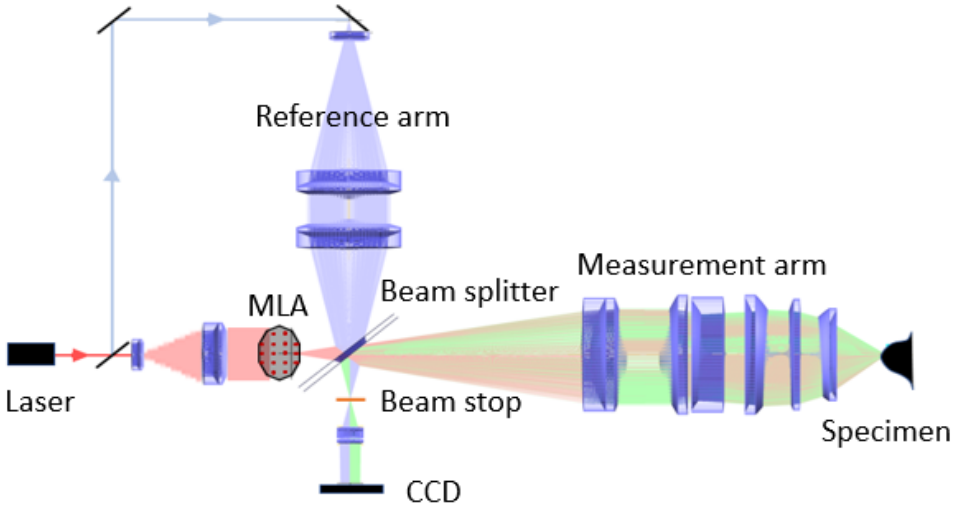

Figure 2.4: Experimental setup of the TWI used in this thesis. Incoming light is plotted in red, light reflected by the specimen is green, and the light coming from the reference arm is colored blue. Only the center point source of the MLA is activated in this illustration.

The experimental setup of the TWI used in this thesis is shown in Figure 2.4. The specimen is illuminated with monochromatic green light, which is reflected by the surface of the specimen and interferes with coherent light from the reference arm at the beam splitter. In Figure 2.4, incoming light is plotted in red, light reflected by the specimen is shown in green, and the light coming from the reference arm is colored blue. The interferograms, i.e. the intensity of the interfering light from the measurement and reference arm, are then collected at the charge-coupled device (CCD).

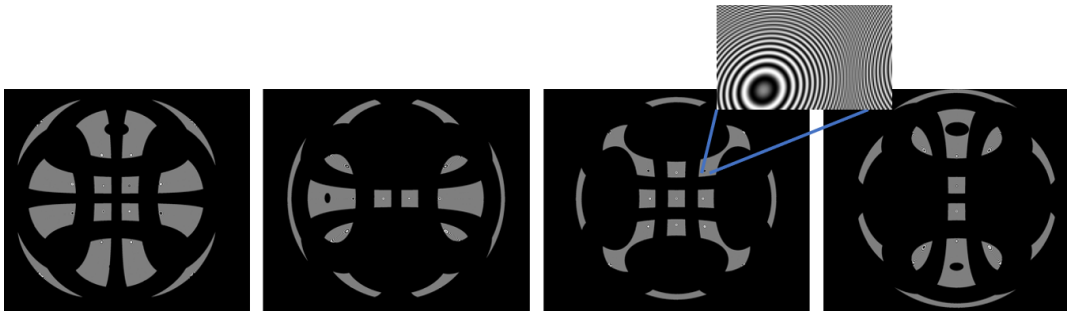

Figure 2.5: Interferograms captured by the TWI using four disjoint masks on the MLA. The images have a resolution of 2048×2048 pixels. The interferograms shown here are simulated using the virtual TWI (cf. Chapter 3).

A special feature of the TWI setup is the 2D micro lens array (MLA) that acts as a 2D point source array. Each point source produces a differently tilted wavefront. These wavefronts

then reach the specimen from different angles, so that the surface under test is well illuminated. Four disjoint masks on the MLA subsequently cover different light sources of the 2D point source array to enhance an overlap of information without having overlapping interferograms on the camera, which would otherwise lead to overlapping fringe patterns that could not be easily evaluated. More precisely, each mask activates every second point source row- and column-wise. Finally, a beam stop in the Fourier plane of the imaging optics limits the fringe density on the CCD to ensure resolvable interferograms. Figure 2.5 shows an example of resulting interferograms where each patch can be attributed to a single light source (note that one source can generate multiple patches).

2.2 Deep learning

Deep learning methods have become very popular due to their ability to learn complex relationships from given data. *Artificial intelligence (AI)* is the overall area to which they belong. According to Oxford University Press (2021a), AI is defined as follows:

> "The theory and development of computer systems able to perform tasks normally requiring human intelligence, ..."

The notion of data is explicitly introduced to the subdomain of AI known as *machine learning (ML)* defined as follows according to Oxford University Press (2021b):

> "The use and development of computer systems that are able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyse and draw inferences from patterns in data."

There are various well-known machine learning algorithms (Duda et al., 2001). Examples comprise principal component analysis, nearest-neighbor classification, support vector machines, decision trees or neural networks. According to Oxford University Press (2021c), *deep neural networks DNNs* or *deep learning* are defined as follows:

> "A type of machine learning based on artificial neural networks in which multiple layers of processing are used to extract progressively higher level features from data."

Figure 2.6a shows a summary of the above introduced terms.

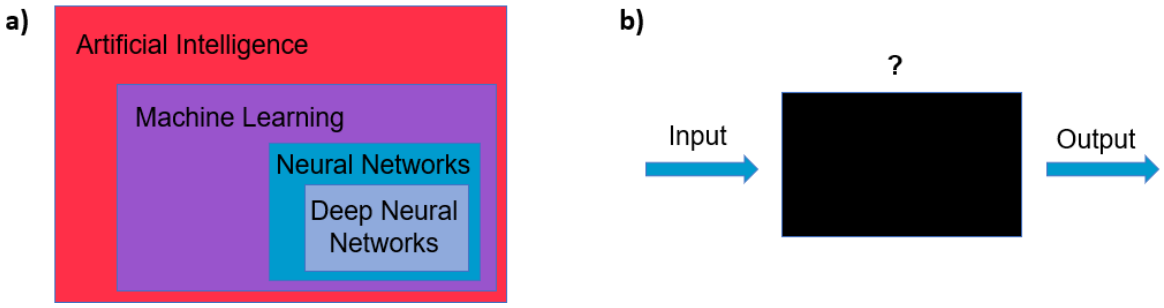

Figure 2.6: a) Overview of frequently used, AI related terms. **b)** In general, deep neural networks are a black box for the user.

Machine learning algorithms can be grouped into three learning paradigms: reinforcement, unsupervised, and supervised learning (Sasakawa et al., 2008; Sutton and Barto, 2018; Duda et al., 2001). *Reinforcement learning* algorithms try to maximize a reward signal by interacting with an environment. An agent exploits and explores possible actions, gets a reward that can depend on multiple sequentially taken actions, and learns from the experience. Reinforcement learning applications comprise natural language processing, chat bots or playing Atari video games. *Unsupervised learning* is based on finding hidden patterns in given input data without having any corresponding target values. Tasks include the detection of different clusters, the projection into a lower-dimensional space, data visualization and data generation. *Supervised learning* relies on training data that contain input values as well as the corresponding target values. The goal is to find the approximation f_θ that maps an input x to its corresponding output y for all data samples (x, y) in the training set D, i.e.

$$\text{Find } f_\theta : x \mapsto y, \text{ such that } \forall (x, y) \in D : \; f_\theta(x) \approx y, \tag{2.2}$$

where θ are the learnable model parameters. Tasks, where the input is mapped to a finite number of discrete class labels, are called *classification* problems. Examples are binary decision making, handwritten digit recognition or image segmentation. In contrast, *regression* problems have continuous target values. Examples comprise pricing, forecasting the air pollution from sensor data or reconstructing a topography from measurement data. This work focuses on the supervised learning setting while considering an image-to-image regression problem.

Machine learning methods and especially deep neural networks are used in a wide range of applications today (Shinde and Shah, 2018; Dargan et al., 2020). However, their main drawback is their black-box character (cf. Figure 2.6b) and the resulting lack of trustworthiness. An incoming signal is processed by the trained model which then generates an output. Why exactly this output occurs and how uncertain it is remains unknown. But answering these questions is crucial if important decisions depend on the model prediction, for instance in medical diagnosis and autonomous driving. Also, adversarial attacks show that a small perturbation in the input data can mislead a trained model to make erroneous predictions (Akhtar and Mian, 2018).

Different approaches are being investigated to deal with this black-box character. The most common are explainability and uncertainty methods. *Explainability* methods try to explain the model behavior and to make the model predictions more transparent and easier to interpret (Baehrens et al., 2010; Binder et al., 2016; Selvaraju et al., 2017; Montavon et al., 2018; Adadi and Berrada, 2018; Samek et al., 2019). In Lapuschkin et al. (2019), for example, it was shown that an apparently well-trained neural network for image classification had learned to correctly classify horse images because of a small tag at the bottom left corner of the images which had remained unnoticed in the training dataset. The focus of this thesis lies on *uncertainty quantification* which is introduced in Subsection 2.2.3. Of course, it is also important to thoroughly test a trained model as to analyze its behavior and generalization capability (Sun et al., 2018; Tian et al., 2018; Xu et al., 2020). Additionally, one could try to detect outliers and unusual input prior to the network prediction (Wang et al., 2019; Martin and Elster, 2021).

The following subsection introduces neural networks in general, discusses different architectural building blocks and presents convolutional neural networks that play an important

role in imaging problems (2.2.1). Then, the optimization problem for the neural network training is stated and the backpropagation algorithm is explained (2.2.2). Different options and hyperparameters for the neural network training are presented and the problem of overfitting is discussed. Afterwards, various approaches and methods are presented of how to incorporate an uncertainty quantification into the neural network prediction, the different types of uncertainty called aleatoric and epistemic uncertainties are explained, and the framework of Bayesian inference (2.2.3) is introduced. Finally, the deep ensembles are introduced separately, since they are relevant for this work (2.2.4).

The interested reader is also referred to Duda et al. (2001), Bishop (2006) and Blei et al. (2017) for more information.

2.2.1 Artificial neural networks

From perceptrons to feedforward neural networks. A well-known predecessor of neural networks is the *perceptron* that was introduced by the American psychologist and AI pioneer Frank Rosenblatt (1958). The perceptron is a model for binary classification. It produces a single binary output $y \in \{0, 1\}$ for an input vector $x \in \mathbb{R}^n$, where n is the input dimension. For example, perceptrons can be applied to model logical gates as AND and OR (Nielsen, 2015). Figure 2.7a shows a perceptron. The output depends on the learnable model parameters, i.e. the weights $w \in \mathbb{R}^n$ and the bias $b \in \mathbb{R}$ (also called threshold value). It is generated by the Heaviside unit step function applied to a weighted mean of the input values and shifted by the threshold b, i.e.

$$y(x) = \begin{cases} 1 & \text{if } w^\top x + b > 0 \\ 0 & \text{otherwise} \end{cases} . \tag{2.3}$$

Combining many of these perceptron units into one *multilayer perceptron* (cf. Figure 2.7b) allows to solve more complex problems and paved the way for the neural networks widely used today (Gardner and Dorling, 1998; Nielsen, 2015).

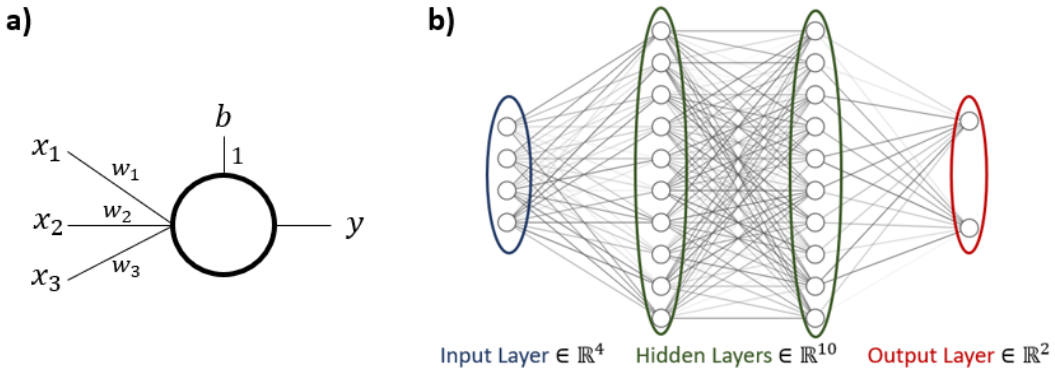

Figure 2.7: a) A perceptron with input $x \in \mathbb{R}^3$, weights $w \in \mathbb{R}^3$, bias $b \in \mathbb{R}$ and output $y \in \{0, 1\}$ is illustrated. **b)** A fully connected feedforward neural network (or multilayer perceptron) with two hidden layers is shown (LeNail, 2019). The gray scale of the connections between two neurons depends on the corresponding weight value.

Artificial neural networks or simply *neural networks* consist of many individual neurons interacting with each other. A neuron receives an incoming signal, might get activated and transmits a signal to the adjacent neurons. The output y of a single neuron can be formalized

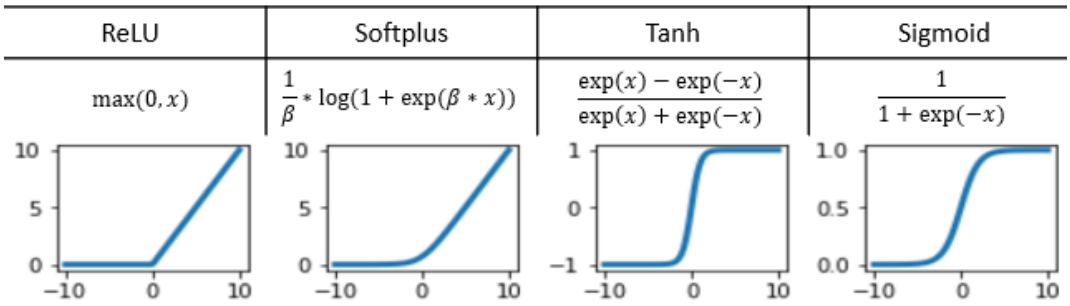

Figure 2.8: Different activation functions. x is the input and β equals one in the plot of the softplus function.

as follows:

$$y(x) = h\left(w^\top x + b\right) \in \mathbb{R},\ x \in \mathbb{R}^n,\ w \in \mathbb{R}^n,\ b \in \mathbb{R}, \tag{2.4}$$

where x is the n-dimensional input and w and b are learnable model parameters. The so-called *activation function* h is a nonlinear transformation that is applied elementwise. An example is the step function from equation (2.3). Other well-known activation functions for regression problems are shown in Figure 2.8 (Paszke et al., 2019; Sharma et al., 2017). Neural networks consist of multiple layers, each containing one or more neurons. More precisely, they have an input layer comprising the input signal, one or multiple hidden layers that depend on the network parameters and whose neurons behave according to equation (2.4), and finally, an output layer that computes a linear combination of the output of the previous layer. In addition, the output layer can include offsets or other transformations depending on the given problem, e.g. constraints on the output space. Figure 2.7b shows a neural network with two hidden layers. A *deep neural network (DNN)* (Schmidhuber, 2015) is a neural network with many hidden layers.

If the information flows only in one direction from the input to the output, the artificial neural network is called a *feedforward* neural network. Otherwise, if feedback connections between the neurons are included and a cyclic information flow occurs, the neural network is called a *recurrent* neural network (Goodfellow et al., 2016). This work focuses on feedforward neural networks. A feedforward neural network f_θ can be defined as follows:

$$f_\theta(x) = W^{(L+1)T} h^{(L)}\left(W^{(L)T} \cdots h^{(1)}(W^{(1)T} x + b^{(1)}) \cdots + b^{(L)}\right) + b^{(L+1)}, \tag{2.5}$$

where $\theta = \left\{W^{(l)} \in \mathbb{R}^{d_{l-1} \times d_l}, b^{(l)} \in \mathbb{R}^{d_l} : l = 1, \ldots, L+1\right\}$ are the network parameters, $h^{(l)}$ are the activation functions of the corresponding layers $l = 1, \ldots, L$, L is the number of hidden layers and $d_0 = d_x$, $d_{L+1} = d_y$ are the dimensions of the input space $x \in \mathbb{R}^{d_x}$ and the output space $f_\theta(x) \in \mathbb{R}^{d_y}$, respectively. Neural networks are universal approximators (Cybenko, 1989; Hornik et al., 1989; Sonoda and Murata, 2017; Park et al., 2020), e.g. feedforward neural networks with a single hidden layer having sufficiently many neurons can approximate any continuous function to arbitrary accuracy on a compact input domain.

Fully connected vs. convolutional neural networks. Figure 2.7b shows a *fully connected* neural network because each neuron of one layer is connected to every neuron of the following layer. If a neuron is directly connected to another neuron that is more than

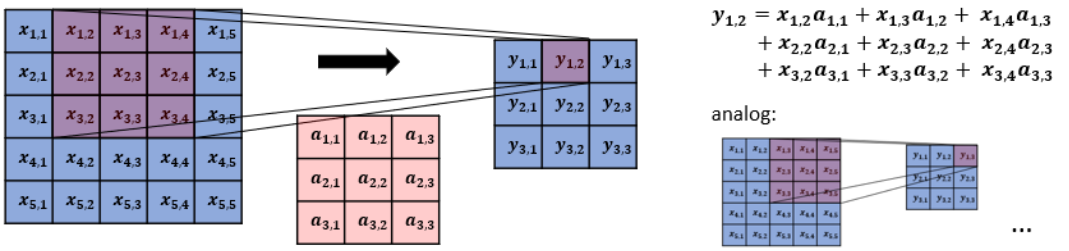

Figure 2.9: The convolutional filter $A = (a_{k,l})_{k,l=1}^{3}$ slides over the input $X = (x_{i,j})_{i,j=1}^{5}$ and produces the output $Y = (y_{m,n})_{m,n=1}^{3}$.

one layer apart, this is called a *skip connection* because the information flow skips a layer. Fully connected neural networks have some disadvantages. They can be computationally expensive for an increased network size and a high-dimensional input. Consider an imaging problem where the input is an RGB image with a resolution of 64×64 pixels, and assume the first fully connected hidden layer to have only 100 neurons. Then, the resulting number of learnable network weights (not including the biases) would already be over a million ($3 \times 64 \times 64 \times 100 = 1,228,800$). The high flexibility of fully-connected neural networks can also lead to overfitted results when the training data are scarce (Bishop, 2006). Moreover, they have no built-in notion of structured data, e.g. local spatial correlations and translational invariances, which would be beneficial for applications related to imaging, time-series and speech.

Convolutional neural networks (CNNs) have been introduced to overcome these shortcomings (LeCun et al., 1989b, 1995; Bishop, 2006; Goodfellow et al., 2016). CNNs include at least one convolutional layer consisting of one or multiple filters that slide across the input as shown in Figure 2.9. These filters are also called *kernels*, their values are learnable model parameters, and they project the input onto so-called *feature maps* because they are able to extract certain features, e.g. edges from images. The underlying operation is a convolution, although most implementations use a cross-correlation that is equal to the convolution without flipping the kernel (Goodfellow et al., 2016; Dumoulin and Visin, 2016). Assume a 2D input $X \in \mathbb{R}^{I \times J}$ and a kernel $W \in \mathbb{R}^{K \times L}$ with $K \leq I, L \leq J$. Then, the cross-correlation $(W * X) \in \mathbb{R}^{(I-K+1)\times(J-L+1)}$ is defined as follows:

$$(W * X)_{m,n} = \sum_{k=1}^{K} \sum_{l=1}^{L} X_{m+k-1,n+l-1} W_{k,l}, \tag{2.6}$$

where $m = 1, \ldots, I - K + 1$, $n = J - L + 1$. Using some convolutional layers instead of fully connected layers can drastically reduce the number of learnable weights and enhance structured feature extraction. The sizes and number of filters need to be chosen as part of the network architecture. In addition, the *stride* can be changed and *padding* introduced. The stride controls the step-size of the filter sliding across the input and padding can increase the input dimension (and consequently the output dimension) by expanding the input with zero-values. Deep convolutional neural networks have been successfully applied to a wide range of applications including inverse problems in imaging (McCann et al., 2017), brain magnetic resonance imaging (Bernal et al., 2019), speech enhancement (Park and Lee, 2016) and object detection (Dhillon and Verma, 2020).

Choice of design. A major challenge of applying deep neural networks is finding a suitable network architecture and optimal training hyperparameters. Table 2.1 shows a summary of popular layers used to build a neural network (Paszke et al., 2019). More details about the training procedure are discussed in the following subsection (2.2.2).

Table 2.1: Different types of architectural layers for neural networks are presented.

type of layer	description
linear	A linear layer is fully connected. It holds that $y = W^\top x + b$, where $x \in \mathbb{R}^{d_x}$ is the input, $y \in \mathbb{R}^{d_y}$ is the generated output for the chosen number d_y of neurons in this layer, and $W \in \mathbb{R}^{d_x \times d_y}$, $b \in \mathbb{R}^{d_y}$ are learnable network parameters.
convolutional	Convolutional layers (LeCun et al., 1989b) are discussed above (cf. Figure 2.9). In general, they have less learnable parameters than fully connected layers, apply linear transformations on subsets of the input (cf. equation (2.6)), and facilitate feature extraction as well as the detection and processing of local spatial correlations. Parameters to be chosen include the kernel size, the number of kernels, the stride and the padding. Learnable parameters are the kernel values and the optional bias.
activation	The activation functions are responsible for the nonlinearity of the neural network. See Figure 2.8 for some examples.
pooling	Pooling layers often follow convolutional layers and are used for downsampling (Dumoulin and Visin, 2016). Common choices comprise max pooling and average pooling layers which take the maximum and average, respectively, over certain regions of the input. The regions are determined by the chosen pooling size and stride.
batch normalization	Batch normalization can reduce the network's sensitivity to its initialization, reduce internal covariate shifts and act as a regularizer (Ioffe and Szegedy, 2015). The output is produced by subtracting the mean of the batch and dividing the difference by its standard deviation. Then, the output is shifted by an offset and scaled. The offset and scaling factor are learnable model parameters.
dropout	Input elements are randomly set to zero with a chosen probability. Dropout layers can reduce overfitting (Srivastava et al., 2014) and can be used for uncertainty quantification (Gal and Ghahramani, 2016).

In addition to fully connected feedforward and convolutional neural networks, there exist more sophisticated, popular architectures. Some of them are presented in the following. *Autoencoders* are widely used for dimensionality reduction (Hinton and Salakhutdinov, 2006). They consist of an encoder, that maps high-dimensional data into a low-dimensional latent space (also called bottleneck), and a decoder that recovers the high-dimensional data from the latent space. Their probabilistic counterpart is the *variational autoencoder* (Kingma and Welling, 2013), where the variables in the latent space are treated as random variables and the decoder yields a distribution instead of a point estimate as an output which can

be employed, e.g. for image generation (Vahdat and Kautz, 2020). *Generative adversarial networks* (Goodfellow et al., 2014) consist of two players called generator and discriminator. The discriminator learns to distinguish real data from artificially generated data while the generator tries to trick the discriminator by learning how to generate realistically looking data. Applications comprise data generation, style transfer and image superresolution (Creswell et al., 2018). *Residual neural networks* have been introduced to ease the training of deep neural networks by minimizing the residual between the input and the output of a specified block of layers (He et al., 2016). Finally, the *U-Net* (Ronneberger et al., 2015) was introduced to better exploit information from the input data by combining a bottleneck architecture with additional skip-connections. U-Nets have been successfully applied to various imaging tasks (Barbastathis et al., 2019).

2.2.2 Neural network training

Optimization problem. Once the architecture of a neural network is chosen, the next step is to find the best model parameters to optimally solve the given problem. Therefore, a function to be optimized must first be defined - the *loss function* $\mathcal{L}$ (depending on the application also *cost*, *risk* or *error function*) (Bishop, 2006). The goal is to find the optimal network parameters θ, such that the loss function is minimized:

$$\min_{\theta} \sum_{(x,y)\in D} \mathcal{L}\left(y, f_\theta(x)\right), \tag{2.7}$$

where D is the set of training data and $f_\theta(x)$ is the neural network output for an input x. The standard loss for regression tasks is the *mean squared error (MSE)* also called L_2*-loss*. It is defined as follows for a data point $(x, y) \in D$, with $x \in \mathbb{R}^{d_x}$ and $y \in \mathbb{R}^{d_y}$:

$$\mathcal{L}_{MSE}(y, \hat{y}) := \frac{1}{d_y}\|y - \hat{y}\|_2^2 = \frac{1}{d_y}\sum_{i=1}^{d_y}(y_i - \hat{y}_i)^2, \tag{2.8}$$

where $\hat{y} = f_\theta(x) \in \mathbb{R}^{d_y}$. Other popular loss functions (Paszke et al., 2019) are the *mean absolute error (MAE)* or L_1*-loss*, i.e. $\mathcal{L}_{MAE}(y, \hat{y}) = \frac{1}{d_y}\|y - \hat{y}\|_1 = \frac{1}{d_y}\sum_{i=1}^{d_y}|y_i - \hat{y}_i|$, and the *Huber loss* that behaves like the MSE around zero and otherwise like the MAE. Of course, it is also possible to handcraft loss functions for specific problems.

Training by backpropagation. Training neural networks (Duda et al., 2001; Bishop, 2006; LeCun et al., 2012; Nielsen, 2015) is based on gradient descent (Cauchy et al., 1847) and *backpropagation* (Rumelhart et al., 1986). The general ideas are described in the following while assuming that all mentioned derivatives exist. For reasons of simplicity, derivatives of the loss function are always evaluated at $(y, f_\theta(x))$, where the tuple (x, y) is a data point from the training set. After initialization, the network parameters θ are iteratively updated using gradient descent, i.e.

$$\theta_{new} = \theta_{old} - \eta\frac{\partial \mathcal{L}}{\partial \theta}, \tag{2.9}$$

where η is the *learning rate* which determines the step size. The gradient is computed using the backpropagation algorithm. Consider the neural network from equation (2.5) with

L hidden layers and network parameters $\theta = \left\{W^{(l)} \in \mathbb{R}^{d_{l-1} \times d_l}, b^{(l)} \in \mathbb{R}^{d_l} : l = 1, \ldots, L+1\right\}$ with $d_0 = d_x$ and $d_{L+1} = d_y$. Assume all activation functions to be the same for simplicity of notation, i.e. $h^{(l)} = h$, $l = 1, \ldots, L$. Then, the backpropagation algorithm is divided into two parts.

First, the *forward pass* computes the weighted input z and the activation a of every neuron j from layer l as follows:

$$z_j^{(l)} := \sum_{k=1}^{d_{l-1}} w_{k,j}^{(l)} a_k^{(l-1)} + b_j^{(l)}, \qquad l = 1, \ldots, L+1, \qquad f_\theta(x) = z^{(L+1)}, \tag{2.10}$$

$$a_j^{(l)} := h\left(z_j^{(l)}\right), \qquad l = 1, \ldots, L, \qquad a^{(0)} := x. \tag{2.11}$$

Second, the corresponding derivatives are computed in the *backward pass* using the results from the forward pass via the chain rule, i.e.

$$\delta_j^{(l)} := \frac{\partial \mathcal{L}}{\partial z_j^{(l)}} = \sum_{i=1}^{d_{l+1}} \frac{\partial \mathcal{L}}{\partial z_i^{(l+1)}} \frac{\partial z_i^{(l+1)}}{\partial z_j^{(l)}} = \sum_{i=1}^{d_{l+1}} \delta_i^{(l+1)} w_{ji}^{(l+1)} h'\left(z_j^{(l)}\right), \qquad l = 1, \ldots, L, \tag{2.12}$$

$$\delta_j^{(L+1)} = \frac{\partial \mathcal{L}}{\partial z_j^{(L+1)}} = \frac{\partial \mathcal{L}}{\partial f_\theta(x)_j}. \tag{2.13}$$

Finally, the partial derivatives of the neural network parameters can be calculated for all $l = 1, \ldots, L+1$, $k = 1, \ldots, d_{l-1}$, $j = 1, \ldots, d_l$, as follows:

$$\frac{\partial \mathcal{L}}{\partial w_{k,j}^{(l)}} = \frac{\partial \mathcal{L}}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial w_{k,j}^{(l)}} = \delta_j^{(l)} a_k^{(l-1)} \text{ and } \frac{\partial \mathcal{L}}{\partial b_j^{(l)}} = \frac{\partial \mathcal{L}}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial b_j^{(l)}} = \delta_j^{(l)}. \tag{2.14}$$

Training options, hyperparameters and the problem of overfitting. The trained neural network should be validated on data that were not included during training. If there is a large discrepancy between the model error on the training data and the validation data, then, the trained model *overfit*s the training data. Overfitting occurs when the model complexity and the optimization procedure allow the model to fit the training data significantly better than can be expected given the amount of available training data and the assumed noise level. The model complexity can be considered in the model selection process. *Occam's razor*, based on a medieval philosopher, can be understood such that if two models perform equally well, the simpler one should be chosen (Blumer et al., 1987). A simpler model is less prone to overfitting and hence, more likely to generalize to unseen data. The *bias-variance dilemma* (or *bias-variance tradeoff*) (Geman et al., 1992; Bishop, 2006) says that the bias of a predictive model decreases with growing model complexity, but its variance increases at the same time. The MSE (equation (2.8)) of a predictive model can be decomposed into a bias and a variance term. A good model needs to find a suitable tradeoff between the two terms. However, more recent work states that the bias-variance tradeoff is no longer valid in an over-parameterized setting which is often the case in modern deep learning applications (Belkin et al., 2019; Neal et al., 2018).

Different techniques are used to reduce overfitting (Ying, 2019) once a model is selected. Examples are the incorporation of dropout layers in the neural network architecture (cf.

Table 2.1), data augmentation to increase the variability of the training data (cf. Table 2.2), regularizing the model parameters by adding a penalty term to the loss function (cf. Table 2.2), or simply stopping the training procedure before the neural network has fully converged which is called *early stopping (Caruana et al., 2001)*. Table 2.2 presents an overview of different options and hyperparameters that can be considered for neural network training.

Table 2.2: Various options and parameter settings are discussed for the neural network training.

option	description
data normalization	The standard procedure is to normalize the data around zero with standard deviation one, which is realized as follows: $$x_{new} = \frac{x - \mu}{\sigma}, \text{ where } \mu = \frac{1}{N}\sum_{i=1}^{N} x_i, \text{ and } \sigma = \left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2\right)^{1/2}.$$ If $d_x > 1$, the operations are carried out pointwise. Another possibility is to subtract the minimum and divide the difference by the difference between the maximum and the minimum over the training data. Then, the training data lie between zero and one. Without data normalization, for example, input features with different scales of magnitude could result in a biased network prediction that depends mainly on the input features with the greater scale of magnitude (Sola and Sevilla, 1997).
data augmentation	In general, the network prediction can be improved and overfitting reduced when more training data are available. Data augmentation techniques (Shorten and Khoshgoftaar, 2019) can help to generate more training data without an additional labeling effort. Examples comprise adding random noise, adversarial training or, for imaging data, geometrical transformations such as rotations and translations.
optimizer	There exist different optimization algorithms that are based on gradient descent and the backpropagation procedure explained above (Paszke et al., 2019). The standard stochastic gradient descent uses random subsets of the training data (defined by the mini-batches) for the parameter updates in accordance with equation (2.9). A momentum term can be added to the parameter update to dampen the change of direction and to reduce the oscillation along the path of steepest descent (Sutskever et al., 2013). The most famous algorithm for training neural networks probably is the Adam optimizer (Kingma and Ba, 2015) that keeps a moving average of past parameter gradients and their squared values. The Adam optimizer computes individual step sizes for each network parameter.
epochs	The number of epochs determines how often the training data are run through during network training.

regularization	Regularization is a popular way to reduce overfitting and limit the model complexity (Bishop, 2006). It can be realized by adding a penalty term to the loss function. The simplest regularizer is the L_2 (or quadratic) regularizer that adds the penalty term $\lambda\|\theta\|_2^2 = \lambda\theta^\top\theta$ to the loss, where θ is the vector of the network parameters. The resulting loss function is given by $$\mathcal{L}_{new}\left(y, f_\theta(x)\right) = \mathcal{L}\left(y, f_\theta(x)\right) + \lambda\|\theta\|_2^2,$$ where λ is the chosen regularization parameter.
mini batch size	Instead of updating the neural network for all data points at once, the training data can be split into multiple batches. The training routine is then applied to every mini-batch. Larger mini-batch sizes speed up the training and reduce the influence of outliers, while smaller mini-batch sizes allow more training steps.
data shuffling	Shuffling the data before every new epoch increases the diversity during network training. Also, it eliminates any bias that might be introduced into the neural network by repeatedly running through the training data in a fixed order during training.
learning rate	The learning rate determines the step size of the parameter updates in the training procedure (cf. equation (2.9)). During training, the parameters might skip their optimum and even diverge if the step size is too large, while never attaining it or at least converging needlessly slowly if the step size is too small (Duda et al., 2001). Therefore, it is important to find a balanced step size. After a chosen initial value, the learning rate can be continuously adapted during training. It is often set to decrease with time.
parameter initialization	All network parameters need to be initialized before the optimization algorithm starts. Normally, the parameters are drawn from a fixed distribution to ensure a uniform learning process (Duda et al., 2001). Such distributions include uniform and normal distributions centered around zero. Very popular are the Glorot (or Xavier) initialization (Glorot and Bengio, 2010) and the He (or Kaiming) (He et al., 2015) initialization that are both based on uniform distributions around zero.

2.2.3 Uncertainty quantification

Epistemic vs. aleatoric uncertainty. The pursuit of trustworthy machine learning shows the importance of active research in the area of uncertainty quantification in deep learning (Gawlikowski et al., 2021; Zhou et al., 2021; Abdar et al., 2021; Schmähling et al., 2021). Sources of uncertainty include imperfect model training, out-of-distribution data, noisy data and unexpected shifts in the data (Kendall and Gal, 2017; Ovadia et al., 2019). The *total* (or *predictive*) *uncertainty* of a new prediction can be divided into an *aleatoric* and an *epistemic*

part (Gal, 2016; Hüllermeier and Waegeman, 2021). The aleatoric uncertainty is sometimes called *data* or *irreducible uncertainty* because it corresponds to the noise included in the data which is irreducible. In contrast, the epistemic uncertainty is also called *model* or *reducible uncertainty* because it could in principle be arbitrarily reduced by enlarging the amount of training data. An example is given in Figure 2.10. Some basic concepts and methods are introduced in the following.

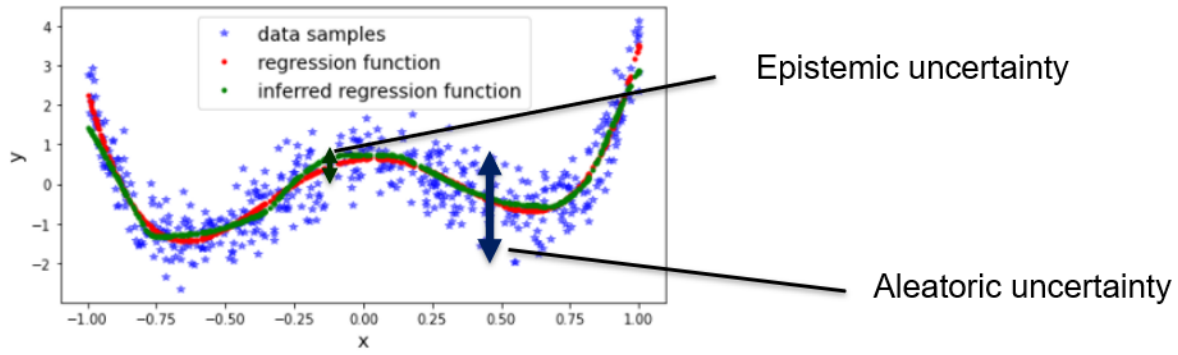

Figure 2.10: The aleatoric uncertainty refers to the irreducible noise included in the data (blue stars), while the epistemic uncertainty describes the distance of the inferred regression function (green curve) to the underlying ground truth (red curve).

Assume the following regression model with univariate target value. Let $D = \{(x_i, y_i) \in \left(\mathbb{R}^{d_x} \times \mathbb{R}\right) : i = 1, \ldots, N\}$ be the training dataset, such that it holds $\forall (x, y) \in D$:

$$y = f(x) + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2),\ x \in \mathbb{R}^{d_x}, \sigma > 0, \tag{2.15}$$

where $f : \mathbb{R}^{d_x} \to \mathbb{R}$ is the underlying regression function and ϵ is some Gaussian noise with zero mean and variance σ^2. Assume that for the chosen neural network architecture there exists a set of parameters θ^* such that the output of the neural network f_{θ^*} equals the underlying regression function, i.e. $\forall (x, y) \in D$:

$$y \sim \mathcal{N}(f_{\theta^*}(x), \sigma^2). \tag{2.16}$$

Frequentist approach. The *likelihood* function is the probability distribution for the observed data D viewed as a function of the parameters θ. It is given by the joint distribution of the observed data samples given the model parameters, i.e.

$$L(\theta) := p(D|\theta) = \prod_{i=1}^{N} p(y_i|x_i, \theta),\ \text{where } y|x, \theta \sim \mathcal{N}(f_\theta(x), \sigma^2),\ \forall (x, y) \in D. \tag{2.17}$$

The *maximum likelihood estimate (MLE)* of the parameters θ is a maximizer of the likelihood function $p(D|\theta)$ (Duda et al., 2001). Maximizing the likelihood with respect to θ is equivalent to maximizing the logarithm of the likelihood, because the logarithm is monotonically increasing, i.e. maximizing the *log-likelihood*:

$$l(\theta) := \log\left(L(\theta)\right) = -\frac{1}{2} \sum_{i=1}^{N} \left[\frac{(f_\theta(x_i) - y_i)^2}{\sigma^2} + \log\left(2\pi\sigma^2\right) \right]. \tag{2.18}$$

In practice, instead of maximizing the log-likelihood, often the negative log-likelihood is minimized.

Note that if σ^2 is already known or fixed beforehand, minimizing the negative log-likelihood is the same optimization problem as minimizing the MSE (cf. equations (2.7) and (2.8)). The variance σ^2 describes the aleatoric part of the uncertainty of the prediction of a new observation y. If unknown, it can be estimated as an additional parameter during the training process. In a heteroscedastic setting, it can also be included as an additional output of the neural network, i.e. $\sigma_\theta^2(x)$. The formalism can be derived analogously for a multivariate output where the variance is replaced with a covariance matrix. In practice, often a diagonal covariance matrix is assumed for simplicity.

Bayesian approach. In contrast to the frequentist approach (i.e. the MLE), in *Bayesian statistics* (O'Hagan and Forster, 2004; Gelman et al., 1995; Duda et al., 2001; Bishop, 2006) the model parameters θ are treated as random variables. The aim is to determine (or approximate) the *posterior* distribution $\pi(\theta|D)$. Therefore, *Bayes' theorem* plays a central role in Bayesian statistics. It has the following form:

$$\pi(\theta|D) = \frac{p(D|\theta)\pi(\theta)}{p(D)}, \tag{2.19}$$

where the *prior* $\pi(\theta)$ can be chosen to include some prior knowledge about the network parameters and the so-called *evidence* in the denominator is the normalizing constant $p(D) = \int p(D|\theta)\pi(\theta)d\theta$. Hence, the posterior is proportional to the product between the likelihood and the prior.

The *maximum a posterior (MAP)* determines the most probable model parameter values for the given data D in a Bayesian setting, by maximizing the posterior distribution. Assume the following prior distribution:

$$\theta \sim \mathcal{N}\left(0, \lambda^{-1} I_{d_\theta \times d_\theta}\right), \ \lambda > 0, \tag{2.20}$$

where d_θ is the dimension (equal to the number) of the model parameters. Then, it follows by using equations (2.18) and (2.19) that the MAP is given by

$$\arg\max_\theta \pi(\theta|D) = \arg\min_\theta \left\{ \sum_{i=1}^{N} \left[\frac{(f_\theta(x_i) - y_i)^2}{\sigma^2} + \log\left(\sigma^2\right) \right] + \lambda \|\theta\|_2^2 \right\}. \tag{2.21}$$

Note, that the MAP is equal to the MLE plus an L^2-regularization term for the model parameters that comes from the prior distribution. Again, the variance $\sigma_\theta^2(x)$ could be an additional output of the neural network in a heteroscedastic scenario.

The MLE and the MAP are only point estimates. *Bayesian neural networks* (Goan and Fookes, 2020; Jospin et al., 2020) derive a posterior distribution $\pi(\theta|D)$ for the neural network parameters. However, one problem is that the integral in the denominator of the Bayes' formula in equation (2.19) is impossible to compute in practice, and the same applies to quantities summarizing the posterior such as the posterior mean or variance.

Variational inference. *Variational inference* (Blei et al., 2017) is a popular method for approximate Bayesian inference in deep learning. For instance, the posterior can be approximated (assuming the density distribution exists) by a chosen parameterized distribution $q(\theta)$, i.e.

$$q(\theta) \approx \pi(\theta|D). \tag{2.22}$$

The parameters of q can be determined by minimizing the distance between the true posterior $\pi(\theta|D)$ and the approximation $q(\theta)$.

A popular measure for the distance between the approximated and the true posterior of the neural network parameters is the *Kullback-Leibler (KL) divergence* (Kullback and Leibler, 1951), which is non-negative but asymmetric, and is defined as follows:

$$\mathrm{KL}\left(q(\theta)\|\pi(\theta|D)\right) := \int q(\theta) \log\left(\frac{q(\theta)}{\pi(\theta|D)}\right) d\theta = \mathbb{E}_q\left[\log\left(q(\theta)\right)\right] - \mathbb{E}_q\left[\log\left(\pi(\theta|D)\right)\right]. \tag{2.23}$$

Instead of directly minimizing the intractable KL divergence between the approximated and the true posterior, often the *evidence lower bound (ELBO)* is maximized in practice (Blei et al., 2017), i.e.

$$\mathrm{ELBO}(q) := \mathbb{E}_q\left[\log\left(p(D|\theta)\right)\right] - \mathrm{KL}\left(q(\theta)\|\pi(\theta)\right) = \log\left(p(D)\right) - \mathrm{KL}\left(q(\theta)\|\pi(\theta|D)\right). \tag{2.24}$$

Uncertainty methods. Examples of deep learning techniques that approximate the posterior of the model parameters via variational inference in combination with the KL divergence as the distance measure are Bayes by Backprop (Blundell et al., 2015), normalizing flows (Rezende and Mohamed, 2015) or dropout (Kingma et al., 2015; Gal and Ghahramani, 2016). Other examples of Bayesian inference for neural networks include the Laplace approximation (MacKay, 1992; Ritter et al., 2018) or Markov chain Monte Carlo for small network sizes (Neal, 1992). Another class of methods widely used in machine learning for uncertainty quantification are *ensemble learning* techniques that produce a better generalization performance and a more robust model prediction (Sagi and Rokach, 2018; Ganaie et al., 2021; Gawlikowski et al., 2021). Their basic idea is to combine the predictions of multiple models. Ensemble learning methods include both frequentist (Carney et al., 1999; Tiwari and Chatterjee, 2010) and Bayesian (Wilson and Izmailov, 2020; Pearce et al., 2020) elements.

GUM. The *guide to the expression of uncertainty in measurement (GUM)* (BIPM et al., 2008c) is an internationally recognized guide that introduces general rules for the evaluation and expression of uncertainties in measurements. According to the GUM (BIPM et al., 2008c), the *uncertainty (of measurement)* is defined as a

> "parameter, associated with the result of a measurement, that characterizes the dispersion of the values that could reasonably be attributed to the measurand."

Such a parameter could be a standard deviation of a probability density function that accounts for all known sources of uncertainty.

The aim of the GUM is to provide general guidance for the understanding and evaluation of uncertainty in measurement and has been successfully applied to various examples from a wide range of applications. It establishes rules of how to propagate uncertainties from the

input quantities to the output quantities. This propagation of uncertainties requires that a mathematical model has been specified which relates the output quantity to the input quantities in a unique way. Furthermore, uncertainties need to be assigned to the input quantities. In addition to the basic GUM document (BIPM et al., 2008c), the supplement GUM-S1 (BIPM et al., 2008a) establishes the propagation of uncertainty distributions via a Monte Carlo method and the supplement GUM-S2 (BIPM et al., 2008b) introduces multivariate output quantities.

The treatment of regression problems according to GUM is not uniquely defined (Elster, 2014). Recently, a new part was added to the GUM (BIPM et al., 2020) that also considers Bayesian statistics and discusses the quantities involved in the measurement model. However, a Bayesian uncertainty analysis employs standard treatments such as the use of Markov chain Monte Carlo methods which do not scale to high-dimensional problems. Here, we consider a high-dimensional regression problem where the chosen regression model is a nonlinear deep neural network having more model parameters than data samples are available. For these reasons the GUM will not be further discussed in this thesis.

2.2.4 Deep ensembles

Deep ensembles (Lakshminarayanan et al., 2017) are an ensemble learning method which is currently seen as providing the state-of-the-art for uncertainty quantification in deep learning (Caldeira and Nord, 2020; Gustafsson et al., 2020; Ovadia et al., 2019; Henne et al., 2020; Ashukha et al., 2020). Multiple neural networks are independently trained and their prediction is treated as a uniformly weighted Gaussian mixture model.

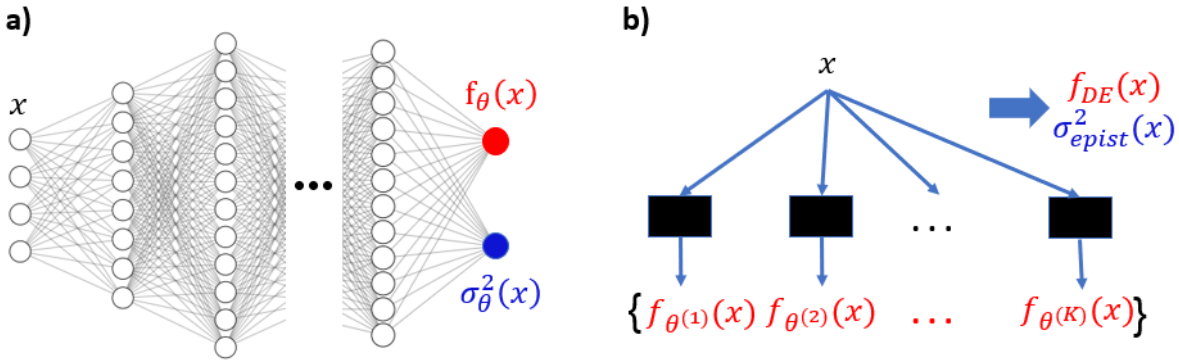

Figure 2.11: a) shows a neural network that has two output neurons (LeNail, 2019), i.e. the estimated regression function f_θ and the corresponding estimate of the aleatoric variance σ^2_θ. **b)** indicates how the individual neural networks (illustrated as black boxes) act together to form the deep ensembles estimation of the regression function $f_{DE}(x) = \frac{1}{K}\sum_{k=1}^{K} f_{\theta^{(k)}}(x)$ and the epistemic variance $\sigma^2_{epist}(x) = \frac{1}{K}\sum_{k=1}^{K}\left(f_{\theta^{(k)}}(x) - f_{DE}(x)\right)^2$.

Assume the univariate statistical model from (2.15) and let K be the number of neural networks included in the deep ensemble. Each ensemble member learns to estimate the underlying regression function f_θ and the corresponding variance of the noise included in the data σ^2_θ which is related to the aleatoric part of the uncertainty, cf. Figure 2.11a. The neural networks are independently trained using the negative log-likelihood function (2.18) as the loss function. Figure 2.11b indicates the prediction behaviour of the deep ensembles. For an input x, the estimate of the underlying regression function $f_{DE}(x)$ and the estimated variance related to the epistemic uncertainty $\sigma^2_{epist}(x)$ are the average and the uncorrected variance of

the individual estimations of the neural networks $f_{\theta^{(k)}}(x)$, $k = 1, \ldots, K$, respectively, i.e.

$$f_{DE}(x) = \frac{1}{K}\sum_{k=1}^{K} f_{\theta^{(k)}}(x),\ \sigma^2_{epist}(x) = \frac{1}{K}\sum_{k=1}^{K}\left(f_{\theta^{(k)}}(x) - f_{DE}(x)\right)^2. \tag{2.25}$$

The variance of the total uncertainty of the prediction for an input x is the following:

$$\sigma^2_{total}(x) = \sigma^2_{epist}(x) + \sigma^2_{aleat}(x),\ \text{where } \sigma^2_{aleat}(x) = \frac{1}{K}\sum_{k=1}^{K}\sigma^2_{\theta^{(k)}}(x). \tag{2.26}$$

This thesis focuses on deep ensembles because they scale well to high-dimensional data, outperform other modern methods and are straightforward to implement (Beluch et al., 2018; Gustafsson et al., 2020; Scalia et al., 2020). Deep ensembles have originally been proposed as a frequentist heuristic but can also be viewed from a Bayesian perspective which is discussed in Chapter 6 where a natural extension to the deep ensembles is proposed as well.

2.3 Inverse problems

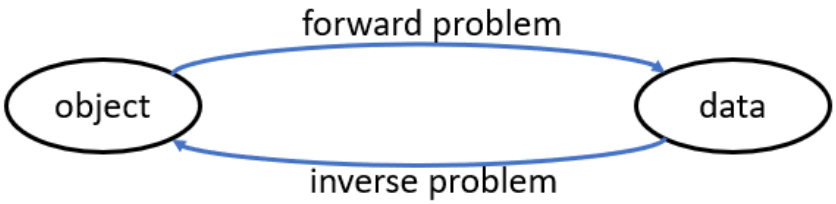

Figure 2.12: The forward problem is often defined as obtaining a model for the measurement data from a given object and the inverse problem is about finding the underlying object given the data and the forward model.

Consider the following mapping for an object x, some data y and an operator F:

$$y = F(x). \tag{2.27}$$

The *forward problem* is defined as finding y for a given x while the *inverse problem* is defined as finding x for a given y (Aster et al., 2018) as illustrated in Figure 2.12. Inverse problems are very common in science and engineering (Yaman et al., 2013; Bertero et al., 2021). Examples comprise determining the hypocenter of an earthquake from observed seismic data, X-ray computed tomography (CT), or reconstructing a topography from collected interferometrical measurement data.

2.3.1 Types of inverse problems

When x and y can be expressed as finite dimensional vectors and the operator F is a mapping f between two finite dimensional spaces (cf. equation (2.27)), the inverse problem is said to be *discrete*. If, however, x and y are functions of continuous variables, the inverse problem is called *continuous*. Continuous inverse problems are often discretized, since numerical methods require a discretization of the inverse problem to compute an approximate solution (Richter, 2020). Inverse problems can also be divided into *linear* and *nonlinear* inverse problems. The

treatment of linear problems is well understood (Aster et al., 2018). In contrast, nonlinear inverse problems are more challenging. Different approaches should be chosen depending on the concrete problem. Many solution techniques comprise some form of linearization.

In discrete linear inverse problems, the operator F can be expressed as a matrix, i.e.

$$Fx = y,\ x \in \mathbb{R}^{d_x}, y \in \mathbb{R}^{d_y}, F \in \mathbb{R}^{d_y \times d_x}. \tag{2.28}$$

Some continuous linear inverse problems can be expressed as a *Fredholm integral equation of the first kind (IFK)* with kernel k, i.e.

$$\int_a^b k(z, \xi)x(\xi)d\xi = y(z). \tag{2.29}$$

It is possible to discretize IFKs, for example via quadrature rules to obtain a linear system of equations that is easier to solve (Aster et al., 2018).

2.3.2 Solving ill-posed problems

A problem is *well-posed in the sense of Hadamard* if there exists a unique solution and the solution depends continuously on the data (Hadamard, 1902; Engl et al., 1996; Bertero et al., 2021). The non-existence of a solution occurs if the system of equations to be solved is inconsistent. This may happen when there are more data points y available than elements to be estimated x and the data are noisy (e.g. measurement data). Next, if there exists a solution, it might not be unique. Non-uniqueness of a solution occurs if the null space of a linear operator F is nontrivial. It is a characteristic of rank deficiency in discrete linear inverse problems that can be induced, for example, by discretizing a continuous inverse problem. The third property of well-posed problems in the sense of Hadamard is the continuous dependence of the solution x on the data y, also described as the stability of the solution (Aster et al., 2018). A solution is unstable if small changes in the data drastically change the solution. Such an issue can arise for discrete inverse problems when F has small singular values and is therefore ill-conditioned, since the solution can then be very sensitive to noise in the data.

In summary, inverse problems are often ill-posed because of various reasons. Some remedies for discrete linear inverse problems (2.28) are discussed in the following (Engl et al., 1996; Aster et al., 2018). If there does not exist a solution, it is still possible to find a best fit approximate solution by minimizing some *misfit measure*. A popular example is the *least squares* solution x_{L^2} that minimizes the L^2-norm of the residuals as follows:

$$x_{L^2} := \arg\min_x \|y - Fx\|_2 = (F^\top F)^{-1}F^\top y, \tag{2.30}$$

where the second equality is only defined if F has full column rank (since $(F^\top F)^{-1}$ does not exist otherwise). Note that the least squares solution can be viewed as a recommended solution from a statistical point of view if the data noise is normally distributed (cf. equation (2.18)).

If the solution is not unique, an additional constraint can be optimized. A typical example is the least squares solution of (2.28) that has the smallest L^2-norm. Let $F = USV^\top$ be the singular value decomposition of F, where $U \in \mathbb{R}^{d_y \times d_y}$ and $V \in \mathbb{R}^{d_x \times d_x}$ are orthogonal matrices and $S \in \mathbb{R}^{d_y \times d_x}$ is a diagonal matrix having the singular values on its diagonal. The

Moore-Penrose pseudoinverse solution $x_\dagger$ is defined as follows:

$$x_\dagger := V_p S_p^{-1} U_p^\top y,\ V_p \in \mathbb{R}^{d_x \times p}, S_p \in \mathbb{R}^{p \times p}, U_p \in \mathbb{R}^{d_y \times p}, \tag{2.31}$$

where p is the number of nonzero singular values and S_p is the diagonal matrix containing the nonzero singular values of F on its diagonal. The Moore-Penrose pseudoinverse solution always exists and is the least squares solution of (2.28) that has the smallest L^2-norm (Engl et al., 1996). Finally, instability can be reduced by employing regularization methods, e.g. the Tikhonov regularization (Aster et al., 2018).

2.3.3 Outlook

Interested readers are referred to Engl et al. (1996), Aster et al. (2018) and Bertero et al. (2021) for more information on inverse problems. Various machine learning techniques have been developed particularly to solve inverse problems. Examples comprise deep convolutional neural networks for inverse problems in imaging (McCann et al., 2017; Lucas et al., 2018), deep residual networks for image recognition (Schmidhuber, 2015) and denoising (Zhang et al., 2017), sparse basis field expansions and hierarchical Bayesian inference for magneto- and electroencephalography (MEG/EEG) data (Haufe et al., 2008; Hashemi et al., 2021), training a regularizer with deep neural networks to obtain well-posed solutions (Li et al., 2020), solving the forward and inverse problem simultaneously through inverse neural networks (Hagemann and Neumayer, 2021), or mixture density networks for non-unique solutions (Bishop, 1994).

The application from computational optical form measurements considered here (cf. Section 2.1) constitutes a nonlinear continuous inverse problem of reconstructing a specimen surface from measurement data. Therefore, it can generally be assumed to be ill-posed. In Chapter 3, the numerical forward model is introduced that can be used to solve the inverse problem, as discussed in more detail in Section 3.2. Towards this end, the known design topography of the specimen provides a well-informed first guess. The inverse problem is discretized by choosing a parameterized model class for the inverse operator to approximate the specimen topography with finitely many parameters to be optimized. In the state-of-the-art reconstruction procedure described in Subsection 3.2.1, a sequence of polynomial functions up to a certain order that are orthogonal on the unit disc are chosen as the model class. In this thesis, a model class of deep neural networks is proposed (cf. Chapter 4).

Of course, only a finite number of data points are available due to the limited camera resolution of 2048×2048 pixels. Problems might arise, if the measurement data are noisy or the numerical forward model does not equal the underlying physical forward model. In these cases there might not be a solution. Also, positioning errors and form errors can be confounding (Gronle et al., 2022). If both are estimated at the same time, the solution might not be unique. Only optical (smooth) surfaces (cf. Subsection 2.1.1) are considered for measurement by the TWI to ensure the uniqueness of the solution up to the positioning problem.

3

Virtual tilted-wave interferometer

The forward model for the inverse problem of the TWI surface form reconstruction procedure is provided by numerical simulations of the experiment, i.e. by a *virtual TWI*. It refers to a computer model of the optical system that imitates the behavior of a physically given TWI. Measurement data from a real specimen measured by the physical TWI can be related to simulated data computed by the virtual TWI. Basically, the virtual TWI has two different roles. First, it plays the role of the forward model in an inverse problem from accurate optical form measurements (Fortmeier et al., 2014). Second, it can be applied in order to explore the behaviour of procedures used to solve the inverse problem. Examples include analyzing the influence of error sources and limitations of the evaluation process (Fortmeier et al., 2013), exploring sources of uncertainty in the reconstruction, distinguishing sensitive system parameters (Scholz et al., 2022), generating a database for machine learning applications and many more.

This chapter is structured as follows. Some simulation tools relevant for this work are presented in Section 3.1, e.g. the simulation toolbox used to realize the virtual TWI, the orthogonal sequence of polynomials employed for modeling topographies, and the wavefront manipulators that are applied to phenomenologically alter the light path through the virtual optical system to improve the quality of the numerical forward model. Afterwards, the use-cases of the virtual TWI are discussed in Section 3.2. The inverse problem of the form reconstruction is introduced, the state-of-the-art evaluation procedure is summarized, and the data generation is explained that is needed for the training and validation of the deep neural networks. A short summary is given in Section 3.3.

3.1 Simulation tools for virtual experiments in optics

Virtual experiments are crucial for scientific progress (Cooper et al., 2015). In optics, virtual experiments allow to rigorously analyze optical measurement techniques (Thalhammer and Wachutka, 2003a). Examples include optimizing the design of experiment (Thalhammer and Wachutka, 2003b), computing expected accuracies (Wiegmann et al., 2011), analyzing the

effects of additional measurement information (Fortmeier et al., 2016) or applying neural networks for misalignment corrections (Zhang et al., 2019).

The evaluation procedure of the tilted-wave interferometer (TWI) is based on both measurement data and a numerical forward model (Liesener, 2007; Garbusi et al., 2008; Baer et al., 2014c). Therefore, a virtual environment is needed to simulate the optical system, which is realized at the PTB using the simulation toolbox SimOptDevice (Schachtschneider et al., 2019; Fortmeier et al., 2022). The simulated optical system has to be adapted to the real measurement system, i.e. calibrated, prior to the reconstruction procedure (Baer et al., 2014a; Fortmeier, 2016).

Based on virtual experiments, Mühlig et al. (2014) and Baer (2016) showed that the TWI calibration procedure is drastically improved by using multiple reference spheres instead of a single one. Steps towards tracing the measurement results to the SI units with low uncertainty are discussed in Schulz et al. (2015) and Fortmeier et al. (2017). Baer (2016) and Harsch et al. (2019) investigated the effect of positioning errors on the TWI measurement using Monte Carlo simulations. Fortmeier et al. (2016) and Harsch et al. (2019) concluded that an additional length measuring interferometer to determine the exact position of the specimen in the optical axis can drastically improve the measurement results. In Scholz et al. (2022), the virtual TWI is used to identify high-impact parameters. For the set of investigated parameters in that study, errors of the wavelength used for the illumination, errors of the radius of one of the calibration spheres, as well as the positioning errors of the specimen and noise in the input data showed greatest impact on the measurement result.

The section is structured as follows. Subsection 3.1.1 briefly introduces the simulation toolbox SimOptDevice. It was developed at the PTB, contains a realization of the virtual TWI, and is used for all the TWI-related simulations throughout this thesis. Subsection 3.1.2 presents the Zernike polynomials which are another building block for the data generation for later machine learning applications. Then, the wavefront manipulators are discussed in Subsection 3.1.3 that are needed to better adapt the virtual optical system to the real optical system of the TWI.

3.1.1 Simulation toolbox SimOptDevice

Virtual experiments can help to develop, improve, analyze and might even be part of complex measurement systems (Bergmann et al., 2016; Scholz et al., 2022). SimOptDevice is a simulation toolbox for optical virtual experiments (Fortmeier, 2016; Stavridis et al., 2018; Schachtschneider et al., 2019). It has been developed at the Physikalisch-Technische Bundesanstalt (PTB) and is implemented in MATLAB®.

The SimOptDevice enables the geometric setup of measuring devices containing axes, pivots, sensors, and optical surfaces such as mirrors, lenses, collimators, etc. Basic features include coordinate system transformations, ray tracing and ray aiming. The latter two are illustrated in Figure 3.1. Given a starting point $p_0 \in \mathbb{R}^3$ and angle $e_0 \in \mathbb{R}^3$, the aim of *ray tracing* is to trace the beam path through the optical system and to compute the intersection point $p_n \in \mathbb{R}^3$ of the ray with a certain interface, e.g. the camera. In contrast, *ray aiming* is the inverse problem of finding the direction e_0 that leads a ray through the optical system from point p_0 to point p_n, e.g. from one source to a certain pixel on the camera. More detailed

information about the implementation are provided in Fortmeier et al. (2014) and Fortmeier (2016).

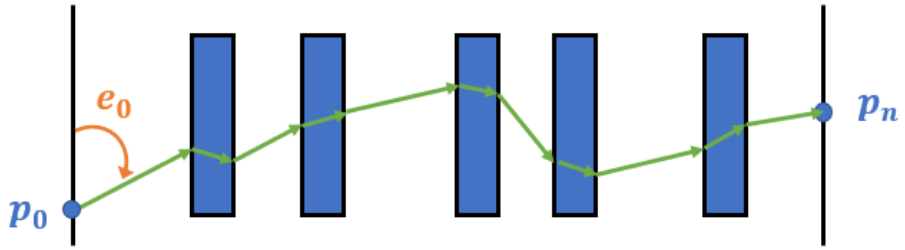

Figure 3.1: Exemplified path of a light ray through an optical system (green). The ray starts at p_0 in the direction of e_0, passes through the optical system with several optical elements (simplified as blue boxes) and arrives at p_n.

Applications of SimOptDevice include the tilted-wave interferometer (Fortmeier et al., 2022), PTB's double-ended interferometer (Fischedick et al., 2021), modulation transfer functions in optomechanical simulations (Schenker et al., 2020), sub-aperture interferometry measurements of a synchrotron mirror (Wiegmann et al., 2011) and small angle deflectometry (Schulz et al., 2010).

3.1.2 Zernike polynomials

Zernike polynomials form a sequence of orthogonal polynomial functions defined over the unit circle and are well known in optics from wavefront modeling (Wang and Silva, 1980). The following definition is based on the ANSI Z80.28-2017 (2017). A *Zernike polynomial function* Z_n^m with radial integer index n and signed azimuthal index m is defined as follows:

$$Z_n^m(\rho,\phi) := R_n^{|m|}(\rho)M(m\phi),\ \rho\in[0,1],\ \phi\in[0,2\pi], \tag{3.1}$$

where the *radial term* $R_n^{|m|}(\rho)$ and the *azimuthal term* $M(m\phi)$ are defined in the following:

$$R_n^{|m|}(\rho) := \begin{cases} \sum_{s=0}^{\frac{1}{2}(n-|m|)} \frac{(-1)^s(n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!}\rho^{n-2s} & \text{if } (n-|m|) \text{ is even} \\ 0 & \text{otherwise} \end{cases}, \tag{3.2}$$

$$M(m\phi) := \begin{cases} \cos(m\phi) & \text{if } m\geq 0 \\ \sin(-m\phi) & \text{otherwise} \end{cases}. \tag{3.3}$$

The *radial parameter* ρ is defined as $\rho=\frac{r}{a}$ for any radial distance r from the aperture center, where a is the value of the aperture radius. The *azimuthal parameter* ϕ expresses the angle between 0 and 2π.

A surface T parameterized by Zernike polynomials can be represented by

$$T(\rho,\phi)=\sum_{n,m} c_n^m Z_n^m(\rho,\phi),\ \rho\in[0,1],\ \phi\in[0,2\pi], \tag{3.4}$$

where $c_n^m\in\mathbb{R}$ are the *Zernike coefficients* associated with the corresponding Zernike polynomial functions Z_n^m. Instead of the double indexing, the Zernike polynomial functions can be ordered

using a single index j provided by

$$j = \frac{n(n+2)+m}{2}. \tag{3.5}$$

Also, any parameterized surface can be represented in a Cartesian coordinate system as follows:

$$T(x,y) = \sum_j c_j Z_j(x,y), \tag{3.6}$$

where (x, y) is defined on the unit disc. Figure 3.2 shows the first 15 Zernike polynomial functions. The frequency of the modeled surface increases with higher orders.

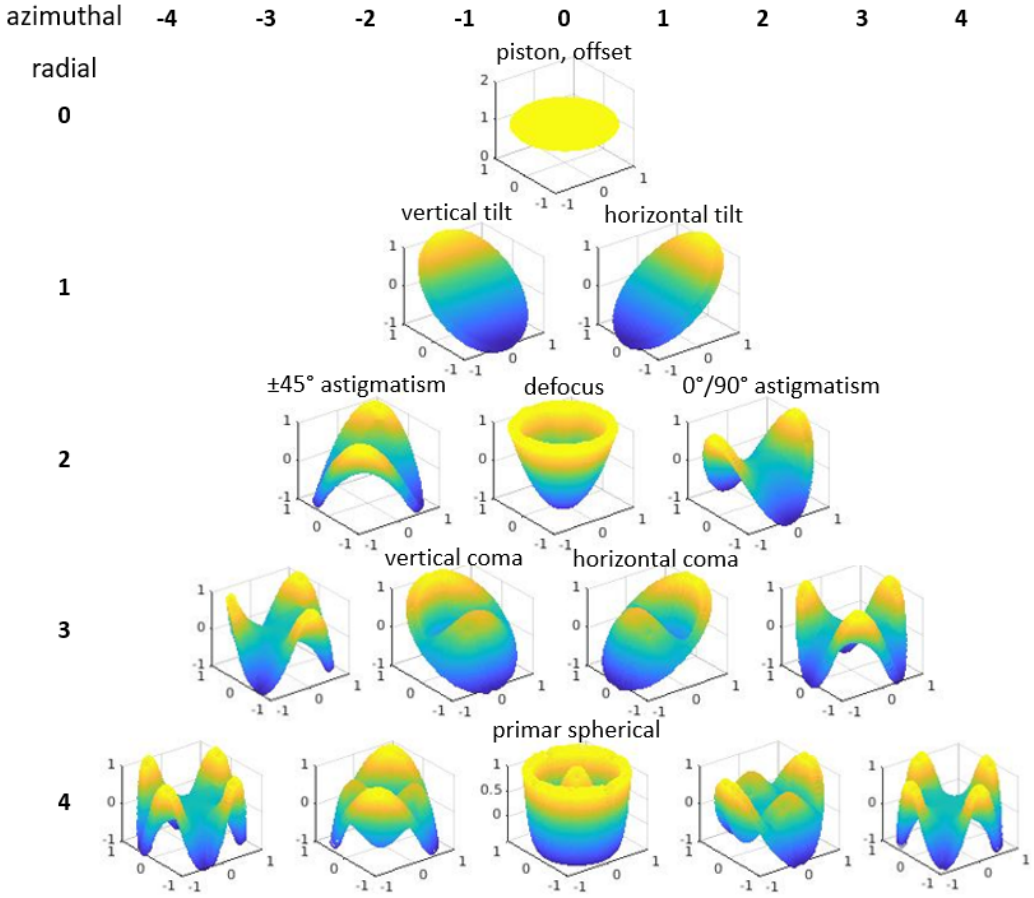

Figure 3.2: The first 15 Zernike polynomials (i.e. $n, |m| \leq 4$) functions are plotted in dependence of their azimuthal and radial indices together with their names commonly used in optics.

3.1.3 Wavefront manipulators

The[1] virtual TWI differs in general from the physical TWI, because there are deviations of the manufactured system from the instrument design. To attain an accuracy that is fit for purpose, the virtual TWI can be refined, i.e. *calibrated* using the observed data for some test specimens whose topographies are known with high accuracy (Baer et al., 2014a; Fortmeier et al., 2022). An example is shown in Figure 3.3a. The calibration is realized by adding optical path lengths at two virtual planes R_1, R_2, called *wavefront manipulators*, to the virtual TWI, as shown

[1] Part of this section is taken from Hoffmann et al. (2021b).

in Figure 3.3b. The light beam passes the first wavefront manipulator R_1 before attaining the topography and passes the second wavefront manipulator R_2 after having attained the topography.

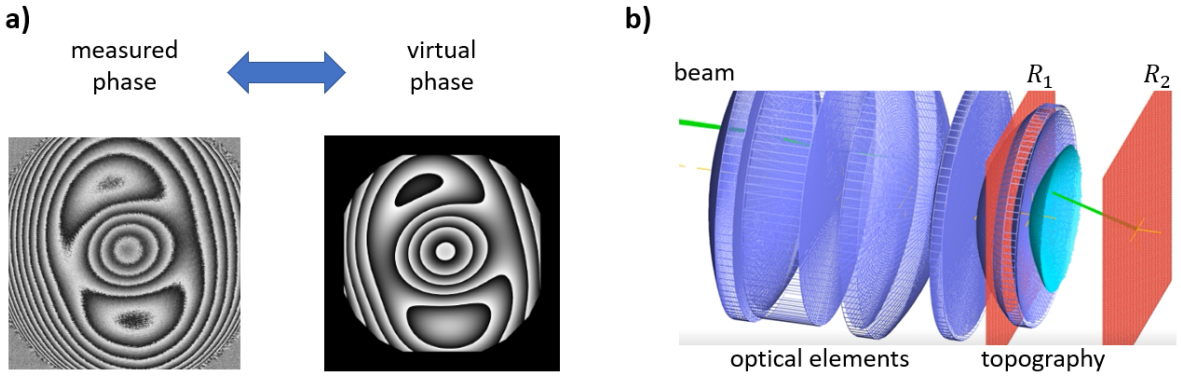

Figure 3.3: a) The goal of the calibration process is to adapt the simulated data to the measured data for some well-known test topographies. **b)** The light path through the optical system is deviated by introducing the wavefront manipulators R_1 and R_2 before and after the topography, respectively. Figure b) is taken from Hoffmann et al. (2021b).

Each virtual wavefront manipulator modifies the light beams phenomenologically by changing the light path through the optical system. More precisely, the optical path lengths L_{R_1} and L_{R_2} of the wavefront manipulator R_1 and the wavefront manipulator R_2, respectively, are added to the optical path length between the two wavefront manipulators L_T. Then, the total optical path length difference (OPD) is computed by subtracting the optical path length of the reference arm L_R (cf. Figure 2.4) from the resulting sum as follows (Fortmeier, 2016):

$$OPD = L_{R_1} + L_{R_2} + L_T - L_R. \tag{3.7}$$

Each wavefront manipulator is parameterized by a double fit of Zernike polynomials explained in the following. The parameterization of the wavefront manipulator R_1 depends on the intersection (u, v) with a beam and its originating light source (U, V), i.e.

$$L_{R_1}(u, v, U, V) = \sum_{i=1}^{I} \left(\sum_{j=1}^{J} Q_{ij} Z_j(U, V) \right) Z_i(u, v), \tag{3.8}$$

where Q is the matrix of Zernike coefficients with dimension $I \times J$ and Z_i, Z_j are the Cartesian Zernike polynomials. Analogously, the wavefront manipulator R_2 depends on the intersection with a beam (m, n) and its corresponding pixel on the CCD (M, N), i.e.

$$L_{R_2}(m, n, M, N) = \sum_{k=1}^{K} \left(\sum_{h=1}^{H} P_{kh} Z_h(M, N) \right) Z_k(m, n), \tag{3.9}$$

where P is the matrix of Zernike coefficients with dimension $K \times H$ and Z_k, Z_h are the Cartesian Zernike polynomials.

Note that two wavefront manipulators are required instead of a single one to ensure that the refinement of the virtual TWI is independent of the surface under test (Fortmeier, 2016). The Zernike coefficients of the parameterized wavefront manipulators are adjusted during the calibration procedure (Fortmeier et al., 2022). This is done in such a way that the simulated

data using the virtual TWI increasingly resemble the measurement data obtained from the real optical system for some spherical calibration specimens that are known with high accuracy.

3.2 Use of the virtual tilted-wave interferometer

For an optical specimen topography T defined on a circular base area and the corresponding measurement data L (resulting from four disjoint masks on the acting 2D point source array), the following mapping is considered:

$$F(T) = L, \tag{3.10}$$

where F is the forward operator, given here numerically by the virtual TWI. The inverse problem (cf. Section 2.3) consists of reconstructing the surface form from the measurement data. It can generally be viewed as a nonlinear continuous inverse problem, since the optical surface is assumed to be a continuous function defined on the unit circle in $\mathbb{R}^2$.

Figure 3.4 summarizes the evaluation procedure of the optical form reconstruction using the TWI (Garbusi et al., 2008; Baer et al., 2013b; Fortmeier et al., 2016). The measurement data of a specimen are collected at the CCD of the TWI in the form of interferograms (cf. Subsection 2.1.3). For each of the four masks, the interferograms are transformed into a phase image using a five-step phase shifting algorithm (Hariharan et al., 1987)[2]. Then, the phase images Φ_{meas} are unwrapped to the corresponding optical path length differences (OPDs) via the Goldstein unwrapper (Goldstein et al., 1988)[2].

The known design topography T_0 can be used as a first guess for the surface form of the specimen T and the numerical forward model is given by the virtual TWI F. Then, the inverse problem translates to reconstructing the deviation of the specimen topography to its design called *difference topography* ΔT from the differences of the OPDs ΔL between the measurement data L and the simulated data $L_0 = F(T_0)$, i.e.

$$T = T_0 + \Delta T, \;\; \Delta L = L - L_0, \;\; L_0 = F(T_0). \tag{3.11}$$

Hence, the solution of the inverse problem is based on finding a map $f : \Delta L \mapsto \Delta T$.

The inverse problem can be discretized by choosing a parameterized model class $\{f_\theta\}_{\theta\in\Theta}$ to approximate the mapping f with finitely many parameters. Zernike polynomials (cf. Subsection 3.2.1) as well as neural networks (cf. Chapter 4) can be chosen as model class. The parameters are determined by solving the following minimization problem:

$$\min_{\theta\in\Theta} \|f_\theta(\Delta L) - \Delta T\|_2. \tag{3.12}$$

Note that the resulting discrete inverse problem is at the same time a regression problem.

The following subsections explain in more detail what the numerical forward model given by the virtual TWI is needed for. A brief overview of the state-of-the-art reconstruction procedure is provided in Subsection 3.2.1, where the numerical forward model is used in every iteration step of the iterative evaluation procedure. Then, Subsection 3.2.2 explains how to generate

[2] in the PTB implementation relevant for this work.

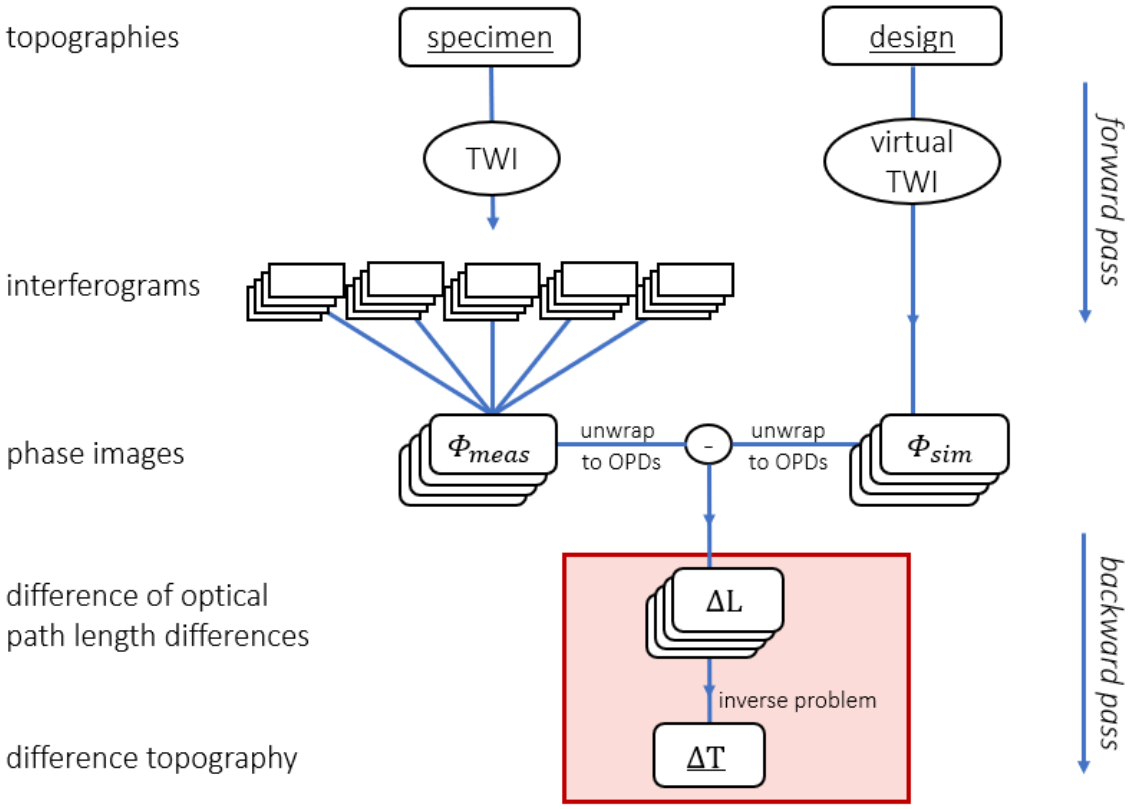

Figure 3.4: Scheme of the TWI evaluation procedure. The collected interferograms of a measured specimen are mapped to phase images Φ_{meas} that in turn are unwrapped to optical path length differences (OPDs). As a first guess, the OPDs corresponding to the design topography can be simulated by the virtual TWI. The task is now to reconstruct the difference topography ΔT from the resulting differences of measured and simulated OPDs ΔL (red box). Note that four disjoint masks were subsequently used on the acting 2D point source array (cf. Subsection 2.1.3).

a dataset for machine learning applications and especially deep neural network training for optical form measurements based on simulations. In addition, virtual experiments allow to generate out-of-distribution test data for validation and to study the generalizability of the trained models by modifying the numerical forward model, as discussed in Subsection 3.2.3.

3.2.1 State-of-the-art TWI evaluation procedure

State-of-the-art TWI evaluation methods use a two-step reconstruction procedure (Baer, 2016; Fortmeier et al., 2022). First, the model class for the discretized inverse problem (cf. the optimization problem from (3.12)) is chosen to be the set of Zernike polynomials (cf. Subsection 3.1.2) up to a fixed order M. Hence, the Zernike coefficients are the parameters to be optimized. Additional parameters such as position parameters or patch offsets can be included as well (Fortmeier et al., 2022). The nonlinear inverse problem is solved by iteratively updating the design topography T_0 using local linearizations as follows (Fortmeier, 2016):

$$T_{k+1} = T_k + \sum_{j=1}^{M} c_j^{(k)} Z_j,\ k = 0, \ldots, K, \tag{3.13}$$

where K is the number of performed iterations, Z_j and c_j, $j = 1, \ldots, M$, are the Cartesian basis Zernike polynomials and the corresponding Zernike coefficients, respectively.

The Zernike coefficients are computed by solving the following linearized discrete inverse problem via least squares (Fortmeier et al., 2014):

$$\Delta L = Jc,\ J = \frac{\mathrm{d}L}{\mathrm{d}c}, \tag{3.14}$$

where L are the measured OPDs and $c \in \mathbb{R}^M$ is the vector of Zernike coefficients. Note that the virtual TWI is used in every iteration step to calculate ΔL (cf. equations (3.11)).

Second, the remaining high-frequency parts are computed from the remaining residuals between the measured and simulated OPDs on a chosen grid and are added to the approximated topography. There exist different approaches to do so. In Baer et al. (2013b), the high-frequency surface deviation is geometrically calculated for each source independently based on the angles of incidence and the intersection points of the rays with the topography provided by the ray tracing procedure. Then, the different parts of the reconstructed surface deviations (depending on the source) are combined via a stitching procedure. In contrast, Fortmeier et al. (2022) solve an additional linearized inverse problem to reconstruct the high-frequency part via the partial derivatives of the simulated OPDs with respect to the height values of the surface under test.

3.2.2 Data generation for neural network training

This work focuses on solving the nonlinear inverse problem of reconstructing an arbitrary difference topography ΔT from the corresponding differences of OPDs ΔL for a fixed design topography (cf. Figure 3.4) via deep neural networks. Potential advantages of the deep learning approach are improved reconstruction accuracies because the nonlinear inverse problem is directly solved instead of applying local linearizations, shorter computation times since deep neural networks provide immediate results once they are trained, and easy-to-apply uncertainty quantifications. Challenges include the data generation, the neural network training, the model validation and its generalization capability to measurement data. Deep neural network training, generalization performance and validation depend on the quality (and quantity) of the available data (Nikolenko et al., 2021). Therefore, the following tackles Objective 1, i.e. to generate a database for the training and testing of deep learning methods (cf. Section 1.2), and presents the basic concepts of data generation for this work.

The inverse problem is treated as a regression problem and we choose the model class (cf. the optimization problem from (3.12)) to be a set of convolutional deep neural networks, where the model parameters are optimized using some training data D. Due to the high model complexity of deep neural networks, the pixelwise height values of a difference topography are directly predicted by the deep learning approach and no iterative procedure is applied. For a fixed design topography, the trained deep learning model can be applied directly to any specimen without further refinement. No misalignment errors are included during the neural network training to preserve the uniqueness of the reconstructed topographies by fixing the position in advance. Nonetheless, the deep learning approach is also applied to measurement data in Chapter 5.

As shown in the red box in Figure 3.4, the deep learning objective is to reconstruct the difference topographies ΔT for given differences of OPDs ΔL. The data generation process

is summarized in Figure 3.5. For a fixed design topography T_0, various specimens T can be generated by randomly drawing Zernike coefficients (cf. Section 3.1.2) and using the resulting polynomials as difference topographies ΔT. The corresponding OPDs can be simulated for a given specimen using the virtual TWI. Then, the differences of OPDs ΔL are computed by subtracting the OPDs L_0 corresponding to the design topography from the simulated OPDs L of the specimen. Note that ΔL has four channels because four disjoint masks are successively blocking the 2D point light source array (cf. Subsection 2.1.3). Figure 3.6 shows an example data point $(\Delta L, \Delta T) \in D$.

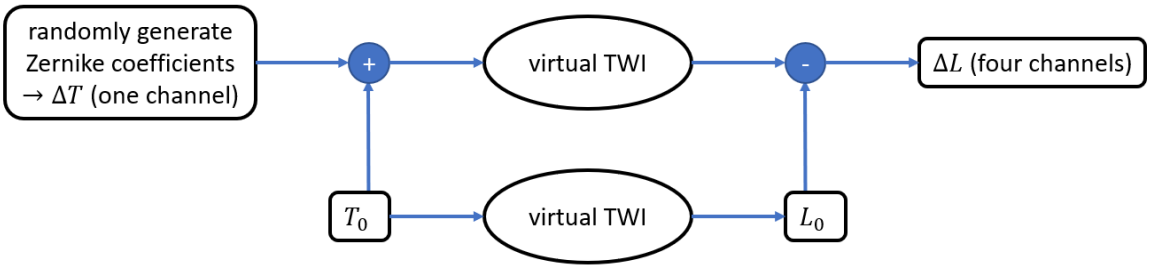

Figure 3.5: General data generation process. Adding a random difference topography ΔT to the known and fixed design topography T_0 generates a new specimen. The virtual TWI computes the corresponding OPDs of a specimen. Then, the differences of OPDs ΔL are obtained by subtracting the OPDs of the design topography L_0 from the resulting OPDs of the specimen.

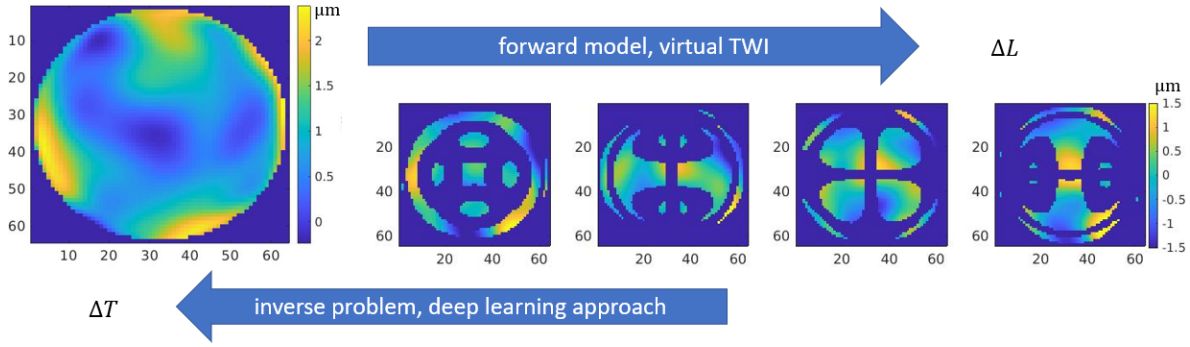

Figure 3.6: A data point consists of a difference topography ΔT and the corresponding differences of OPDs ΔL. The images have a resolution of 64×64 pixels and an underlying asphere as design topography.

In this work, two different design topographies with a circular base are employed; a two-radii surface (Fortmeier et al., 2019) and an asphere (cf. Appendix A.1). They have a base-diameter of about 40 mm and 30 mm, and a peak-to-valley value of ca. 6 mm and 4 mm, respectively. To generate different specimens, the Zernike coefficients for up to the first 136 Zernike polynomial functions are drawn randomly. An additional weighting is applied to the Zernike coefficients. The weights decrease for higher order polynomials and provide for a larger share of rotational symmetric coefficients, resulting in more realistic specimens (Schachtschneider et al., 2018; Fortmeier et al., 2020). Also, we always set the first three Zernike coefficients, that correspond to the piston and tilt parameters (cf. Figure 3.2), to zero to not introduce any positioning errors.

Up to approximately $40,000$ training data points are generated for a design topography. An example of such a data point is shown in Figure 3.6 for the asphere as design topography. The generated data have a resolution of 64×64 pixels. This resolution is enough for the purposes

of this thesis, but can be adapted in the future. For the moment it saves memory, computation time, and ensures a fast data generation process. All training data are provided using exactly the same virtual TWI for both the design topography and the specimens (cf. Figure 3.5), i.e. the virtual TWI is perfectly calibrated (cf. Subsection 3.1.3). The data generation process is implemented in MATLAB®.

3.2.3 Model validation using virtual experiments

The prediction behaviour of a (trained) deep learning model should be tested thoroughly to ensure a trustworthy output (Taylor, 2006). Therefore, all generated test data are disjoint from the training data, i.e. the model has not seen them during training. Independent test data are important because the model might have learned to overfit the training data, meaning that the predictions are very accurate on the training data but do not generalize well to data not included in the training process (Bishop, 2006).

Similar to the training data, we generate the test data using the numerically given forward model as described in the previous subsection. All training data are provided by a *perfect optical system*, where the same virtual TWI is used for the simulated data of both the design topography and the specimens (cf. Figure 3.5). A first independent test dataset is generated analogously. However, the prediction behaviour of a deep learning model is of interest not only for in-domain data, but also its generalizability to out-of-distribution data (Abdar et al., 2021). Indeed, our goal is to apply the developed deep learning approach to a real-world environment, where no perfect optical system exists. To this end, we use virtual experiments to generate multiple test sets originating from *non-perfect optical system*s, where the numerical forward model is imperfect. It follows that the employed virtual TWIs to simulate the data for the design topography and specimens, respectively, are not identical. In order to simulate an imperfect numerical forward model, we disturb the virtual TWI phenomenologically by modifying its wavefront manipulators (cf. Subsection 3.1.3).

3.3 Summary

This chapter introduces the virtual TWI as a numerical forward model for the inverse problem related to the TWI surface form reconstruction. The virtual TWI employed in this thesis is based on the SimOptDevice - a simulation toolbox developed at PTB for virtual experiments in optics. State-of-the-art TWI evaluation methods require the virtual TWI to update the reconstructed topography iteratively by solving a linearized inverse problem via least squares optimization. Here, we extend the virtual TWI to generate a database that enables the investigation of deep learning methods for the inverse problem related to the TWI, which addresses Objective 1, i.e. to generate a database for the training and testing of deep learning methods (cf. Section 1.2). Various specimens are generated using Zernike polynomial functions, and out-of-distribution data are simulated by modifying the wavefront manipulators of the virtual TWI which phenomenologically disturbs the light path through the optical system.

4

Deep learning for the tilted-wave interferometer

Deep[1] neural networks and machine learning in general are experiencing an ever greater impact on science and industry. Their application has proven beneficial in many different domains, including autonomous driving (Grigorescu et al., 2020), anomaly detection in quality management (Staar et al., 2019), signal processing (Mousavi and Baraniuk, 2017), analysis of raw sensor data (Moraru et al., 2010), medical health care (Esteva et al., 2019), and quality assurance (Kretz et al., 2020). Deep learning has also been successfully employed in optics. Examples comprise the compensation of lens distortions (Chung, 2018), correcting abberated wave fronts in adaptive optics (Vdovin, 1995), correcting misalignment errors (Baer et al., 2013a), (Zhang et al., 2020), aberration detection (Yan et al., 2018), or phase predicitons (Rivenson et al., 2018).

In this chapter, we develop a deep learning approach to solve the high-dimensional inverse reconstruction problem from computational optical form measurements based on the TWI evaluation procedure (cf. Figure 3.4). A recent study comparing different state-of-the-art techniques that measure optical freeform surfaces (Fortmeier et al., 2020) showed deviations between the participants ranging from 15 nm to 110 nm on some real specimens neglecting any spherical contribution. The aim of this chapter is to demonstrate that this field in optics can also benefit from deep learning while tackling Objectives 2 and 3, i.e. to develop a deep learning approach for the form measurement of optical aspheres and freeform surfaces and to extend it to incorporate an uncertainty quantification to the estimated solution (cf. Section 1.2). Motivated by the ultimate goal of measuring optical surfaces in a real-world scenario, we propose a hybrid approach combining deep learning methods with a state-of-the-art calibration procedure (Fortmeier et al., 2022) to refine the numerical forward model.

Recall that the estimation of the solution to the inverse problem related to the TWI evaluation procedure is treated as a regression problem and can be stated as follows. Find a map f for a fixed design topography, such that the resulting differences of OPDs ΔL map to

[1] This chapter is based on Hoffmann and Elster (2020) and Hoffmann et al. (2021b).

the corresponding difference topography ΔT, i.e.

$$f : \mathbb{R}^{C\times M\times M} \to \mathbb{R}^{M\times M},\ \Delta L \mapsto \Delta T, \tag{4.1}$$

where C is the number of input channels, i.e. the number of disjoint masks sequentially used to block parts of the 2D point light source array (cf. Subsection 2.1.3), and $M \times M$ is the image resolution.

Section 4.1 shows in a proof-of-principle study that deep learning approaches can be successfully applied to computational optical form measurements using the TWI evaluation procedure. Afterwards, an uncertainty quantification is incorporated into the developed approach in Section 4.2 via deep ensembles and the proposed method is tested thoroughly. The analysis and results presented in this chapter are based on virtual experiments with known ground truth. Finally, our findings are summarized in Section 4.3.

4.1 Computational optical form measurement

Applied[2] state-of-the-art methods solve the high-dimensional, nonlinear, inverse reconstruction problem related to the TWI evaluation procedure iteratively through local linearizations (cf. Subsection 3.2.1). A general advantage of neural networks is the ability to produce instant results once they are trained. Furthermore, it is interesting to explore whether deep learning could also improve the quality of the inverse reconstruction as a nonlinear approach. In a proof-of-principle study, this section demonstrates that computational optical form measurements can benefit from deep learning. A data-driven approach based on deep neural networks is explored to solve the inverse problem of the accurate optical measurement of optical aspheres and freeform surfaces using the TWI.

The section is organized as follows. Subsection 4.1.1 presents the employed data for training and testing. Then, Subsection 4.1.2 introduces the proposed deep learning framework and explains how to combine this approach with a conventional calibration method for better generalization performance. Subsequently, the results are shown in Subsection 4.1.3 and the outcome of the proof-of-principle study is discussed in Subsection 4.1.4.

4.1.1 Data

When solving an inverse problem with neural networks, it is a common practice to generate data through physical simulations (Lucas et al., 2018), (McCann et al., 2017). Here, the data generation is carried out as described in Subsection 3.2.2. Various difference topographies ΔT are generated through randomly chosen weighted Zernike polynomials (cf. Subsection 3.1.2). Then, they are added to a specific design topography at a fixed measurement position to create different virtual specimens. Following the forward pass, the numerical forward model given as virtual TWI (cf. Chapter 3) is used to compute the OPDs L_0 and L of the design topography T_0 and the modeled specimens $T = T_0 + \Delta T$, respectively. For a fixed design topography, the generated training dataset D is defined as follows:

$$D := \{(\Delta L, \Delta T) \,|\, \Delta T \text{ randomly generated, } \Delta L = L - L_0\}\,. \tag{4.2}$$

[2]Most of this section is taken from Hoffmann and Elster (2020).

More precisely, the OPDs are computed from the corresponding topographies using the virtual TWI while assuming the numerical forward model to be perfect.

Towards the aim of generalizing the deep learning approach to real-world data, the model is tested on in-domain data assuming a perfect numerical forward model as well as out-of-distribution data generated by simulating an imperfect forward model (cf. Subsection 3.2.3). An overview is given in Table 4.1. All training data are generated using a perfect optical system while the system conditions are varied for testing.

Table 4.1: Different optical system settings are presented that depend on the employed numerical forward models.

optical system	formula	virtual TWI description
perfect	$\Delta L = F_0(T) - F_0(T_0)$	F_0 is the numerical forward model used to generate the training data.
non-perfect	$\Delta L = F_1(T) - F_0(T_0)$	F_1 is a numerical forward model simulating the ground truth TWI, i.e. simulating measurement data.
calibrated	$\Delta L = F_1(T) - \hat{F}_1(T_0)$	$\hat{F}_1$ is a refined numerical forward model that is phenomenologically adapted to approximate the ground truth TWI.

Two different design topographies are considered, namely the two-radii freeform surface and the asphere with a base-diameter of ca. 40 mm and 30 mm, respectively. About $22,000$ data points are generated for each design. 10% of the data is used exclusively for testing and is not included in the network training. An example is shown in Figure 3.6 for the asphere as design topography with $C = 4$ and $M = 64$ in (4.1).

4.1.2 Approach

Regression problem. We develop a method that combines the TWI evaluation procedure with a data-driven deep learning approach. Therefore, the inverse problem (4.1) is treated as an image-to-image regression problem and the inverse mapping f is approximated by a parameterization f_θ with parameter space Θ, i.e.

$$f_\theta : \mathbb{R}^{C\times M\times M} \to \mathbb{R}^{M\times M},\ \Delta L \mapsto \Delta T, \tag{4.3}$$

where $M \times M$ is the dimension of the images, and C is the number of channels in the input. Note that the image dimension of the input equals the image dimension of the output here. This is not mandatory, but it suits the neural network architecture described below. The parameterized model class $\{f_\theta\}_{\theta\in\Theta}$ is determined by the neural network architecture, where θ are the learnable model parameters. Using the L_2-loss function (cf. equation (2.8)) for the model training, the following minimization problem is implicitly solved for all tuples $(\Delta L, \Delta T) \in D$:

$$\min_{\theta\in\Theta} \|f_\theta(\Delta L) - \Delta T\|_2. \tag{4.4}$$

While the CCD gives a resolution of 2048×2048 pixels, $M = 64$ is chosen here which suffices for the purpose of this work. For the asphere and the two-radii surface $C = 4$ and $C = 1$ were used, respectively. This is because the two-radii freeform surface has a big patch in the first channel which almost covers the entire CCD for the selected measurement position. Furthermore, even though some pixels are missing, the first channel sufficed for the purpose of this deep learning proof-of-principle study.

Deep neural network. Deep learning techniques became very popular for computational imaging applications (Barbastathis et al., 2019) with the introduction of convolutional networks (LeCun et al., 2015). Therefore, we choose the U-Net (Ronneberger et al., 2015) as neural network architecture. U-Nets have been successfully applied in various image-to-image deep learning applications (Işıl et al., 2019; Barbastathis et al., 2019). An example U-Net structure is shown in Figure 4.1. The input passes through several convolution and rectified linear unit layers on the left side before being reduced in dimension in every vertical connection by max pooling layers. After reaching its bottleneck at the bottom, the original data dimension is restored, step by step, through transposed convolution layers on the right side. During each dimensional increase step, a depth concatenation layer is added which links the data of the current layer to the data of the former layer with same dimension. These skip connections are depicted as horizontal lines in Figure 4.1.

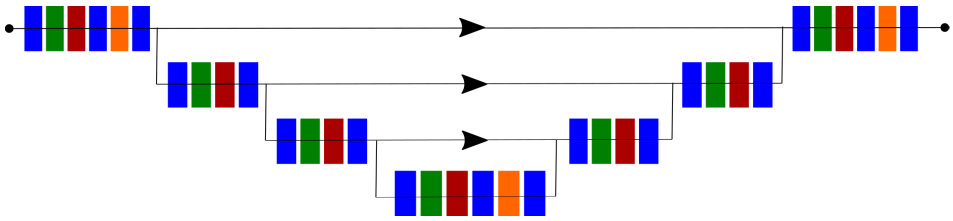

Figure 4.1: Example of a U-Net architecture taken from Hoffmann et al. (2021b).

Here, the chosen U-Net architecture consists of a total of 69 layers. The training set is used to normalize all input and output data prior to feeding them into the network. The U-Net is trained using an Adam optimizer (Kingma and Ba, 2015) and the mean squared error as the loss function. The training is carried out for the multi-radii freeform surface with an initial learning rate of 0.0005, a drop factor of 0.75 every five periods, and a mini batch size of 64. In addition, a L_2 regularization of the network parameters is employed to stabilize the training with a regularization parameter of 0.004. For the asphere, the training is carried out for 15 epochs with a mini batch size of eight samples, an initial learning rate of 0.0005 which decreased every three epochs by a learning rate drop factor of 0.5, and a regularization parameter of 0.0005. The implementation is realized in MATLAB® (version R2019b) using the Deep Learning Toolbox 14.0 (The MathWorks, 2018).

Generalization to non-perfect systems. While the great success of deep neural networks is based on their ability to learn complex relations from data without knowing the underlying physical laws, including existing physical knowledge into the models can further improve results (de Bézenac et al., 2019; Karpatne et al., 2017; Raissi, 2018). In this proof-of-principle study, we follow such an approach by developing a hybrid method which combines physical knowledge

with data-driven deep convolutional neural networks. The employed scientific knowledge is twofold; training data is generated by physical simulations using the virtual TWI (cf. Chapter 3) and a conventional calibration method (Fortmeier et al., 2022) is used to generalize the trained network to non-perfect optical systems.

The generalization procedure is motivated by the ultimate goal to apply the trained deep neural network to real-world data (cf. Objective 4 in Section 1.2 and Chapter 5). A workflow of the proposed hybrid approach is summarized in Figure 4.2a based on the nomenclature of Table 4.1. First, the selected U-Net is trained on data generated under perfect system conditions (A). In a real-world scenario, however, no perfect optical system exists. Thus, the trained model is evaluated on test data derived from a non-perfect system (B). We introduce an additional state-of-the-art calibration step into the evaluation procedure prior to the model prediction (C), where the imperfect numerical forward model is refined phenomenologically (Baer et al., 2014a; Fortmeier et al., 2022). This can be realized by inserting wavefront manipulators into the virtual TWI (cf. Subsection 3.1.3) and adapting the beam path through the optical system in a least-squares sense using known, well fabricated spherical topographies at different measurement positions to compare the OPDs measured by the TWI and its virtual counterpart. The refinement of the virtual TWI is achieved through the state-of-the-art calibration procedure implemented at the PTB (Fortmeier et al., 2022). In 4.2b, the results are summarized for the different test sets to visualize the effect and aim of the hybrid method. Note that the topographies are the same in the three test cases, but the input to the deep learning model changes. The results are discussed more precisely in the following.

a)

legend	optical system	type of data
A	perfect	training, test
B	non-perfect	test
C	calibrated	test

b)

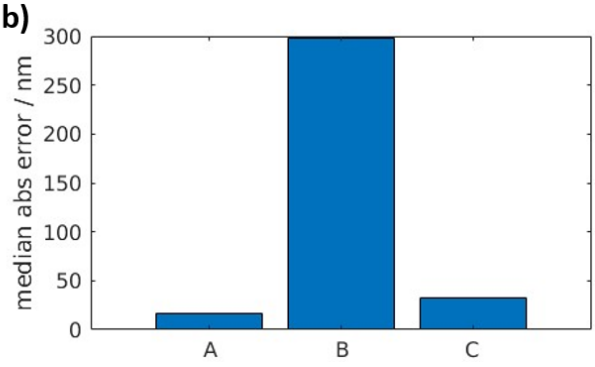

Figure 4.2: a) Workflow of the hybrid method based on the nomenclature of the numerical forward model from Table 4.1. The U-Net is trained on data generated by a perfect system (A) and tested on data from a non-perfect system (B). The hybrid method constitutes a refinement of the imperfect numerical forward model by a state-of-the-art calibration method (C). **b)** Results for the test sets from a).

4.1.3 Results

The following results are all based on simulated data. As mentioned above, two different design topographies are considered, namely an asphere and a two-radii freeform surface. First, the results for simulated data from a perfect system environment are presented. The neural networks that are trained for the design topography of an asphere and a two-radii freeform surface artefact are addressed, respectively. Next, additional strategies which could improve the models are discussed as well. Finally, the results are shown for data simulated by a

non-perfect system and compared to the results provided by the proposed hybrid method applied in a calibrated environment.

The presented results are reproducible in the sense that when repeating the simulations for similar train and test sets, essentially the same findings are observed. The topographies have a circle as base area. Since the required input and output of the deep neural network are images, the area outside of the circle shape is defined with zeros which the model learns to predict. Nonetheless, we consider only the difference topography pixels inside the circle shape in the presented results.

The root mean squared error (RMSE) of two vectors $y, \hat{y} \in \mathbb{R}^{d_y}$ and the root mean squared deviation (RMSD) of a vector y are defined as follows:

$$RMSE(y, \hat{y}) := \left(\frac{1}{d_y} \sum_{i=1}^{d_y} (y_i - \hat{y}_i)^2 \right)^{\frac{1}{2}}, \quad RMSD(y) := \left(\frac{1}{d_y} \sum_{i=1}^{d_y} y_i^2 \right)^{\frac{1}{2}}. \tag{4.5}$$

Note that the pixels of a topography form a vector. If the RMSE or RMSD relates to a set of vectors (e.g. set of topographies), then the mean over the RMSE and RMSD of the individual elements of the set is taken, respectively.

Perfect system. About 2200 samples are used for testing that were not included in the training. First, the two-radii freeform surface is considered as the design topography. Three randomly chosen prediction examples are shown in Figure 4.3a. The RMSE of the U-Net predictions on the test set is 33 nm. For comparison, the difference topographies in the test set have a total RMSD of 559 nm. The median of the absolute errors of the U-Net is about 18 nm, while the median of total absolute deviations in the test set is 428 nm.

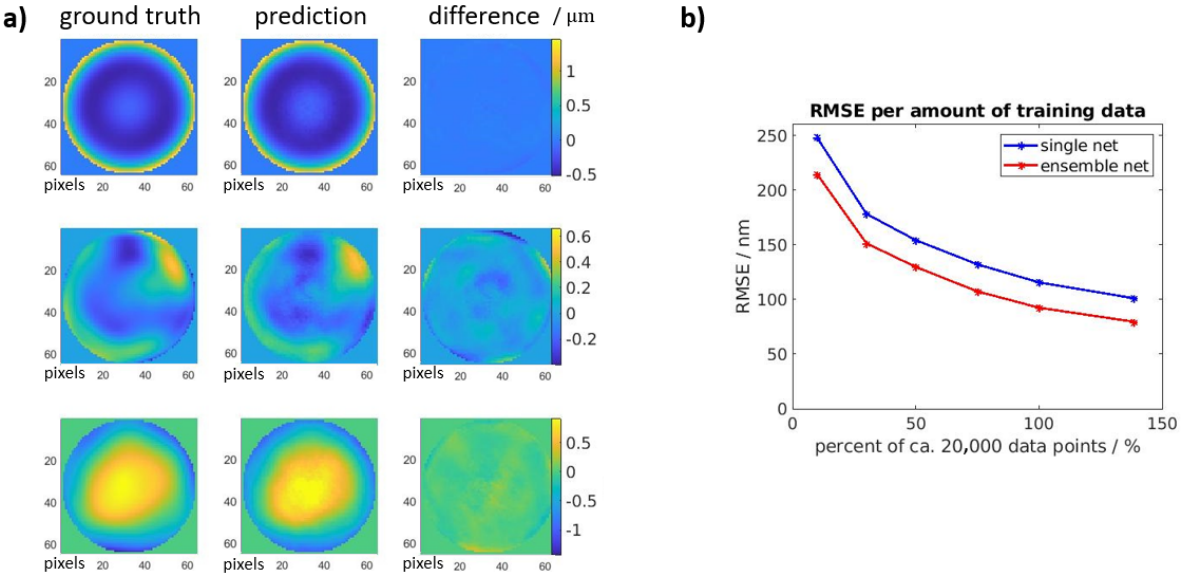

Figure 4.3: a) Three examples of predicted test difference topographies. The errors in the third column show the pointwise differences between the ground truth and the prediction. The root mean squared error (RMSE) from top to bottom are 16 nm, 84 nm, and 74 nm. The base-diameter of the topographies is about 40 mm. **b)** RMSE of a single network (blue) and a network ensemble (red), depending on the amount of data used for training. The test set is the same for all evaluations and was not included in training. Both figures are taken from Hoffmann and Elster (2020).

For the asphere as the design topography the RMSE is 102 nm, while the difference topographies in the test set have a RMSD of 589 nm. The median of the absolute errors of the predictions of the U-Net is 52 nm and the median absolute deviation of the test set is 451 nm for comparison. One possible explanation for the discrepancy in the accuracy of the predictions between the network for the asphere and two-radii freeform surface as the design topographies is the following. The input of the respective U-Nets and their resulting architecture are varying. As mentioned above, the network concerning the asphere has four input channels. These are shown in Figure 3.6. In each channel, various different areas are illuminated at the CCD, resulting in a distribution of information into different and smaller patches. The two-radii freeform surface on the other side, illuminates one big circle shaped patch in the first channel for the selected measurement position. This channel, which contains most of the important information in a single patch, forms the only input to the corresponding network.

However, the results for the asphere can be improved further. One way to do so is to increase the amount of training data. The results are shown in Figure 4.3b. As the input has four channels for the asphere, it seems natural that more data is needed for training than for the two-radii freeform surface. A second approach is to use a network ensemble (Lakshminarayanan et al., 2017) rather than a single trained network. To this end, 15 U-Nets are trained from scratch and the ensemble output is taken as the mean of the ensemble predictions. Again, the results are shown in Figure 4.3b. In this way, the accuracy improves to a RMSE of 80 nm using an ensemble of 15 U-Nets, each trained from scratch on almost 28.000 data points. It should be noted that a further improvement seems possible as the amount of data is crucial for training and the network's architecture of the asphere is more complex due to more input channels.

Generalization to non-perfect systems. In any real world application, no experiment is carried out under perfect system conditions. This motivates the idea of disturbing the numerical forward model used for the specimen topography to simulate a real measurement scenario by adding wavefront manipulators to the virtual TWI (cf. Table 4.1). Now, the generalization capacity of the deep neural network can be tested. The trained model needs to cope with data coming from a non-perfect system after having trained on a perfect simulation environment in the first stage. Using a state-of-the-art calibration, the trained model can be applied to a calibrated system environment (cf. Figure 4.2).

Here, the focus lies on the two-radii freeform surface as the design topography. Thirty difference topographies are randomly chosen from the former test set, i.e. not included in the U-Net training. They have a total root mean squared deviation of 545 nm and are ranged from 296 nm to 6.1 μm in their absolute maximal deviation from peak to valley. The results are shown in Table 4.2, where the same trained neural network is used for differently produced inputs. The RMSE of the network predictions is 30 nm on the perfect TWI system. This increases to 538 nm after having disturbed the virtual TWI. The trained model is incapable of predicting properly. However, the error can be reduced to 67 nm by using a calibrated forward pass to produce the input data. Hence, the proposed method can also generalize to

non-perfect systems. Of course, the accuracy of the hybrid method depends on the quality of the calibration, which will be discussed in more detail in Section 4.2.

Table 4.2: RMSE and median of the absolute errors for the predictions of the same trained U-Net using different inputs. The results for the perfect system also used to generate the training data are shown in the first column, for the non-perfect system without calibration in the second column, and for the proposed hybrid method in the third column.

	perfect system	**non-perfect system**	**calibrated system**
RMSE	30 nm	538 nm	67 nm
Median	16 nm	298 nm	33 nm

4.1.4 Discussion

The obtained results are promising and suggest that deep learning can be successfully applied in the context of computational optical form measurements in answer to Objective 2, i.e. to develop a deep learning approach for the form measurement of optical aspheres and freeform surfaces (cf. Section 1.2). We analyzed various specimens based on two different design topographies. The findings show that it is possible to generate a database using the numerical forward model given by the virtual TWI to successfully train a U-Net to solve the inverse reconstruction problem. An alternative parameterization could be one that predicts Zernike coefficients instead of the direct height values of the topography. However, here the interest lies in solving a direct image-to-image regression task and not to limit the prediction to a certain polynomial degree.

First experiments were carried out for in-distribution data, where the disjoint training and test data are generated under perfect system conditions, i.e. using the same forward model to compute the OPDs for the design and specimen topographies, respectively. In a second step, we corrupted the numerical forward model for the test data to simulate a more realistic experimental environment. Simply applying the trained model on the new dataset results in a meaningless prediction with high prediction error. A remedy is proposed by including a state-of-the-art calibration method to refine the numerical forward model, so that it is less corrupted. Using the data from the calibrated system as an input to the neural network, the developed hybrid approach is able to generalize well.

The presented results are based on simulated data only and they constitute a proof-of-principle study rather than a final method that is ready for application. There is still room for improvement considering the measurement accuracies for the aspherical design topography having four input channels to the neural network. Testing the approach on real measurements, incorporating an uncertainty quantification, and comparing the results to a state-of-the art method are next steps. Nevertheless, these initial results are encouraging and once trained, a neural network solves the inverse problem much faster than the currently applied state-of-the-art methods.

4.2 Incorporating an uncertainty quantification via deep ensembles

However[3], their black-box character and the resulting lack of trustworthiness are probably the most crucial shortcomings of deep learning approaches. Many examples exist that demonstrate unreasonable behavior of trained networks. For instance, the technique of layer-wise relevance propagation has revealed that an apparently well-trained image classification algorithm had adopted a “Clever Hans" decision strategy (Lapuschkin et al., 2019); the model had learned to classify horse images correctly by focusing on the bottom left corner of the image - there was an unnoticed tag remaining on horse images in the training dataset. Adversarial attacks can mislead trained networks into making unreasonable predictions by only slightly perturbing the input data (Akhtar and Mian, 2018). In 2018, an autonomous driving car crashed into a pedestrian in Arizona because the self-driving system did not classify her correctly (NTSB, 2019).

These examples demonstrate the importance of understanding the behavior of deep neural networks in order to ensure their trustworthiness. Much effort has been devoted to developing corresponding approaches as already discussed in Section 2.2. We focus on the uncertainty quantification of the prediction. Uncertainties are often classified as epistemic and aleatoric (Gal, 2016; Hüllermeier and Waegeman, 2021). Sources of uncertainty include imperfect training, unexpected shifts in the data, systematic errors and out-of-distribution data, to mention just a few (Ashukha et al., 2020; Kendall and Gal, 2017; Ovadia et al., 2019; Scalia et al., 2020). The most common approaches to uncertainty quantification include Bayesian neural networks (Kononenko, 1989; Yao et al., 2019), dropout based methods (Kingma et al., 2015; Gal and Ghahramani, 2016) and ensemble techniques (Dietterich, 2000; Lee et al., 2015; Lakshminarayanan et al., 2017). This thesis focuses on the latter, because ensemble learning scales well to higher dimensional data, performs best in recent uncertainty studies and is straightforward to implement (Caldeira and Nord, 2020; Gustafsson et al., 2020; Ovadia et al., 2019; Scalia et al., 2020).

The goal of this section is to incorporate a quantification of the uncertainty for the reconstructed optical surfaces based on the TWI. At the same time it is of interest to explore the potential of ensemble techniques for uncertainty quantification in deep learning in terms of the large-scale inverse problem known from computational optical form measurements. Uncertainty quantification is generally an important field in computational science and engineering (Dieck, 2007; Ghanem et al., 2017). Considering the TWI application, there exists previous work in which quantities have been identified that influence the uncertainty (Fortmeier et al., 2017). Some of these quantities have been analyzed via experimental measurements, Monte-Carlo simulations or the Jacobian used for the iterative topography updates (Fortmeier et al., 2013, 2014; Baer et al., 2014b; Harsch et al., 2019; Schindler, 2020).

The contribution of this section is twofold. First, we extend the previously proposed deep learning approach from Section 4.1 to incorporate an uncertainty quantification of its predictions. This is achieved through ensemble learning. In contrast to the previous section, the neural networks are trained on a calibrated dataset, which will be discussed in more detail.

[3]Most of this section is taken from Hoffmann et al. (2021b).

Second, we systematically insert an increasing out-of-distribution calibration error into the optical system and analyze its effect on the reliability of the developed uncertainty quantification. Furthermore, we investigate the influence of noise. High-dimensional uncertainty quantification for scalable deep learning techniques is less treated in literature than low-dimensional problems (Gustafsson et al., 2020), which makes the results of the chosen application useful for other machine learning applications as well.

The section is structured as follows. Subsection 4.2.1 introduces the data used for the study. The employed deep neural network, ensemble learning and the corresponding uncertainty quantification are introduced in Subsection 4.2.2. Results are then presented in Subsection 4.2.3 with a particular focus on the impact of systematic calibration errors on the uncertainty quantification. Finally, in Subsection 4.2.4, the potential benefit of deep learning for computational optical form measurements is discussed and possible future research motivated.

4.2.1 Data

The data generation is carried out as explained in Subsection 3.2.2. A dataset containing various topographies is generated through randomly drawn sets of Zernike coefficients. The Zernike polynomials parameterize the difference topographies ΔT, i.e. the deviation of the specimen from the known design topography. Then, the OPDs through the optical system are computed for the design topography and each generated specimen at a fixed position, respectively. Hence, for each specimen, a difference of OPDs ΔL is obtained from the difference between the OPDs $L = F(T)$ from the specimen T and the OPDs $L_0 = F(T_0)$ from the design topography T_0, where F is the employed numerical forward model (cf. virtual TWI, Chapter 3).

The asphere used as design topography is characterized in Subsection 3.2.2. Each sample in the generated dataset consists of a set of differences of OPDs ΔL and the difference topography ΔT between the specimen and the design topography. An example is shown in Figure 3.6. The difference of OPDs ΔL consists of four images because four disjoint masks are sequentially used to block different groups of the point light sources. In total, almost 40,000 (virtual) topographies are generated for training and about 2,000 topographies are generated for testing. The RMSD of the difference topographies in the generated test dataset is 564 nm and their median deviation is 473 nm. The peak to valley differences of the topographies of the test set range from 42 nm to 11.6 μm. Examples are shown in Figure 4.4 to illustrate the diversity of the test data. The exact amount of training and test data are not optimized. It was shown in the previous Section (cf. Figure 4.3b) that the number of available training data is crucial for the prediction capability. However, 40,000 training data are enough to achieve accurate predictions. More data would likely lead to even better results, but the goal is to systematically analyze the uncertainty quantification of ensemble learning. Also, the generated test set already represents a large diversity of possible difference topographies.

The numerical forward model of the optical system used to solve the inverse problem is never perfect when considering a real-world environment. State-of-the-art methods include a calibration step, where the numerical forward model is refined using the observed data for some test specimens whose topographies are known with high accuracy (Baer et al., 2014a;

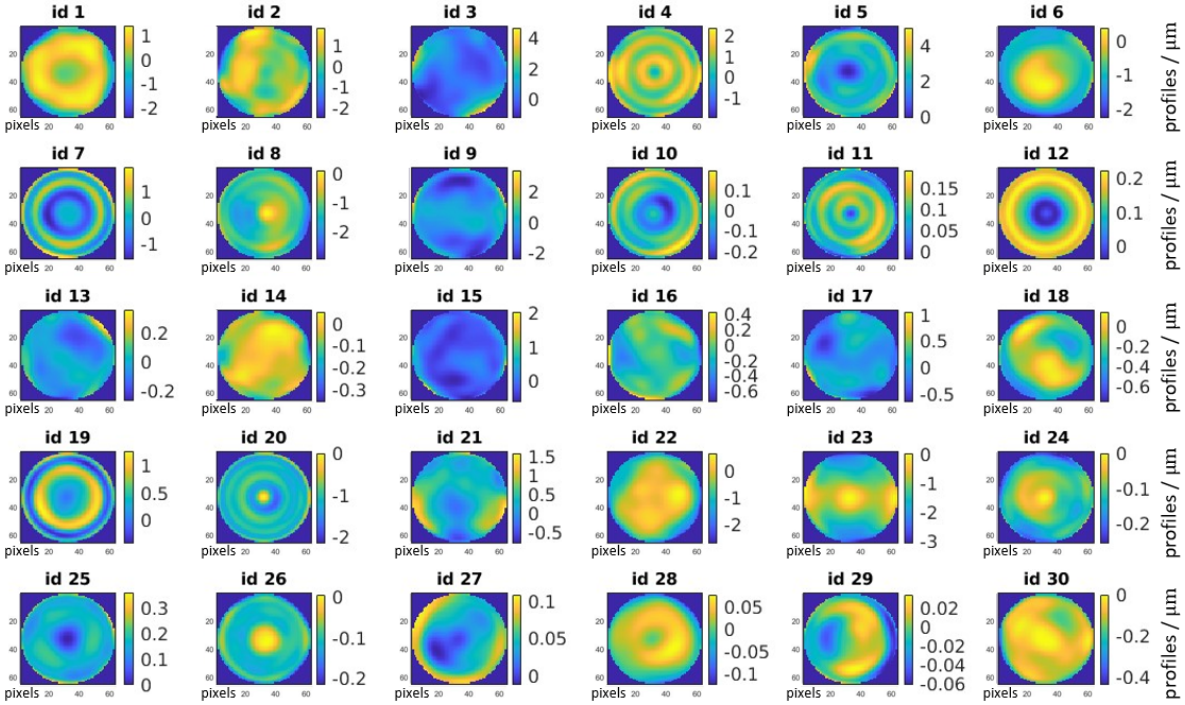

Figure 4.4: Some examples of generated test topography deviations ΔT are shown. The aperture has a radius of about 15 mm. The plots are taken from Hoffmann et al. (2021b).

Fortmeier et al., 2022). The calibration is realized by adding two wavefront manipulators to the virtual TWI (cf. Subsection 3.1.3). In this work, we adopt the same strategy to cope with non-perfect optical systems, and analyze the impact on the deep neural network prediction.

In the proof-of-principle study from the last section (4.1.1), the training data were generated without including wavefront manipulators to the model of the optical system, and the simulated data were constructed under the assumption of a perfect optical system. Now, systematic investigations on the impact of errors in the numerical forward model are carried out. For this purpose, all applied numerical forward models use activated non-zero wavefront manipulators to modify the light beams phenomenologically. The training data are determined by simulating data for a *perfectly calibrated* optical system, i.e. the virtual TWI (including the wavefront manipulators) is applied for computing the simulated data for the specimen and the design topography. In contrast, the test data are generated using a non-perfect optical system by deliberately adding calibration errors to the optical system. These errors represent errors caused by an imperfect calibration and will be termed *calibration errors* in the following. To construct test data containing such calibration errors, the wavefront manipulators of the perfectly calibrated optical system are systematically modified, which alters the beam path through the optical system.

Investigating the predictive ability of the trained deep neural network for test data corrupted by calibration errors allows the generalizability of the trained model to be explored for a more realistic scenario. From the point of view of machine learning, the test set constructed under calibration errors is an out-of-distribution test set, as no calibration errors are contained in the training set. A particular interest lies in the behavior of the calculated uncertainties and in the extent to which they reflect the errors in the reconstructed topographies for out-of-distribution

data. In total, the introduced calibration error affects the differences of OPDs ΔL up to a root mean squared deviation of 219 nm on the test dataset.

Noisy test data are generated by adding Gaussian noise to the input of the existing test data. Note that the training set is fixed and does not adapt to errors introduced into the test set.

4.2.2 Approach

General. The nonlinear inverse problem consists of finding the topography of the specimen such that the modeled OPDs best fit the observed ones. For a fixed design topography, the goal is to measure the deviation of the specimen from its design given the OPDs computed from the observed CCD intensities. The inverse problem is treated as a regression problem, analogue to the previous proof-of-principle study with a four channel input, i.e. $f_\theta : \mathbb{R}^{C\times M\times M} \to \mathbb{R}^{M\times M},\ \Delta L \mapsto \Delta T$, $\theta \in \Theta$, $C = 4$, $M = 64$ (cf. regression problem in Subsection 4.1.2). The parameterization f_ϕ is in general defined through the chosen neural network architecture and its concrete structure as for example the used activation functions, the actual depth of the network or the individual layers. All trainable network parameters are real numbers.

Similar to the previous study, we choose the U-Net (Ronneberger et al., 2015) as the network architecture, because it achieved accurate results in the proof-of-principle study and in various imaging tasks (Esser and Sutter, 2018; Barbastathis et al., 2019; Işıl et al., 2019). An example of a U-Net structure is shown in Figure 4.1. The input is processed through several layers, most of which are convolution layers, and results in the predicted output from left to right. The network has a bottleneck structure, which means that the dimension of the image data is reduced after each bundle of layers until it attains its minimal dimension in the center and increases afterwards in the same way. Furthermore, the arrows indicate skip connections between the bundles of layers of the same dimension, which means that the output of the last layer with the same dimension is depth concatenated to the input after the dimensional increase. The rectified linear unit (ReLU) activation function (Nair and Hinton, 2010) is chosen for the U-Net because it is the state-of-the-art for deep neural networks. We normalize all data pixelwise by subtracting the mean and dividing by the standard deviation of the training data. The model output is rescaled accordingly.

Deep ensembles. The landscape of deep neural networks and the associated question of finding global optima under certain assumptions are topics of ongoing research (Sun et al., 2020). Training multiple networks and making the prediction a decision of the constructed network ensemble (Hansen and Salamon, 1990) is a straightforward solution to deal with the problem of getting stuck in local minima. Deep ensembles have been proposed in Lakshminarayanan et al. (2017) for uncertainty quantification because of their predictive variety. The random initialization, along with the random shuffling of data points during training, is considered to induce sufficient diversity to the neural network ensemble. Dropout layers (Baldi and Sadowski, 2013) are also included in this work during training to encourage diversity. In contrast to Lakshminarayanan et al. (2017), we focus on the epistemic part of the uncertainty, since the model uncertainty is the relevant part of the uncertainty when inferring an underlying

regression function, and not on predicting an extra variance per output neuron to model the aleatoric uncertainty of the observed data.

An introduction to the deep ensembles (Lakshminarayanan et al., 2017) is provided in Subsection 2.2.4. A lot of effort goes into finding an appropriate neural network architecture, especially when considering high-dimensional regression tasks. For example, simpler convolutional neural networks or autoencoders without skip-connections did not achieve accurate results in this application in contrast to the U-Net. Consider an ensemble with K ensembles members. The *ensemble prediction* is defined as the average over all predictions of the ensemble members for an input x, i.e.

$$f_{DE}(x) := \frac{1}{K} \sum_{j=1}^{K} f_{\theta_j}(x) \ , \theta_j \in \Theta, \, j = 1, \ldots, K. \tag{4.6}$$

Similarly, the *ensemble uncertainty* is defined as the standard deviation over the individual predictions:

$$u(f_{DE}(x)) := \left(\frac{1}{K} \sum_{j=1}^{K} \left(f_{\theta_j}(x) - f_{DE}(x) \right)^{.2} \right)^{.\frac{1}{2}} . \tag{4.7}$$

Here, the tuple (x, y) equals the differences of OPDs and the corresponding difference topography, i.e. $(\Delta L, \Delta T)$. Hence, the ensemble prediction (4.6) and the ensemble uncertainty (4.7) are matrices. In (4.7), $^{.2}$ and $^{.\frac{1}{2}}$ indicate the elementwise square and square root, respectively. This definition of ensemble uncertainty is in line with the uncertainty definition from Lakshminarayanan et al. (2017) when omitting the aleatoric part. If the ensemble uncertainty is considered for an entire topography, the scalar valued *topography uncertainty* is referred to as the root mean square of the pixel values in $u(f_{DE}(x))$ of the predicted topography, i.e.

$$\sqrt{\frac{1}{d_y} \sum_{i=1}^{d_y} u(f_{DE}(x))_i^2}, \tag{4.8}$$

where d_y is the number of topography pixels.

Training. Previous work indicates that few ensemble members suffice to attain good results (Ashukha et al., 2020) and training time gets expensive with an increasing number of networks. Here, the ensemble is chosen to contain eight networks, i.e. $K = 8$. This number is probably not optimal, but is a good trade-off between the computational capacity of the ensemble training and the accuracy of the ensemble prediction and its uncertainty quantification, since the prediction capability and the uncertainty estimation of the ensemble have already converged, as shown in Table 4.3 in the results section. We choose an ensemble of eight U-Nets to be trained with the mean squared error loss, because the mean squared error loss is more robust based on previous experience than for example the negative log likelihood loss function. The U-Nets are independently trained for the ensemble using the mean squared error as loss function together with the Adam optimizer (Kingma and Ba, 2015) and an L_2 regularization with regularization factor 0.002 (cf. Subsection 2.2.2). Note that by minimizing the chosen loss function, i.e. $\frac{1}{N'} \sum_{i=1}^{N'} \|f_{\theta'}(x_i) - y_i\|_2^2 + \lambda \|\theta'\|_2^2$, where λ is the regularization parameter, N' is

the current batch size and θ' the current network parameters, equation (4.4) is approximately optimized.

Each network consists of 69 layers, including five max pooling layers and transposed convolution layers, respectively. The initial learning rate equals $5 * 10^{-5}$, with a learning rate drop factor of 0.75 every fourth epoch. In total, each network trains for 25 epochs with a mini batch size of 64, and the data are randomly shuffled every epoch. These parameters were selected after an initial grid search and tests with other choices of parameters. Similarly to the exact number of ensemble members and amount of training data, the selected parameters could be further optimized. However, we did not do so as we found that they serve our purpose sufficiently well.

To sum up, an ensemble of $K = 8$ deep neural networks $(f_{\theta_i})_i$, $\theta_i \in \Theta$, $i = 1, \ldots, K$, is trained that all have the same U-Net architecture and are independently trained on the same training data. Diversity is induced through random initialization, random data shuffling per epoch and inclusion of dropout layers during the training procedure. Training takes around 45 min for one network on a single GPU (Tesla V100-NVLINK on an Intel Xeon Gold 6226). The trained deep neural network ensemble takes about 300 milliseconds to predict 30 difference topographies (on a single GPU, Tesla V100-NVLINK on an Intel Xeon Gold 6226). In Fortmeier et al. (2014), the TWI evaluation procedure takes about 15 seconds for one topography reconstruction considering 13,350 rays, and assuming five iterative reconstruction steps (on an Intel(R), Core(TM), i7-2600, single CPU). The implementation is realized in MATLAB® (version R2020b) using the Deep Learning Toolbox 14.1 (The MathWorks, 2018).

4.2.3 Results

In the following, the obtained results are presented when considering a test set constructed using a perfectly calibrated system as well as test data produced by an optical system containing calibration errors. In addition, results are shown when the input data from the test set are corrupted by noise. Recall that the training data are constructed by simulating a perfectly calibrated optical system. The test set constructed after introducing calibration errors into the numerical forward model used to generate the data can be seen as an out-of-distribution test set and explores the generalizability of the trained deep learning approach. We focus on analyzing the uncertainty quantification produced by the neural network ensemble. As mentioned above, the uncertainty is primarily an estimation of the epistemic uncertainty (cf. Subsection 2.2.3), as the quality of the inferred regression function is of interest. Nonetheless, the behavior of the ensemble uncertainty is analyzed on out-of-distribution test data by means of the systematically introduced calibration errors, and also through analyzing test data whose input is corrupted by white noise.

Perfectly calibrated system. First, we evaluate the performance of the trained neural network ensemble on the test dataset generated by the same optical system as the training data, i.e. a perfectly calibrated system. The average root mean squared error (RMSE) equals 77 nm. Some results of the ensemble prediction are shown in Figure 4.5a. Three example topographies with greatly varying height sizes are reconstructed and the difference between the ground truth and the prediction is given. The neural network ensemble has no difficulties in reconstructing

the different topographies. The main error occurs at the edge of the topographies. This is not surprising, since the input data is more error-prone at the topography edges. There exists almost no redundant information since the patches rarely overlap at the edges, and some rays even leave the optical system without returning to the CCD. It is well-known that outliers have a high impact on the L_2 measure. Therefore, the median error (i.e. the median of the absolute errors) of an image is a more stable measure than the RMSE to capture the total predictive capacity of the network ensemble. Its average over the entire test set equals 20 nm.

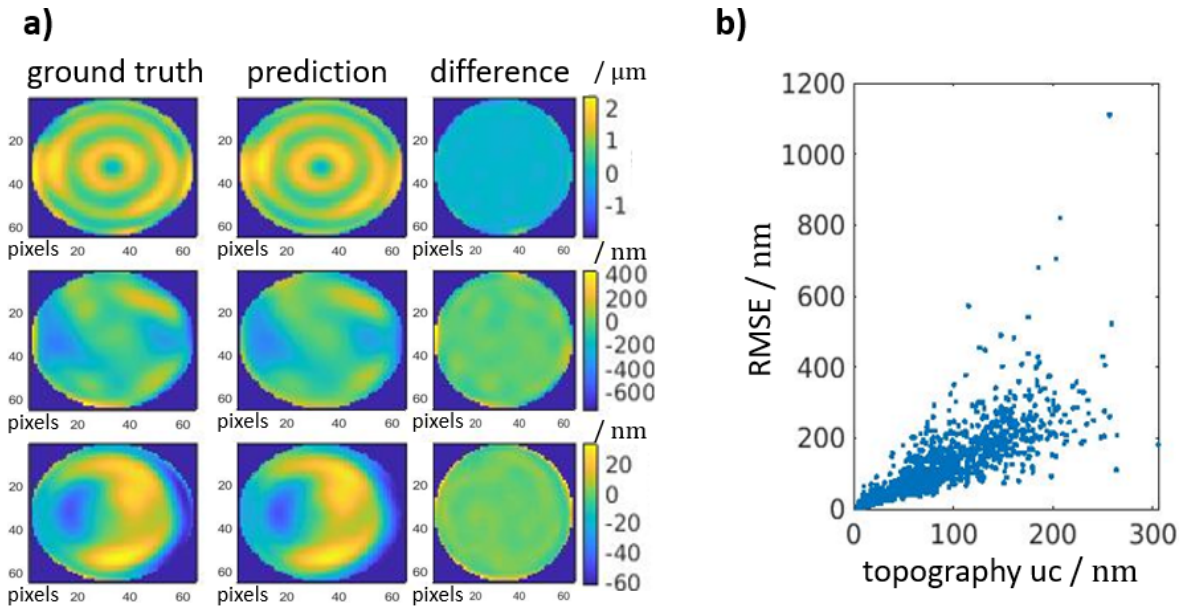

Figure 4.5: a) The ground truth, the ensemble prediction and their difference are shown for three example topographies with median errors of 44.5 nm, 17.2 nm and 1.5 nm, respectively. **b)** The RMSE (cf. equation (4.5)) of the ensemble predictions of the topographies are plotted against their corresponding topography uncertainties (cf. equation (4.8)). Both figures are taken from Hoffmann et al. (2021b).

The relationship between the topography uncertainty and the RMSE of the topography predictions by the ensemble is shown for the test dataset in Figure 4.5b. The uncertainty grows as the RMSE increases around a straight line with slope one. A more detailed analysis of the uncertainty estimation is given in Figure 4.6a, where the profiles of some test topographies are plotted along with the profiles of the ensemble predictions and the estimated *uncertainty tubes*, i.e. the profiles of ΔT, $f_{DE}(\Delta L)$ and $f_{DE}(\Delta L) \pm 1.96u(f_{DE}(\Delta L))$. The ground truth (in red) rarely leaves the uncertainty tube (in blue) and at the same time, the uncertainty tube is not too wide. The uncertainty tube is, in general, widest at the borders of the topographies and smallest at their centers. This behavior is in accordance with the corresponding sizes of the RMSE. Some further examples are presented in Appendix A.2, Figure A.2.

The factor 1.96 for the uncertainty tube equals the 0.025 and 0.975 quantiles of the standard normal distribution at -1.96 and 1.96, repsectively. This choice implies that, if the errors are normally distributed around the prediction of the ensemble with a standard deviation equal to $u(f_{DE}(x))$, then the uncertainty tube will encompass the difference between the predictions and the ground truth in 95% of the cases. The assumption of normally distributed errors was found to be justified in this application.

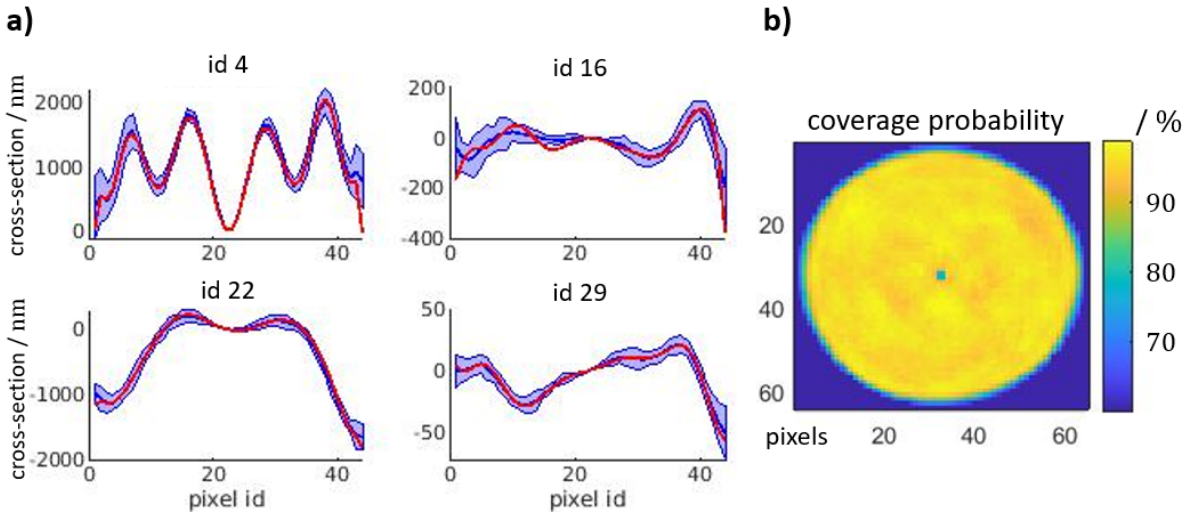

Figure 4.6: a) The profiles of four example topographies (cf. Figure 4.4) are plotted in red, along with the profile cross-sections of the ensemble predictions and the estimated uncertainty tubes in blue. **b)** The uncertainty estimation covers the prediction error well if the coverage probability is close to one. Both figures are taken from Hoffmann et al. (2021b).

Figure 4.6b shows the coverage probability of the uncertainty estimation. For each pixel of the topographies, the corresponding *coverage probability* is estimated as follows:

$$cp_i = \frac{1}{|D_{test}|} \sum_{(x,y)\in D_{test}} g\left(f_{DE}(x)_i, y_i\right), \quad g(\hat{y}_i, y_i) := \begin{cases} 1, & \text{if } \frac{|y_i - \hat{y}_i|}{1.96 u(\hat{y})_i} \leq 1 \\ 0, & \text{otherwise} \end{cases}, \tag{4.9}$$

where $|D_{test}|$ is the number of samples in the test data, y_i is the ground truth topography height at pixel $i = 1, \ldots, d_y$, $\hat{y}_i$ is the corresponding predicted topography height, and $u(\hat{y})_i$ the pixelwise uncertainty estimate as defined in (4.7). The coverage probability indicates how likely it is that the ground truth is found around the prediction within the chosen uncertainty tube. Figure 4.6b shows the pixelwise coverage probabilities, which indicate that the calculated uncertainties characterize the size of the errors of the predictions well. The *mean coverage probability* can be defined as the average over the pixelwise coverage probabilities, i.e.

$$\frac{1}{d_y} \sum_{i=1}^{d_y} cp_i. \tag{4.10}$$

The observed mean coverage probability equals 94%, which fits the intended 95% coverage probability well.

Table 4.3 shows the results in dependence of the ensemble size. Altogether, the neural network ensemble makes accurate predictions and provides a trustworthy uncertainty estimate, not only per image, but also pixelwise, for the perfectly calibrated optical system.

Systematically introduced calibration error. For the results presented above, training and test data were generated by means of the same perfectly calibrated optical system. However, the network ensemble should also make trustworthy predictions on out-of-distribution data. Indeed, in any real measurement scenario, there will remain a calibration error. Therefore, we analyze the quality of the ensemble prediction and its uncertainty quantification under the

Table 4.3: The table shows how the results change with the number of neural networks in the ensemble in terms of the prediction capability (second row) and how good the estimated uncertainty covers the prediction errors (third row). The results are produced on the perfectly calibrated test dataset without any additional noise.

number of ensemble members	1	2	3	4	5	6	7	8	25
median absolute error in nm	35	29	25	24	23	22	21	20	19
mean coverage probability in %	0	60	77	84	88	91	93	94	96

influence of a systematically introduced, growing calibration error. To this end, the optical system generating the test data is increasingly disturbed from the optical system used to produce the training data. The chosen topographies in the test set remain the same.

A first overview of the results is summarized in Table 4.4. The calibration error is induced stepwise from zero to one hundred per cent, cf. first row. The second row shows the actual impact of the corrupted numerical forward model, used to simulate the data from the specimens, on the resulting OPDs, which are used as input data by the trained neural networks in order to predict the sought topography. Then, the RMSE of the ensemble prediction is given in the fourth row, which achieves consistently better results than the prediction of a single trained network (third row). A more robust measure of the ensemble prediction is the median error in the fifth row, because it is more stable against outliers at the edges of the topographies. The sixth row displays the mean topography uncertainties. Finally, the mean coverage probabilities are calculated in the last row (cf. equation (4.10)).

Table 4.4: The influence of the introduced calibration error is analyzed on the test dataset. The rows contain the following values: the percentage of induced calibration error, its impact on the input data expressed as the root mean squared error, the averaged root mean squared error over the single network predictions, the root mean squared error of the ensemble prediction, the median absolute error of the ensemble prediction, the topography uncertainty and the mean coverage probability.

calibration error (in %)	0	10	20	30	40	50	60	70	80	90	100
flawed input (in nm)	0	28	55	75	96	116	137	156	176	198	219
rmse single net (in nm)	95	137	178	229	280	331	383	435	484	538	596
rmse ensemble (in nm)	77	104	145	194	241	288	336	384	429	479	536
median error (in nm)	20	29	42	55	68	82	95	110	125	139	154
topography uc (in nm)	55	90	108	131	155	178	201	225	246	267	285
total cp (in %)	94	94	84	78	75	73	71	70	69	68	67

Analogous to Figure 4.6a, the profiles of the four test topographies are plotted (in red) in Figure 4.7a, along with their ensemble predictions and their estimated uncertainty tubes (in blue) for the full calibration error. In the first column, the ground truth topographies mostly are within the range defined by the predicted topographies and their calculated uncertainties. Furthermore, the ensemble predictions recognize the basic shapes of the sought topographies, except for their edges. In contrast, the topographies from the second column are not recognized well. Instead, the ensemble predictions resemble one another and predict a topography with much stronger peak to valley variability. Considering the ground truth topographies (cf. Figure 4.4), Figure A.3 shows that the impact of the calibration error dominates the ensemble prediction for small topographies, while it has a smaller impact on larger difference topographies. The stepwise change in the ensemble prediction and its uncertainty estimation for the increasing calibration error are shown in Appendix A.2, Figures A.4 and A.5.

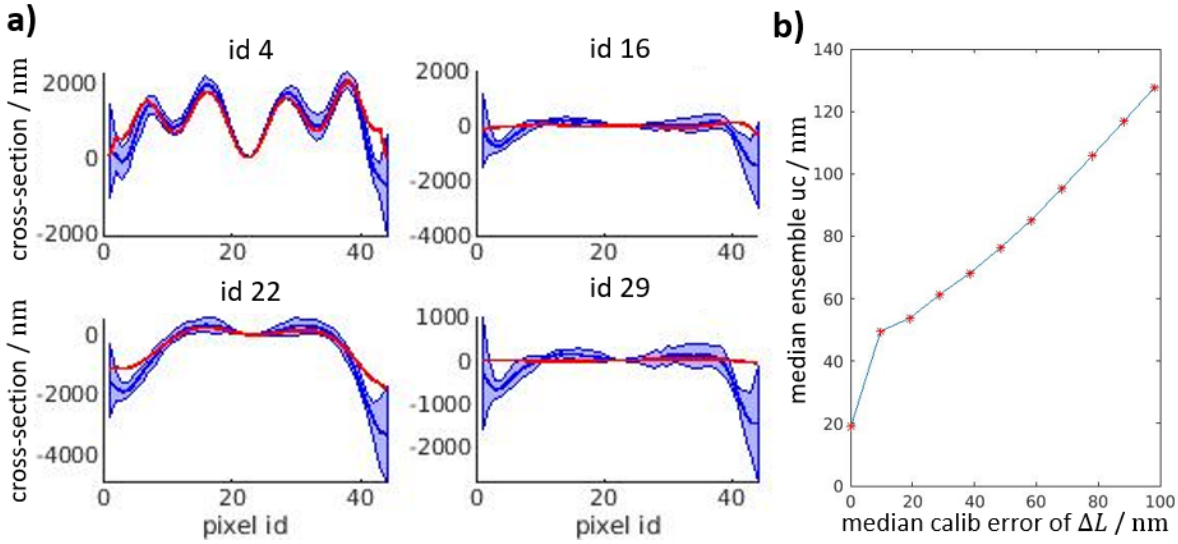

Figure 4.7: a) The profile cross-sections of four example topographies (cf. Figures 4.4 and 4.6a) are plotted (red), along with the ensemble predictions and the estimated uncertainty tubes (blue) under full calibration error. **b)** The median of the ensemble uncertainty on the test dataset is plotted against the median of the absolute calibration error on the input data ΔL. Both figures are taken from Hoffmann et al. (2021b).

The dependency between the ensemble uncertainty and the increasing calibration error is plotted in Figure 4.7b. The ensemble uncertainty grows monotonically with the growing calibration error with a slope of almost one, which is a desirable behavior for trustworthy predictions. Figure 4.7b shows the results over the entire test dataset, while Figure 4.7a presents the pixelwise results. In contrast, Figure 4.8 shows the obtained results on the image level, where the RMSE of the ensemble prediction is plotted against the topography uncertainty for the different degrees of induced calibration error. Again, the estimated uncertainty correlates well with the prediction error (as well as with the calibration error), even though the uncertainties are slightly underrated for very large calibration errors.

Finally, the coverage and mean coverage probabilities (cf. equations (4.9) and (4.10)) are shown in Figure 4.9 in relation to the growing calibration error. Although the coverage probability slowly decreases, the mean coverage probability stays at 94% after having induced 10% of the calibration error and still correctly covers two thirds of the pixels for the maximal

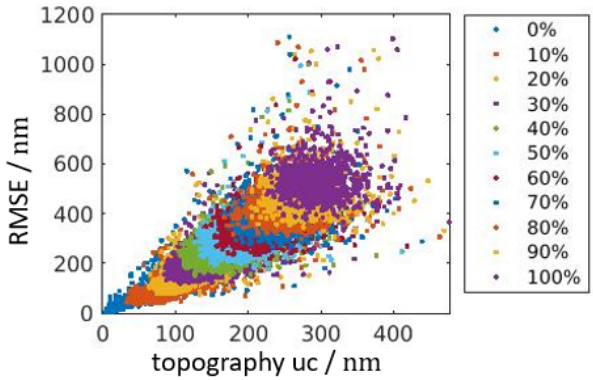

Figure 4.8: The RMSE of the ensemble prediction is plotted against the topography uncertainty. Each color represents a different amount of the systematically introduced calibration error. The figure is taken from Hoffmann et al. (2021b).

induced corruption of the input data in the test set. Furthermore, not the center but the topography edges are less well covered with increasing calibration error.

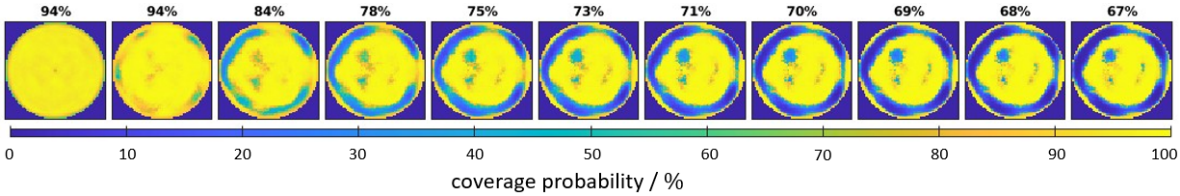

Figure 4.9: The coverage probabilities per pixel are plotted together with their mean coverage probability on the test set for increasing calibration error (from left to right). The figure is taken from Hoffmann et al. (2021b).

In sum, the ensemble prediction and its uncertainty quantification are best (for the investigated test cases) when the optical system is perfectly calibrated and get worse as the quality of the calibration decreases. However, the uncertainty increases with a growing calibration error and appears to still reliably characterize the size of the errors in the predictions.

Noisy data. The results shown above examine how the systematically deviated numerical forward model of the optical system, that is used to simulate the test data, influences the neural network ensemble. Another source of error is noise in the data. Therefore, the ensemble prediction and its uncertainty estimation are analyzed in terms of using noisy input data. Again, only the test data are modified, while the trained ensemble stays fixed.

Random white noise is added pixelwise to the input data with a standard deviation of 10 nm and 50 nm, respectively. This is done for the test data generated by the perfectly calibrated optical system as well as for the test data produced by the optical system after corrupting them with an increasing calibration error. Figure 4.10a shows the previous results from Figure 4.8 plotted against the results of the deep learning model when additionally corrupting the input data with normally distributed noise with a standard deviation of 50 nm. The root mean squared error of the ensemble predictions is plotted together with the topography uncertainties for the stepwise increasing introduced calibration error. There is no sudden loss in the prediction capability or the reliability of the uncertainty estimation to be seen for noisy input data. Only for the perfectly calibrated case (dark blue) are the errors

slightly larger for the noisy input data. However, also the topography uncertainties are slightly larger in this case than the topography uncertainties if the input data is not disrupted with white noise, which is a desirable behaviour.

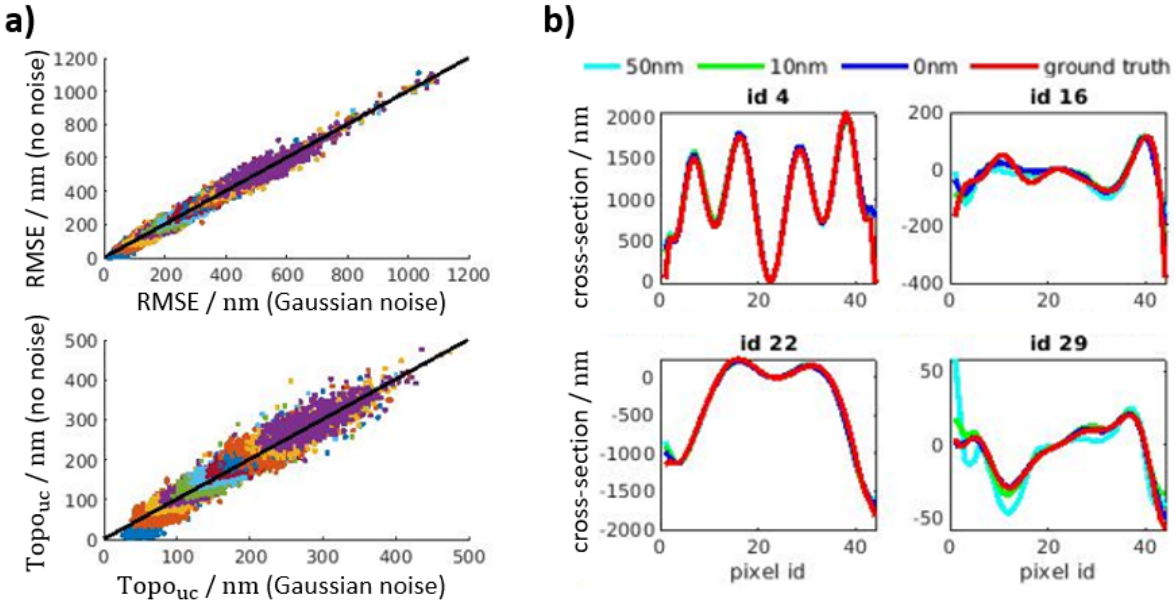

Figure 4.10: a) The root mean squared errors and topography uncertainties are plotted when feeding the network ensemble test data with and without additional Gaussian noise. The Gaussian noise has a standard deviation of 50 nm. The colors represent the amount of introduced calibration error in analogy to Figure 4.8. **b)** The ensemble predictions and their ground truth are plotted for four different test topographies under the influence of varying Gaussian noise (legend shows standard deviations) for the perfectly calibrated system. Both plots are taken from Hoffmann et al. (2021b).

A closer look at some example topographies is given in Figures 4.10b and 4.11, where the profiles are plotted. The former figure shows the ground truth together with the ensemble prediction for the perfectly calibrated data with and without noise. The noisy data have almost no impact on the ensemble prediction for the large topographies in the first row. In contrast, especially the noisy data with a standard deviation of 50 nm have a visible influence on the ensemble prediction of the smaller topography at the bottom right. Nonetheless, the main shape of the topography is still recognized.

Figure 4.11 shows the ground truth along with the ensemble prediction and estimated uncertainty for the noisy input data with a standard deviation of 50 nm. The first row a) shows the results for the perfectly calibrated optical system and can be compared to Figure 4.6a. Here, the uncertainties are much larger for the small topography (id 29) under the influence of noise, which is an appropriate behavior, as the prediction is also most strongly influenced by the noise for small topographies (cf. Figure 4.10b). The second row b) shows the same plots but for the maximal calibration error, which seems to dominate the noise; there is no visible difference between the plots in b) and those in Figure 4.7a.

In total, the neural network ensemble is not largely affected by noisy data, even though it did not encounter them during training. The prediction error might increase in some cases, but then the uncertainty estimate rises as well.

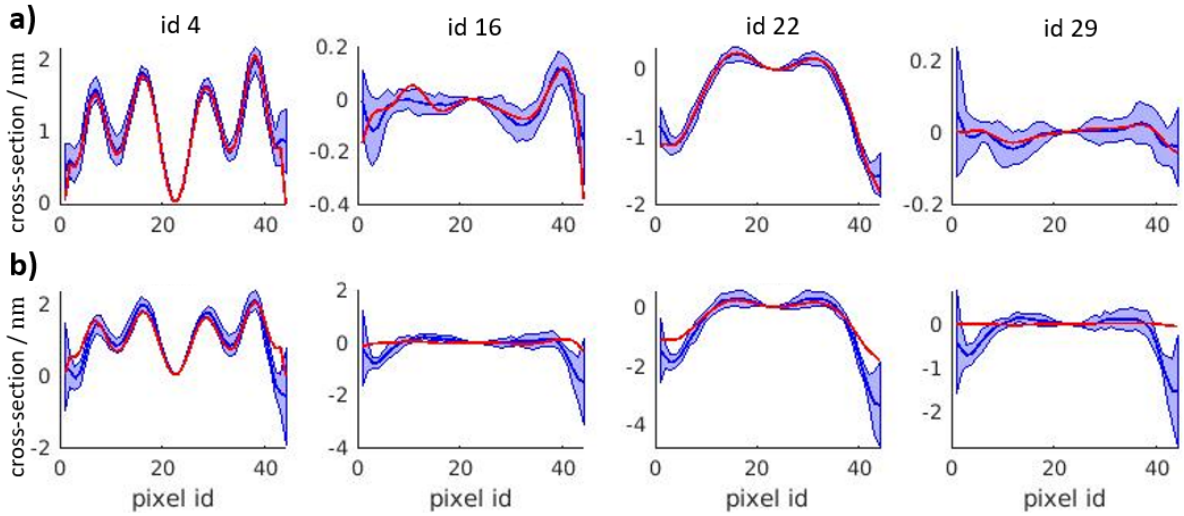

Figure 4.11: The profiles of the ensemble prediction and uncertainty tube are plotted (blue) with the underlying ground truth (red) for perturbed input data by white noise with a standard deviation of 50 nm for **a)** the perfectly calibrated system and **b)** the full calibration error. The plots are taken from Hoffmann et al. (2021b).

4.2.4 Discussion

In this section, our results showed that ensemble learning is a powerful tool for reliably solving the inverse problem of reconstructing topographies from given differences of OPDs. A major advantage of ensemble learning compared to other deep learning uncertainty methods is their scalability to high-dimensional data and the simplicity of implementation if a well suited architecture can be found. This is because the training procedure just has to be carried out multiple times without further adjustments. Other uncertainty methods result in a computationally more expensive approximate Bayesian inference (Neal, 1995) or require a modified loss function (Gal et al., 2017). Moreover, ensemble methods outperform other scalable state-of-the-art uncertainty methods in recent comparison studies (Gustafsson et al., 2020; Scalia et al., 2020).

The proposed ensemble method based on the deep ensembles, provides a sensible uncertainty quantification to its results in answer to Objective 3, i.e. to extend the developed deep learning approach to incorporate an uncertainty quantification to the estimated solution (cf. Section 1.2), which is shown on the pixel level as well as on the image level. This is true not only for a diverse test set that is disjoint from the training data, but also under the influence of different additional error sources. The neural network ensemble was successfully tested on out-of-distribution data realized by systematically deviating the computer modeled optical system used to simulate the test data, which can be associated with an increasing calibration error. The estimated uncertainty grows in relation to the introduced calibration error and the prediction error, respectively. Furthermore, the ensemble behavior is analyzed under the influence of noisy input data using different amounts of white noise. The noisy input has almost no effect on the prediction. An impact can be seen mainly for small topographies. However, the uncertainty estimation is able to reflect this behavior as well, meaning that because the different neural networks in the ensemble have different imperfections, the ensemble uncertainty covers the observed errors well.

From an application point of view, ensemble learning can be used to reliably solve the considered inverse topography reconstruction problem up to a certain accuracy faster than applied conventional methods after having trained the neural networks once for a specific design topography. Additionally, the neural network ensemble provides a consistent uncertainty quantification. Including other error sources such as positioning errors of the specimen or applying the proposed method to real data could be next steps. Deep ensembles could also be applied to validate the current status of a calibration using a fixed reference specimen, since the ensemble uncertainty should increase if the calibration worsens over time. In this way, the presented uncertainty estimate could be used to determine whether a new calibration of the system in necessary.

It is difficult to compare the developed deep learning approach to the conventional TWI, since there does not exist a unique TWI. Several research sites and companies use different implementations which are adapted by ongoing research (Mahr GmbH, 2021; Pruss et al., 2017; Fortmeier et al., 2022) and various hyperparameters influence the reconstruction. Our experience with the PTB implementation of the conventional TWI on simulated data suggests that the conventional method is much more accurate under perfect system conditions. However, the results from the two approaches converge rapidly towards each other when disturbing the optical system. In addition to providing a reliable uncertainty quantification of the predicted topography, the TWI application can benefit from faster calculation times using the deep learning approach once the model is trained.

From a machine learning perspective, this application exemplifies the ability of ensemble methods, and especially deep ensembles, to make trustworthy predictions and to provide an uncertainty quantification. The great potential of ensemble methods lies in their simple and straightforward implementation when solving high-dimensional problems. The results show that the uncertainty is mainly of epistemic nature. Future work could address the explicit incorporation of aleatoric uncertainty in the observed data as well. However, analyzing the effects of noisy input data is of more interest than of noisy target data, because an inverse problem is considered here. Also, establishing a high-dimensional benchmark dataset to test and compare scalable uncertainty methods could benefit the machine learning community.

4.3 Summary

A data-driven deep learning approach is developed and explored to solve the large-scale, nonlinear inverse problem for the accurate optical measurement of optical aspheres and freeform surfaces based on the TWI evaluation procedure. A proof-of-principal study shows that deep learning can be used to solve the inverse problem. For a fixed design topography and position, the deep neural network learns to correctly reconstruct the difference topographies from given differences of OPDs. Once the model is trained, it provides an instant output prediction to a new input. Ensemble learning is used to extend the scope of the developed method in order to provide an uncertainty quantification of the solution to the inverse problem predicted by the deep learning approach. The employed architecture of the deep neural networks is the U-Net and the ensemble learning method is based on the deep ensembles. Combining the deep learning model with a state-of-the-art calibration procedure, that refines

the numerical forward model, leads to an improved generalization performance and is an important step towards applying the proposed method on real word data.

The approach is developed and tested using virtual measurements with a known ground truth. Training is performed on a database generated in a perfect or at least perfectly calibrated optical system environment. By systematically inserting calibration errors to the numerical forward model as well as noisy data, the reliability of the developed approach and its uncertainty quantification is explored, and its generalization capacity is tested on out-of-distribution data. Results are encouraging and the proposed application exemplifies the ability of deep ensembles to make trustworthy predictions on the basis of high-dimensional data in a real-world context.

The first three objectives are discussed in this chapter, i.e. to generate a database for training and testing as well as to develop a deep learning approach with an incorporated uncertainty quantification of the model prediction for the optical form measurement of optical surfaces based on the TWI evaluation procedure. It can be concluded from our findings that computational optical form metrology can benefit from deep learning (in a virtual environment). Next steps include a more detailed comparison to the current state-of-the-art and the application on measurement data from a real specimen.

5

Measuring a real specimen

The[1] need for accurate measurement techniques increases with ongoing technological advances. A manufactured good can be fabricated only as accurately as it can be measured. Non-spherical optics, like aspheres or freeform surfaces, are indispensable for modern optical systems (Rolland et al., 2021). In order to track the state-of-the-art in the form measurement of such optical surfaces, interlaboratory comparison studies are performed to analyze the differences between the various high accuracy measuring systems, including tactile and optical instruments (Bergmans et al., 2015; Schachtschneider et al., 2018; Fortmeier et al., 2020; Fortmeier and Schulz, 2022). The tilted-wave interferometer (TWI) (Garbusi et al., 2008; Baer et al., 2014c) is one of the state-of-the-art interferometrical measurement techniques for optical form measurements of optical aspheres and freeform surfaces (cf. Section 2.1).

The evaluation procedure of the TWI leads to a high-dimensional inverse problem of reconstructing the specimen topography from measurement data. State-of-the-art TWI methods consist of three major steps for reconstructing the surface form: the prior refinement of the numerical forward model, i.e. the calibration step (Garbusi et al., 2008; Baer et al., 2014a; Mühlig et al., 2014); the reconstruction of the form deviation represented by Zernike polynomials (Fortmeier et al., 2014); and the reconstruction of the remaining high-frequency form deviations (Baer et al., 2013b; Fortmeier et al., 2022). Measurement uncertainties are not yet available for tilted-wave interferometry, but previous work addresses a number of relevant investigations (Fortmeier et al., 2014; Baer, 2016; Fortmeier et al., 2017; Schindler, 2020; Scholz et al., 2022).

This chapter tackles Objective 4, i.e. to apply and validate the developed deep learning approach to solve the inverse problem of the TWI on measurement data of a real specimen (cf. Section 1.2), and shows that the previously developed deep learning approach for computational optical form measurement can be applied to measurement data it has never seen before, even if it was trained on purely simulated data. The results obtained include a quantification of the uncertainty of the reconstructed topography and are shown to be reliable by comparing the deep learning approach to a state-of-the-art method for optical form measurements, namely

[1] Most of this chapter is taken from Hoffmann et al. (2021a).

the tilted-wave interferometer distributed by Mahr GmbH (MarOpto TWI 60) (Mahr GmbH, 2021). The chapter is structured as follows. First, we apply the proposed deep learning approach to measurement data from a real specimen in Section 5.1. Then, the results are compared to the state-of-the-art in Section 5.2. Finally, conclusions are drawn in Section 5.3.

5.1 Application to measurement data using deep ensembles

Chapter 4 introduced a deep learning approach to solve the inverse topography reconstruction problem of the TWI. The approach incorporates a quantification of the uncertainty of the reconstructed topography and achieves accurate results and reliable uncertainty estimates in a simulation-based environment. Now, the goal is to extend the application of the developed approach to measurement data.

A reliable deep learning model should generalize well to out-of-distribution data. This means that the uncertainty quantification should relate to the gap between the model prediction and its ground truth not only with respect to the data originating from the training data domain but also for data generated from outside. It was previously shown that the deep learning approach generalizes well for simulated out-of-distribution data when introducing a calibration error into the optical system used for the data generation (cf. Subsection 4.2.3). The application to measurement data is presented in the following. It includes additional challenges, such as the occurrence of misalignment errors or additional image artefacts of the measurement, and the ground truth is unknown.

Subsection 5.1.1 provides an overview of the data used in this chapter, i.e. the measurement data of a real aspherical specimen. Then, in Subsection 5.1.2, the deep learning approach is briefly recapitulated and extended for the application on measurement data. The results are presented in Subsection 5.1.3.

5.1.1 Data

The goal of this chapter is to apply the deep learning approach to measured data from a real specimen and to compare it to a state-of-the-art method. Analogous to the training and test data used in the last chapter, the specimen T considered here has the same asphere as underlying design topography T_0 (cf. Appendix A.1). The measured interferograms have a resolution of 2048×2048 pixels. Figure 5.1 shows the measured phase images resulting from the collected interferograms that are computed by a five-step phase shifting algorithm (Hariharan et al., 1987) and setting the background pixels to zero using the SimOptDevice (Schachtschneider et al., 2019). Remember that there are four channels representing the four disjoint masks used in the focal plane of the microlens array (cf. Subsection 2.1.3) to block the light coming from neighboring microlenses.

The measured OPDs L are calculated using the Goldstein unwrapper (Goldstein et al., 1988), while the OPDs L_0 corresponding to the design topography are simulated by a calibrated virtual TWI, i.e. the refined numerical forward model (cf. Chapter 3). The differences between the measured and the simulated OPDs are shown in Figure 5.2a. Note that the images are sampled down by selecting every 32nd pixel, because the input and output images of the deep neural networks have a resolution of 64×64 pixels. Less memory is needed to store the data

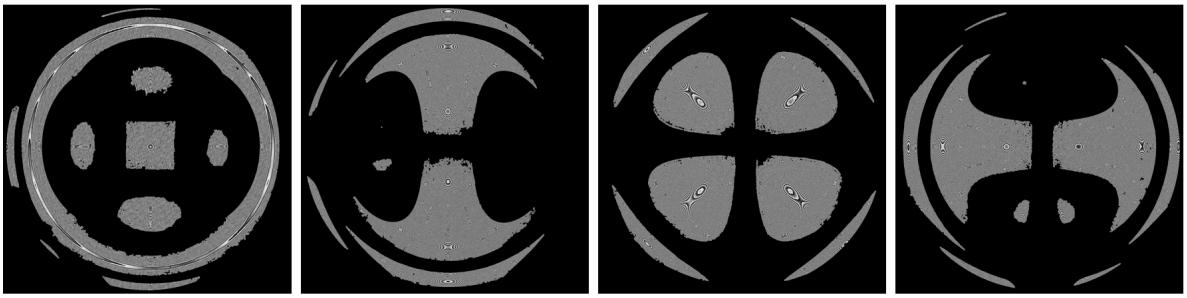

Figure 5.1: Measured phase data of the aspherical specimen after setting the background and invalid pixels to zero. Each patch represents a sub-interferogram.

and train the neural networks for lower resolutions. Furthermore, generating the training data is much faster because less rays need to be traced through the optical system. This resolution is sufficient for the purposes of this work, but it can be adapted as required in future work.

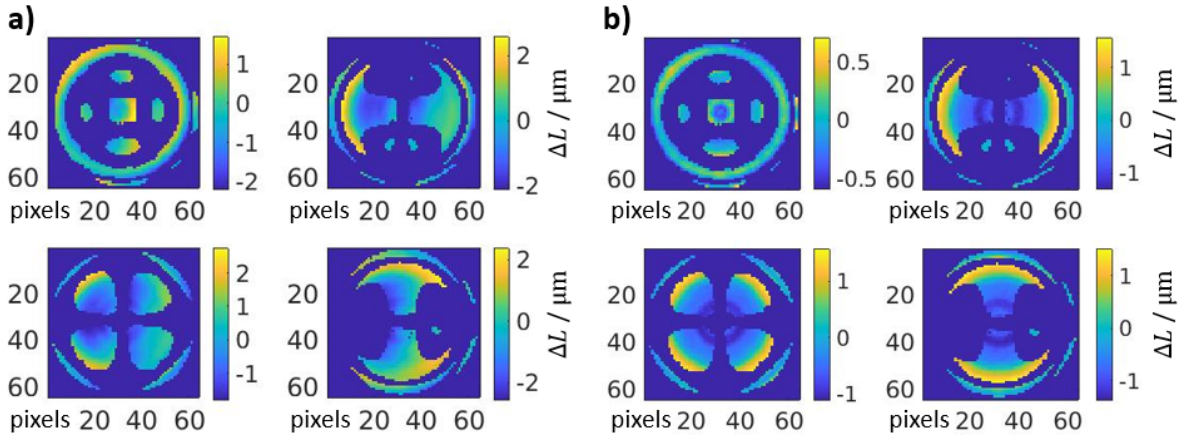

Figure 5.2: a) The down-sampled differences ΔL between the measured and simulated OPDs represent the four channels of input to the deep neural networks. **b)** Analogue to a), but including a pre-processing step where the positioning error between the specimen and the design topography is corrected prior to the simulation. Both figures are taken from Hoffmann et al. (2021a).

Positioning errors directly influence the measured OPDs and hence the topography reconstruction (Gronle et al., 2022). However, no misalignment was taken into account during the neural network training. Therefore, we introduce a prior misalignment correction in a pre-processing step (see below, Subsection 5.1.2). The input to the deep neural networks then comprises the difference between the measured OPDs, which are the same as before, and the simulated OPDs computed after correcting the misalignment of the topography. Now, the input data is more centered, as shown in Figure 5.2b.

5.1.2 Approach

Solving the inverse problem related to the TWI. The deep learning approach is applied to measured data from a real specimen based on the trained deep learning model from the previous chapter (cf. Subsection 4.2.2). A deep learning hybrid approach was proposed for the TWI surface reconstruction procedure combining a conventionally refined numerical forward model to generate the input data with a deep learning model solving the inverse problem related to TWI.

Figure 5.3 summarizes the deep learning hybrid approach. The OPDs L of a real specimen T are calculated from the interferograms measured by the physical TWI, cf. Figure 5.3a. The known design topography T_0 constitutes a first guess to the reconstruction problem and the refined numerical forward model F, i.e. the virtual TWI, yields the corresponding simulated OPDs L_0. For a fixed design topography, the deep learning model learns to predict a difference topography ΔT from the corresponding differences ΔL between the measured OPDs and the simulated OPDs, as shown in Figure 5.3b. The model is trained on purely simulated data, which means that all data for the design topography and for the specimens are simulated (cf. Subsection 3.2.2). No positioning errors are included during training. Also, the deep learning model is trained on perfectly calibrated data, i.e. the same numerical forward model is used for the simulated data of both the design topography and the specimens.

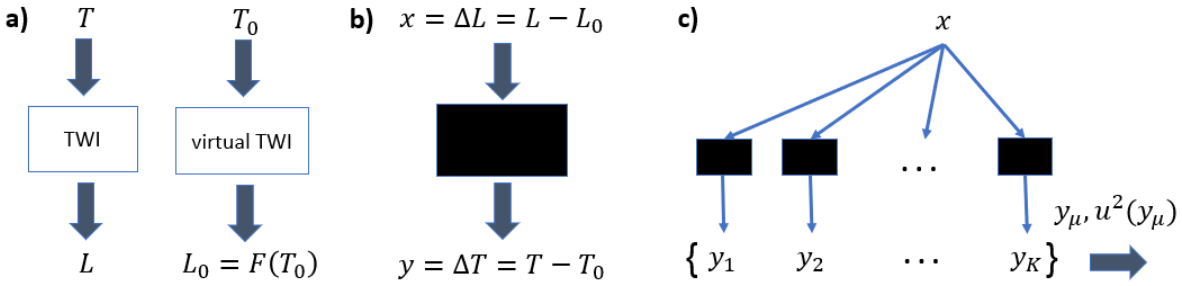

Figure 5.3: a) Based on the TWI, OPDs L calculated from the measured interferograms are provided for the specimen topography T. Using the (refined) numerical forward model F, the virtual TWI yields simulated OPDs L_0 for the design topography T_0. **b)** The deep neural network is illustrated as a black box. Its input is the difference between the OPDs from the specimen and the OPDs from the design topography. The output is the difference topography, i.e., the deviation of the specimen from its known design topography. **c)** Having independently trained multiple networks, the ensemble prediction ($y_\mu = \frac{1}{K}\sum_{i=1}^{K} y_i$) and uncertainty of the reconstructed topography ($u^2(y_\mu) = \frac{1}{K}\sum_{i=1}^{K}(y_\mu - y_i)^2$) are the mean and uncorrected sample variance of the individual member predictions. The operations are performed pixel-wise. The figures are partly taken from Hoffmann et al. (2021a).

The deep learning model is based on the deep ensembles (Lakshminarayanan et al., 2017) that represent a popular method for incorporating an uncertainty quantification into a deep neural network prediction and for making the prediction itself more robust. Multiple neural networks are independently trained instead of a single one. As sketched in Figure 5.3c, the ensemble prediction $y_\mu = f_{DE}(x)$ is the mean value of the individual network predictions and the (squared) uncertainty of the reconstructed topography $u^2(y_\mu)$ corresponds to the uncorrected sample variance. In (Lakshminarayanan et al., 2017), the uncorrected sample variance was introduced as a heuristic. It can, however, be justified from a Bayesian perspective as will be explained in Chapter 6. The approach chosen here uses an ensemble of eight deep neural networks, all having the same U-Net architecture (Ronneberger et al., 2015). The high-dimensional inverse problem of reconstructing the optical surface is treated as an image-to-image regression problem, and U-Nets have previously been successfully applied to various imaging tasks (Barbastathis et al., 2019).

Pre-processing. No misalignment was taken into account during the neural network training. The difference topography is therefore estimated a second time including a prior misalignment correction. The estimation of the specimen positioning error is realized by minimizing the

differences between the measured and simulated OPDs in a least-squares sense considering the parameters of the degrees of freedom of an asphere, i.e., the rotation angle parameters and the position in space, but excluding the position along the optical axis, which is measured with a distance measuring interferometer. Furthermore, the numerical forward model is refined in a conventional calibration step (Fortmeier et al., 2022) to reduce the corruptness of the non-perfect optical system that provides the input data related to the design topography to the already trained deep learning model (cf. Figure 4.2).

Post-processing. An additional post process is added to the deep learning hybrid approach when applied to measurement data to reduce the influence of misalignment errors to the reconstructed surface form. It modifies the individual ensemble member predictions before calculating the mean and uncorrected sample variance (cf. Figure 5.3c). The steps are summarized as follows:

1. Add the predicted difference topographies to the design topography to reconstruct the specimen.
2. Rotate and shift the point clouds of the reconstructed specimens into the design to align the coordinate systems of the design function and the reconstructed topography (Schachtschneider et al., 2018).
3. Subtract the design topography to deduce the resulting form deviation of the specimen from its design.
4. Calculate the average and uncorrected sample variance over the ensemble members.

The implementation is realized in MATLAB® (version R2021b) using the Deep Learning Toolbox 14.1 (The MathWorks, 2018) and the SimOptDevice (Stavridis et al., 2018; Schachtschneider et al., 2019; Fortmeier et al., 2022) developed at the PTB.

5.1.3 Results

In the following, the results are presented when applying the trained deep neural network ensemble on the differences of OPDs shown in Figure 5.2 to reconstruct the deviation of the specimen under test from its design. Figure 5.4 shows the output of the deep learning model. Coma (cf. Figure 3.2) can be detected in Figure 5.4a as a drop-shaped, tail-like distortion which can be caused by a positioning error, while the prediction in Figure 5.4b is more centered due to the prior misalignment correction. The horizontal and vertical cross sections of the predicted difference topographies are plotted together with the 95% symmetric credible intervals of the uncertainty of the reconstructed topography, i.e. $y_\mu \pm 1.96\sqrt{u^2(y_\mu)}$ (cf. Figure 5.3c).

Including the post-processing step for measurement data described above, the results of the reconstructed difference topography are presented in Figure 5.5. In addition to the post process, Zernike polynomial functions (up to the first 78 coefficients) are fitted to the resulting difference topographies to smooth the results. This is done in order to later compare the results to the results of the state-of-the-art measurement system that will be introduced in Section 5.2 focusing on the lower spatial frequencies of the form reconstruction. In Figure A.6

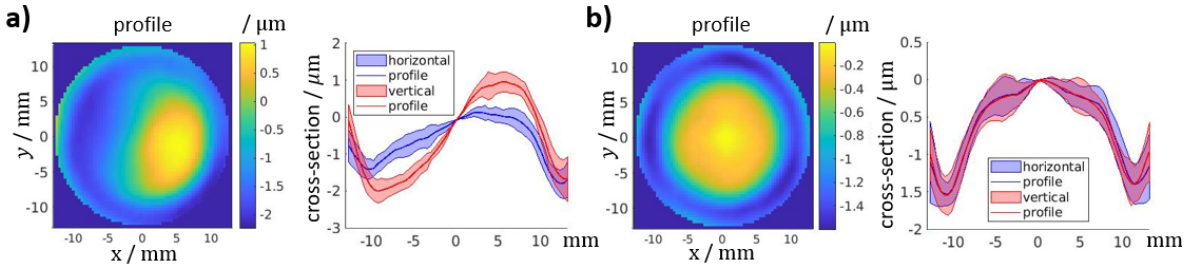

Figure 5.4: a) Output of the deep neural network ensemble for the input data from Figure 5.2a. The horizontal and vertical cross sections of the topography profile are plotted together with the 95% symmetric credible interval of the uncertainty of the reconstructed topography in the second plot. **b)** Analogous to a), but with the misalignment of the specimen corrected prior to the data simulation.

in Appendix A.2, the results are shown without additionally fitting Zernike polynomials. Note that the fitted Zernike polynomials could be employed to upsample the prediction to higher resolutions. The root mean deviation of the variance corresponding to the uncertainty of the reconstructed topography drops from 113 nm without prior misalignment correction to 77 nm when the prior correction of the topography position is included.

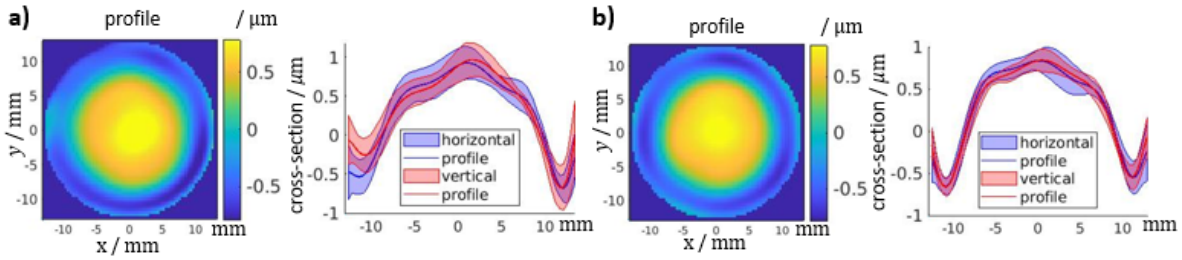

Figure 5.5: a) Reconstruction of the difference topography using the deep learning hybrid approach for the measured data of the aspherical specimen. The second plot shows the horizontal and vertical cross sections of the topography profile shown in the first plot, together with the 95% symmetric credible interval of the uncertainty of the reconstructed topography. **b)** Analogous to a), but with the misalignment of the specimen corrected prior to the data simulation. The figures are taken from Hoffmann et al. (2021a).

5.2 Comparison to a state-of-the-art method

The reconstruction results from the deep learning hybrid approach are compared to the results produced by a state-of-the-art method for optical form measurements, namely the tilted-wave interferometer distributed by Mahr GmbH (MarOpto TWI 60) (Mahr GmbH, 2021). The comparison procedure is similar to the one used in recent interlaboratory comparison studies (Schachtschneider et al., 2018; Fortmeier et al., 2020). Again, all processing steps are carried out using the simulation toolbox SimOptDevice (Schachtschneider et al., 2019).

5.2.1 Comparison procedure

In a first step, the point clouds of the reconstructed specimens need to be transformed into a format that is comparable. To this end, we restrict the data to the largest common aperture radius, and align the point clouds in the Cartesian coordinate system by fitting the reconstructed topographies into the design topography in a least-squares sense. This allows

the estimated difference topographies to be compared after subtracting the design from the reconstructed specimens. An additional step is included to subtract the best fit sphere (BFS) in a least-squares sense using the MATLAB implementation of the Levenberg-Marquardt algorithm. Comparing the results while neglecting the dominating spherical part can be of interest to show non-spherical deviations and because spherical contributions may arise from a remaining misalignment of the topographies along the optical axis (Fortmeier et al., 2016; Gronle et al., 2022). The steps of the comparison procedure are summarized as follows:

1. Restrict the data of the reconstructed specimens to the largest common aperture radius.
2. Rotate and shift point clouds of the reconstructed specimens into the design.
3. Subtract the design topography from the reconstructed specimens and repeat step 1.
4. Subtract the respective BFS from the reconstructed difference topographies.

The results are compared after steps 3 and 4. The reconstructed topography and its uncertainty predicted by the deep learning model are propagated by performing each step of the comparison procedure for the individual ensemble member predictions, and by taking the mean and uncorrected sample variance only afterwards. The achieved resolution of the deep learning hybrid approach (64×64) is much smaller than the resolution of the measured data on the camera (2048×2048). Also, no high-frequency parts were included during the neural network training, since the training topographies are generated using up to the first 136 Zernike polynomial functions (cf. Subsection 3.2.2). Therefore, Zernike polynomials are fitted to the reconstructed difference topographies up to the first 78 coefficients before plotting the results. The chosen polynomial degree could be higher or lower, but the chosen order ($n, m = 11$) (cf. Subsection 3.1.2) is sufficient for the purpose of this work. In this way, only lower spatial frequencies of the form reconstruction are compared while higher frequency structures are neglected. The presented results are produced by evaluating the Zernike polynomials on a fixed equidistant grid. For completeness, the results are shown without the smoothening Zernike polynomial fit and with a higher resolution in Figure A.6 in Appendix A.2.

5.2.2 Results

A summary of the results is displayed in Tables 5.1 and 5.2. The reconstructed difference topographies (the results obtained after step 3) are shown in Figure 5.5 for the deep learning hybrid approach with and without prior misalignment correction. For comparison, the reconstructed difference topography from the state-of-the-art method is depicted in Figure 5.6a. The vertical (left) and horizontal (right) cross sections of the reconstructed topographies are directly compared in Figures 5.7a and 5.7c, respectively, with and without the misalignment correction prior to applying the trained deep learning model. Most of the reconstructed surface from the state-of-the-art method (red) lies in the 95% symmetric credible interval of the deep learning results (blue). More precisely, 92% of the reconstructed form from the state-of-the-art method is covered by the described uncertainty of the reconstructed topography of the deep learning prediction when no prior misalignment correction is carried out, while 93% of the topography is covered when the prior correction of the topography position is included. The root mean squared deviation (RMSD) (cf. equation (4.5)) of the reconstructed difference

Table 5.1: The root mean squared deviation (RMSD) of the reconstructed difference topographies, before and after subtracting the best fit sphere (BFS), are presented for the state-of-the-art method as well as for the deep learning based hybrid approach without (hybrid method 1) and with (hybrid method 2) prior misalignment correction. Also, the radii of the best fit spheres are shown, where the mean is taken over the ensemble members for the hybrid methods. All uncertainties refer to the uncorrected standard deviation over the individual ensemble member predictions.

	state-of-the-art	**hybrid method 1**	**hybrid method 2**
RMSD (in nm)	489	494	476
uncertainty (in nm)	-	30	22
RMSD after subtracting the BFS (in nm)	239	192	211
uncertainty (in nm)	-	22	25
radius of BFS (in m)	59.7	54.7	58
uncertainty (in m)	-	4.4	4.5

Table 5.2: The first row shows the root mean squared error (RMSE) between the reconstructed difference topographies from the state-of-the-art method and the deep learning hybrid approach before and after subtracting the BFS. Method 1 refers to the deep learning hybrid approach without prior misalignment correction, while method 2 includes it. The second row presents the root mean variances of the ensemble member predictions and the third row shows how much of the reconstructed form from the state-of-the-art method is covered by the 95% symmetric credible interval of the deep learning results.

	before subtracting the BFS		**after subtracting the BFS**	
	method 1	method 2	method 1	method 2
RMSE (in nm)	135	81	133	82
root mean variance (in nm)	113	77	107	70
coverage (in %)	92	93	91	89

topography is 489 nm for the state-of-the-art method, 476 nm for the deep learning method with prior misalignment correction, and 494 nm without such correction. The RMSE between the two methods is 81 nm with, and 135 nm without the prior misalignment correction included in the deep learning approach.

Figures 5.6b-d show the results after subtracting the corresponding BFS (step 4). The result from the state-of-the-art form reconstruction method (Fig. 5.6b) has a RMSD of 239 nm. The subtracted BFS has a radius of 59.7 m. Next, the result from the deep learning hybrid approach without prior misalignment correction (Fig. 5.6c) has a RMSD of 192 nm, and a RMSE of 133 nm compared to the state-of-the-art method, which lies to 91% in the area covered by the uncertainty of the reconstructed topography. The subtracted BFS has a mean radius[2] of 54.7

[2]It is common for the radius of the BFS to be in the range of meters when considering the reconstructed difference topographies for optical aspheres and freeform surfaces (Schachtschneider et al., 2018; Fortmeier and Schulz, 2022).

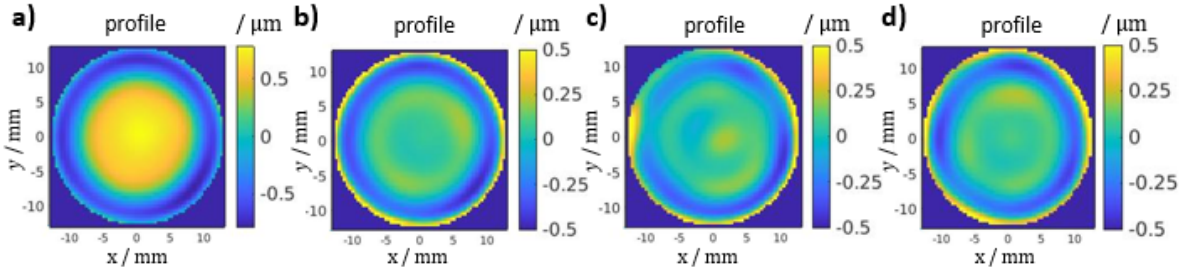

Figure 5.6: a) The reconstructed difference topography is plotted for the state-of-the-art method for optical form measurement. **b)** Result from a) after subtracting the best fit sphere. **c)** Result from Fig. 5.5a) after subtracting the best fit sphere. **d)** Result from Fig. 5.5b) after subtracting the best fit sphere. The figures are taken from Hoffmann et al. (2021a).

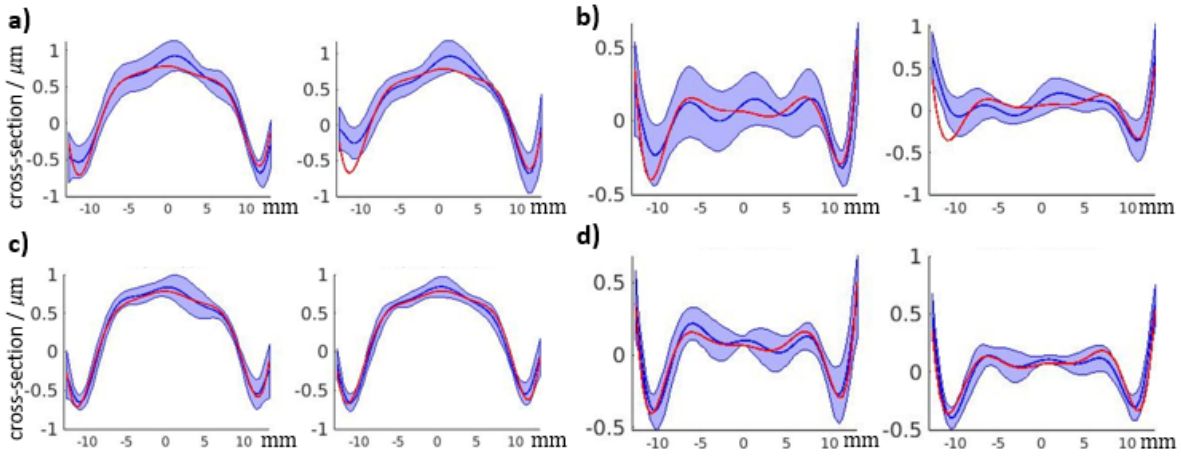

Figure 5.7: The vertical (left) and horizontal (right) cross sections of the reconstructed difference topographies plotted for the state-of-the-art TWI method (red) and the deep learning hybrid approach (blue) together with the latter's uncertainty of the reconstructed topography, i.e., the 95% symmetric credible interval: **a)** without prior misalignment correction; **b)** after subtracting the best fit sphere; and **c)** with prior misalignment correction; **d)** after subtracting the best fit sphere. The plots are taken from Hoffmann et al. (2021a).

m, ranging from 48.7 m to 60.5 m for the individual ensemble member predictions. Finally, the result from the deep learning hybrid approach with prior misalignment correction (Fig. 5.6d) has a RMSD of 211 nm and a RMSE of 82 nm compared to the state-of-the-art method which it covers to 89%. The subtracted best fit sphere has a mean radius of 58 m, ranging from 50.9 m to 66.7 m for the individual ensemble member predictions. Again, the cross sections of the topography profiles from the state-of-the-art method (red) are plotted together with the deep learning hybrid approach and its uncertainty of the reconstructed topography (blue) with and without prior misalignment correction, respectively, in Figures 5.7b and 5.7d.

5.3 Discussion and summary

In Chapter 4, it was shown that the introduced deep learning hybrid approach produces reliable results, which also include a quantification of the uncertainty of the reconstructed topography, for in-domain as well as out-of-distribution test data in a simulated environment. In this chapter, we demonstrate how to use the deep learning approach in a real-world environment and how to apply it to measurement data from a real specimen of an asphere. Given that the ground truth for real specimens is not known, however, the results are compared to the results

of a state-of-the-art method for the optical form measurement of optical aspheres and freeform surfaces (Mahr GmbH, 2021). Higher frequency structures are neglected because the deep learning model is trained on, and produces, low resolution images. Also, higher frequencies and positioning errors were not included in the generated topographies of the training data.

It is difficult to compare the computing efficiency of the two approaches due to their different requirements. Once the deep neural networks are trained for a design topography, and the measurement data of the specific specimen are acquired and preprocessed, the form reconstruction and the described uncertainty quantification are performed instantaneously. In contrast, state-of-the-art TWI methods do not need any training for different design topographies. However, their form reconstruction time depends on different hyperparameters, such as the chosen resolution, the amount of rays to be traced through the optical system, and the number of steps used for the iterative form reconstruction procedure (Fortmeier et al., 2014).

Nonetheless, the results show that the deep learning hybrid approach provides reliable results for the measured data, since the two methods produce similar results, especially when a prior misalignment correction is performed for the input data of the deep learning model. Furthermore, the form reconstructed by the state-of-the-art method lies well within the estimated uncertainty of the reconstructed topography of the deep learning hybrid approach. Hence, Objective 4, i.e. to apply the developed deep learning approach to solve the inverse problem of the TWI on measurement data of a real specimen and to validate it against a state-of-the-art method (cf. Section 1.2), is successfully dealt with.

Future work should include an extended comparison study with other kinds of design topographies and specimens that can be measured by the TWI, e.g. as part of an interlaboratory comparison study (Schachtschneider et al., 2018; Fortmeier et al., 2020). Also, higher frequency parts, misalignment errors and a higher resolution of the form reconstruction could be incorporated in the data generation and investigated for the deep learning hybrid approach in the future.

6

Deep ensembles from a Bayesian perspective

Quantifying[1] the uncertainty associated with the prediction of a trained neural network has its own merit and is particularly relevant when decisions are taken in sensitive applications such as medical diagnosis (Leibig et al., 2017) or autonomous driving (Michelmore et al., 2020). Several methods have been proposed to quantify the uncertainty associated with the prediction of a deep neural network (Abdar et al., 2021; Jospin et al., 2020). Currently, deep ensembles (Lakshminarayanan et al., 2017) can be seen as state-of-the-art for uncertainty quantification in deep learning (cf. Subsection 2.2.4).

Deep ensembles are based on a combination of multiple networks, an idea that dates back to the 20th century. In Hansen and Salamon (1990), better accuracies were achieved for an ensemble of networks than when employing a single network; the use of multiple networks to account for model uncertainty has been proposed in Carney et al. (1999). Ensemble learning has gained in popularity ever since (Sagi and Rokach, 2018). A breakthrough in the application of ensembling for deep learning was the introduction of deep ensembles (Lakshminarayanan et al., 2017), where the prediction of an ensemble of independently trained neural networks is treated as a uniformly weighted Gaussian mixture model. In Ashukha et al. (2020), deep ensembles are even used to introduce a metric to measure the performance of other ensembling methods by counting the number of networks needed to reach a specified performance.

Uncertainty quantification by deep ensembles has been proposed without referring to a Bayesian background and has even been explicitly classified as a non-Bayesian approach (Lakshminarayanan et al., 2017; Ovadia et al., 2019; Hu et al., 2020). Several ideas have been put forward to embed deep ensembles into a Bayesian framework. Examples comprise early stopping (Duvenaud et al., 2016), regularizing parameters around different values drawn from a prior distribution (Pearce et al., 2020) and creating a link to Gaussian processes via the neural tangent kernel (He et al., 2020). In Wilson and Izmailov (2020), deep ensembles are associated to Bayesian model averaging (Hoeting et al., 1999), while in Gustafsson et al.

[1] Most of this chapter is taken from Hoffmann and Elster (2021).

(2020), they are represented as an approach that builds on an approximation of a multimodal posterior for the network parameters.

In following this latter perspective, we argue that deep ensembles provide an approximate Bayesian inference in which the true posterior is replaced with an average of delta distributions around local maximum a posterior (MAP) estimates for the parameters of the network. The statistical model and prior distributions required to achieve this result are specified below. By using an average of delta distributions as an approximation of the posterior, the formulas applied in deep ensembles (Lakshminarayanan et al., 2017) follow directly. To improve this approximation, we replace the delta distributions with Gaussian distributions in this work, which is a technique known from Laplace approximation (Ritter et al., 2018). A similar idea is presented in Wilson and Izmailov (2020), where the approximated posterior distribution is generalized to a mixture of Gaussians which are independently trained as stochastic-weight-averaging Gaussians (SWAG) (Maddox et al., 2019). The approximation of the posterior by a family of Gaussian distributions is well-known from Bayesian neural networks (Lampinen and Vehtari, 2001) and variational inference in general (Blei et al., 2017).

By applying variational inference (Blei et al., 2017), the evidence lower bound (ELBO) is maximized in this work to determine the additional parameters of the Gaussian mixture distribution. However, rather than maximizing the ELBO with respect to all parameters during network training, we propose a two-step approach. First, conventional training of the ensemble of neural networks yields local MAPs of the network parameters, which are taken as the locations of the Gaussian mixture distribution. In the second step, the remaining parameters are determined by maximizing the ELBO. In this way, training of the deep ensembles does not have to be altered, and the improved approximation of the posterior can be obtained via post-processing.

Previous work suggests that model uncertainty can be captured using only a few or even just one Bayesian layer at the end of the network (Zeng et al., 2018; Brosse et al., 2020; Kristiadi et al., 2020). Considering the last layer to be linear and only the weights of that layer to be random, an analytical solution for the maximization of the ELBO can be derived. As a consequence, the proposed variational inference can be easily applied by simple post-processing of the results obtained by conventional deep ensembles.

The improved approximation of the posterior leads to a modification of the formulas used to perform uncertainty quantification by deep ensembles. Specifically, the epistemic part of the uncertainty is enlarged by an additional contribution. Arguing that, in the context of regression problems, the epistemic part of the uncertainty can be the relevant part when the goal is to infer the regression function, the proposed modification can then be necessary to arrive at a reliable uncertainty quantification.

This chapter tackles Objective 5, i.e. to investigate and improve the uncertainty quantification by deep ensembles for regression problems (cf. Section 1.2). Applying the developed approach to both simulated and real data, we demonstrate the impact on the uncertainty quantification by comparing the results with those obtained by conventional deep ensembles. Analytical derivations enable the easy implementation of the proposed approach. The approach and the derived analytical formulas are based on analyzing deep ensembles from

a Bayesian perspective. The resulting statistical formulations are presented and support new interpretations of deep ensembles.

The chapter is structured as follows. In Section 6.1, we specify the conditions under which uncertainty quantification by deep ensembles can be viewed as an approximate Bayesian method and the proposed extension is derived. In Section 6.2, we explore the impact of the proposed approach and compare it to the classical deep ensembles in different experiments. We also apply it to the inverse problem related to the TWI. Finally, the results are summarized and discussed in Section 6.3.

6.1 Theoretical formulas and derivations

An introduction to uncertainty quantification, Bayesian statistics and variational inference is provided in Subsection 2.2.3, while the basic frequentist framework of deep ensembles (Lakshminarayanan et al., 2017) is presented in Subsection 2.2.4. In this work, we focus on regression problems. The following notations are going to be used from now on. Let the data $D = \{(x_i, y_i), i = 1, \dots, N\}$ be independent and identically distributed (i.i.d.) samples following the heteroscedastic regression model given as

$$y|x \sim N\left(\eta_\theta(x), \sigma_\theta^2(x)\mathrm{I}\right) , \tag{6.1}$$

where $x \in \mathbb{R}^{d_x}$, $y \in \mathbb{R}^{d_y}$, $\sigma_\theta^2(x) \in \mathbb{R}$, and the conditional distribution for y given x is taken as a Gaussian distribution with mean $\eta_\theta(x)$ and covariance matrix $\sigma_\theta^2(x)\mathrm{I}$. Both $\eta_\theta(x)$ and $\sigma_\theta^2(x)$ are modeled by deep neural networks. The mean $\eta_\theta(x)$, viewed as a function of x, will be termed the regression function, and its inference is of particular interest. We consider D as the training data.

Subsection 6.1.1 specifies the statistical formulas, the prior distributions and the form of approximation required such that uncertainty quantification by deep ensembles (Lakshminarayanan et al., 2017) can be viewed as an approximate Bayesian method. This approximation is then improved by extending the family of distributions used to approximate the posterior from delta distributions to Gaussian distributions in Subsection 6.1.2. We maximize the ELBO to determine the additional parameters, leading to a simple post-processing procedure of conventional deep ensembles.

6.1.1 Deep ensembles as a Bayesian approximation

In a Bayesian framework (Gelman et al., 1995; Robert, 2007; O'Hagan and Forster, 2004), the posterior distribution $\pi(\theta|D)$ is determined (or approximated) and used for inference. For example, as an estimate of the regression function $\eta_\theta(x)$ and its associated uncertainty one could take the posterior mean and posterior standard deviation of $\eta_\theta(x)$. If one is interested in predicting a future observation y at a specified x, on the other hand, the posterior predictive distribution $\pi(y|x, D)$ should be considered. The posterior predictive distribution accounts for both the uncertainty in the parameters θ and the additional uncertainty of y expressed by the covariance matrix $\sigma_\theta^2(x)\mathrm{I}$.

The total uncertainty associated with the posterior predictive distribution of the neural network predictions can be divided into an aleatoric part and an epistemic part (Gal, 2016; Hüllermeier and Waegeman, 2021). While the former is related to the irreducible part of the uncertainty, the epistemic uncertainty characterizes the uncertainty about the model (i.e. the model parameters θ in this setting). The epistemic uncertainty can be reduced by increasing the size of the available training data. The aleatoric uncertainty is represented in the regression model (6.1) by the covariance matrix $\sigma_\theta^2(x)\mathrm{I}$ and characterizes the uncertainty about the (future) realization of an observation y given its mean $\eta_\theta(x)$.

In the deep ensembles method (Lakshminarayanan et al., 2017), an ensemble of individually trained neural networks is considered. To predict an observation y, the mean of the predictions of the single networks is formed (cf. Subsection 2.2.4). The (squared) uncertainty associated with this prediction is taken as the sum of the covariance matrix of the individual predictions and the estimated aleatoric part of the prediction (i.e. $\sigma_{\widehat{\theta}}^2(x)\mathrm{I}$ in model (6.1)). Deep ensembles were originally introduced as a non-Bayesian method (Lakshminarayanan et al., 2017). However, it has been repeatedly brought up that deep ensembles can be viewed from a Bayesian perspective (Abdar et al., 2021; Gustafsson et al., 2020; Wilson and Izmailov, 2020). More precisely, conventional deep ensembles can be viewed as an approximate Bayesian inference where the posterior $\pi(\theta|D)$ of the model parameters θ given the data D is approximated by a family of delta distributions, i.e.

$$\pi(\theta|D) \approx q(\theta) := \frac{1}{K}\sum_{k=1}^{K} \delta(\theta - \widehat{\theta}^{(k)}) \,, \tag{6.2}$$

while choosing the prior $\pi(\theta)$ for the neural network parameters normally distributed as

$$\theta \sim N\left(0, \lambda^{-1}\mathrm{I}\right) \,, \tag{6.3}$$

where λ is the L^2-regularization parameter for the model parameters.

This is shown in the following based on the statistical model in (6.1). Each neural network of the ensemble is independently trained according to (Lakshminarayanan et al., 2017) by minimizing the following loss function:

$$loss = -L(\theta; D) + \frac{1}{2}\lambda\|\theta\|^2 \,,\; L(\theta; D) = -\frac{1}{2}\sum_{i=1}^{N}\left(\frac{\|y_i - \eta_\theta(x_i)\|^2}{\sigma_\theta^2(x_i)} + d_y \log\left(\sigma_\theta^2(x_i)\right)\right) \,, \tag{6.4}$$

where $\|.\|$ is the L^2-norm, and $L(\theta; D)$ equals (up to a constant) the log likelihood for the statistical model in (6.1). The parameter estimates $\widehat{\theta}^{(k)}, k = 1, \ldots, K$, obtained for the ensemble of K networks, are at the same time (local) estimates for the maximum a posteriori probability (MAP) of θ when assigning the prior (6.3). By taking the average of K probability mass functions at the local MAPs as an approximation of the posterior $\pi(\theta|D)$ (cf. (6.2)), the approximation of the posterior predictive distribution for the statistical model in (6.1) is then given by

$$\pi(y|x, D) = \int \pi(\theta|D) N\left(y; \eta_\theta(x), \sigma_\theta^2(x)\mathrm{I}\right) \mathrm{d}\theta \approx \frac{1}{K}\sum_{k=1}^{K} N\left(y; \eta_{\widehat{\theta}^{(k)}}(x), \sigma_{\widehat{\theta}^{(k)}}^2(x)\mathrm{I}\right) \,, \tag{6.5}$$

which equals the expression given in (Lakshminarayanan et al., 2017) for deep ensembles.

By using the approximation (6.5), mean and covariance matrix of the posterior predictive distribution are obtained as

$$E\left(y|x,D\right) = \frac{1}{K}\sum_{k=1}^{K}\eta_{\widehat{\theta}^{(k)}}(x)\,, \tag{6.6}$$

$$Cov\left(y|x,D\right) = \frac{1}{K}\sum_{k=1}^{K}\left\{\left(\eta_{\widehat{\theta}^{(k)}}(x) - E\left(y|x,D\right)\right)\left(\eta_{\widehat{\theta}^{(k)}}(x) - E\left(y|x,D\right)\right)^{\top} + \sigma^2_{\widehat{\theta}^{(k)}}(x)\mathrm{I}\right\}\,, \tag{6.7}$$

where E stands for expectation and Cov for covariance, cf. Appendix B.1. Note that equation (6.6) is a prediction of a future observation at x and yields at the same time the inferred regression function $\eta_\theta(x) = E\left(y|x,D\right)$ according to the statistical model (6.1). Equation (6.7) describes the total uncertainty associated with the prediction of the neural network ensemble. Mean and covariance matrix of the posterior predictive distribution are in accordance with the results given in Lakshminarayanan et al. (2017) for the original approach of the deep ensembles.

In Lakshminarayanan et al. (2017), the comparison between different methods is based on the posterior predictive distribution. The covariance matrix (6.7) comprises the reducible epistemic part of the uncertainty as well as the irreducible aleatoric part. However, when the goal is to infer the underlying regression function, an accurate estimation of the epistemic uncertainty is essential. The epistemic uncertainty is given by the spread of the single estimates of the regression function obtained by the ensemble of trained networks. Mean and covariance matrix of the regression function are deduced in the following lemma.

Lemma 1. *The approximate Bayesian inference associated with deep ensembles yields a posterior $\pi(\eta|x,D)$ for the regression function $\eta \equiv \eta_\theta(x)$ given by*

$$\pi(\eta|x,D) = \frac{1}{K}\sum_{k=1}^{K}\delta(\eta - \eta_{\widehat{\theta}^{(k)}}(x))\,, \tag{6.8}$$

with mean and covariance matrix given by

$$E\left(\eta|x,D\right) = \frac{1}{K}\sum_{k=1}^{K}\eta_{\widehat{\theta}^{(k)}}(x)\,, \tag{6.9}$$

$$Cov\left(\eta|,x,D\right) = \frac{1}{K}\sum_{k=1}^{K}\left\{\left(\eta_{\widehat{\theta}^{(k)}}(x) - E\left(\eta|x,D\right)\right)\left(\eta_{\widehat{\theta}^{(k)}}(x) - E\left(\eta|x,D\right)\right)^{\top}\right\}\,. \tag{6.10}$$

Proof. The proof is given in Appendix B.2. □

Note that the expectation of the posterior for the regression function (6.9) equals the expectation of the posterior predictive distribution (6.6), while its covariance matrix (6.10) contains only the epistemic part of the covariance matrix of the posterior predictive distribution (6.7).

The average over delta distributions taken at point estimates of the parameters θ (6.2) is a crude approximation of the posterior. Nonetheless, deep ensembles outperform many other Bayesian approaches due to their ability to explore different modes of the posterior distribution

(Fort et al., 2019). Figure 6.1, which is inspired by Fort et al. (2019) (Fig. 1) and Wilson and Izmailov (2020) (Fig. 3), illustrates this behavior.

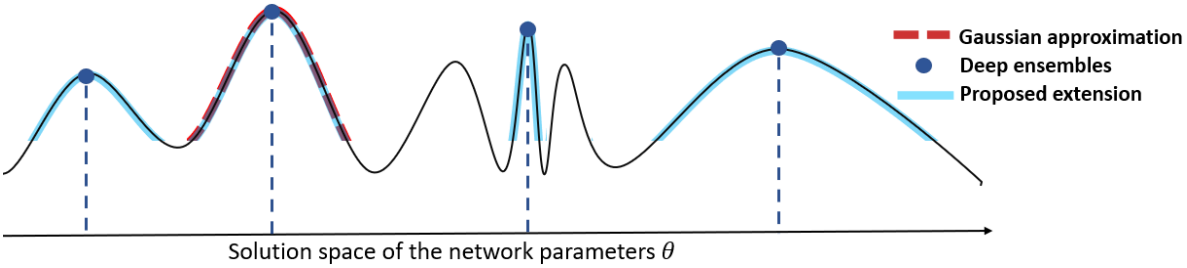

Figure 6.1: This example figure illustrates the solution space of the network parameters on the x-axis and the negative loss function on the y-axis. Deep ensembles explore multiple modes in contrast to other Bayesian approaches which focus on a single mode but consider more complex distributions. The proposed extension combines these two ideas. The figure is taken from Hoffmann and Elster (2021).

6.1.2 Proposed extension to Gaussian mixture distributions

When the aleatoric part of the covariance matrix of the posterior predictive distribution (6.7) is dominant, the covariance matrix of the regression function (6.10) is expected to be significantly smaller than that of the posterior predictive distribution (cf. example in Figure 6.3 and results in Table 6.2). It follows that the (crude) approximation of the posterior in (6.2) with a family of delta distributions, which ignores the nonzero width of the true posterior around its (local) MAPs, is irrelevant for the total uncertainty in these cases. However, this approximation can lead to a significant underrating of the epistemic uncertainty.

An improved approximation of the posterior distribution $\pi(\theta|D)$ is given by

$$q(\theta) = \frac{1}{K} \sum_{k=1}^{K} N(\theta; \widehat{\theta}^{(k)}, \gamma_k \mathrm{I}) \,, \tag{6.11}$$

where the γ_k correspond to the variances around the (local) MAPs $\widehat{\theta}^{(k)}$. The approximation (6.11) of the posterior extends the approximation in (6.2) by allowing for a finite width of the posterior around the (local) MAPs, and it reduces to (6.2) for $\gamma_k \to 0$.

The general idea behind variational inference (Blei et al., 2017) is to approximate the posterior $\pi(\theta|D)$ by a distribution that can be more easily handled. Specifically, one can seek the best approximation of the posterior within a parametric family of distributions such as the Gaussian mixture distribution in (6.11). One way to do so is to use a Laplace approximation, for which the parameters of a single Gaussian approximation are calculated from the local properties of the posterior at the MAP (Ritter et al., 2018). In deep learning approaches, however, the parameters are typically determined by maximizing the evidence lower bound (ELBO) defined as follows (cf. Subsection 2.2.3):

$$\mathrm{ELBO}(q) := E_q \left[\log\left(p(D|\theta)\right)\right] - \mathrm{KL}\left(q(\theta) \| \pi(\theta)\right) = \log\left(\pi(D)\right) - \mathrm{KL}\left(q(\theta) \| \pi(\theta|D)\right) . \tag{6.12}$$

Instead of maximizing the ELBO with respect to all parameters during network training, we suggest here a two-step procedure. First, the networks are independently trained in

accordance with the conventional deep ensembles training procedure. The resulting (local) MAPs of the network parameters are taken as the first moments of the Gaussian mixture distributions. Second, the remaining parameters of (6.11), which are the variances γ, are deduced by maximizing the ELBO from equation (6.12) in a post-processing step. This can be done numerically for example via a grid search.

In the following, assume that the only random weights in the network are those in the last layer. Furthermore, we will consider only the randomness of $\eta_\theta(x)$, while $\sigma^2_\theta(x)$ remains a point estimate for easier formulas, but could be extended in a similar manner. Also, σ^2_θ would not depend on the last layer but rather be an individual network parameter, when choosing a homoscedastic approach instead, which could be an alternative to the heteroscedastic model in (6.1). Figure 6.2 visualizes the parameters treated as random in the proposed approach. Maximizing the ELBO can then be performed analytically, as stated in Theorem 1 below. Mean and covariance matrix of the regression function for the improved approximate Bayesian inference are given in Theorem 2, while Algorithm 1 summarizes the proposed approach.

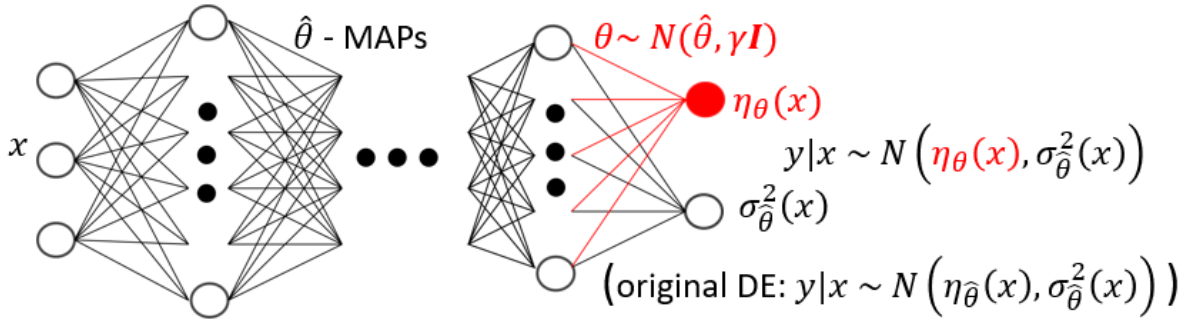

Figure 6.2: The proposed model of a single neural network is visualized here. A sample $y|x$ from the posterior predictive distribution is drawn from the network outputs $N(\eta_\theta(x), \sigma^2_{\hat{\theta}}(x))$. All network weights are fixed to the MAP estimates after training, except for the last linear unit of the predicted mean value $\eta_\theta(x)$. The weights of this last linear unit are normally distributed around the trained MAPs $\widehat{\theta}$ with variances $\gamma_k, k = 1, \ldots, K$. The figure is taken from Hoffmann and Elster (2021).

Theorem 1. *Let only the weights of the last layer in the neural network, which are related to the prediction of the regression function $\eta_\theta(x)$, be random and assume a linear transformation from the last layer to the network output. Approximate maximization of the ELBO* (6.12) *can then be done analytically when considering the approximation of the posterior distribution given in* (6.11)*, leading to the expression* (B.20) *in the Appendix. Furthermore, the variances in* (6.11) *tend to zero for an infinite amount of training data.*

Proof. The proof is given in Appendix B.3. □

Theorem 2. *The improved approximation for the Bayesian inference associated with deep ensembles (from delta to normal distributions, cf. equation* (6.11)*) yields a posterior $\pi(\eta|x, D)$ for the regression function $\eta \equiv \eta_\theta(x)$ given by*

$$\pi(\eta|x, D) = \frac{1}{K} \sum_{k=1}^{K} \int \delta(\eta - \eta_\theta(x)) N\left(\theta; \widehat{\theta}^{(k)}, \gamma_k I\right) \mathrm{d}\theta \,, \tag{6.13}$$

with mean and covariance matrix given approximately by

$$E\left(\eta|x,D\right) \approx \frac{1}{K}\sum_{k=1}^{K}\eta_{\widehat{\theta}^{(k)}}(x)\,, \tag{6.14}$$

$$Cov\left(\eta|x,D\right) \approx \frac{1}{K}\sum_{k=1}^{K}\left\{\left(\eta_{\widehat{\theta}^{(k)}}(x)-E\left(\eta|x,D\right)\right)\left(\eta_{\widehat{\theta}^{(k)}}(x)-E\left(\eta|x,D\right)\right)^{\top}+\gamma_k J_k J_k^{\top}\right\}, \tag{6.15}$$

where $J_k = \partial\eta/\partial\theta$ is evaluated at $\theta = \widehat{\theta}^{(k)}$. The Jacobian matrix J_k can be easily calculated in terms of the output of the next-to-last layer when considering only the weights of the last layer in the neural networks to be random, and when the last layer is linear.

Proof. The proof is given in Appendix B.4. □

Algorithm 1: improved approximation of the posterior $\pi(\theta|D)$

Result: $q(\theta) := \frac{1}{K}\sum_{k=1}^{K} N(\theta;\widehat{\theta}^{(k)},\gamma_k \mathrm{I})$, instead of $q(\theta) := \frac{1}{K}\sum_{k=1}^{K}\delta(\theta-\widehat{\theta}^{(k)})$;

1. initialize and independently train networks $\eta_{\theta^{(k)}}$, $k = 1,\ldots,K$, in accordance with the deep ensembles approach (Lakshminarayanan et al., 2017);
2. compute γ_k, $k = 1,\ldots,K$, according to (B.20).

The improved approximation of the Bayesian inference associated with deep ensembles yields the same expectation (6.14) for the regression function as the original approximation (6.9), together with a covariance matrix (6.15) that includes the additional term $\gamma_k J_k J_k^{\top}$ (compared to (6.10)). In this way, the epistemic part of the uncertainty of the conventional deep ensembes is enlarged.

Samples from the posterior (6.13) for the regression function can be easily obtained by first sampling a $k \in \{1,\ldots,K\}$ with probability $\frac{1}{K}$. Second, a $\theta^{(k)}$ is sampled from $N(\widehat{\theta}^{(k)},\gamma_k \mathrm{I})$ and third, $\eta_{\theta^{(k)}}(x)$ is evaluated. To sample from the posterior predictive distribution, a fourth step is added: sample a y from $N\left(\eta_\theta(x),\sigma^2_{\hat{\theta}^{(k)}}(x)\mathrm{I}\right)$ of the selected kth network. The sampling procedure is summarized in Algorithm 2.

Algorithm 2: sample from the posterior of the regression function $\pi(\eta|x,D)$ and from the posterior predictive distribution $\pi(y|x,D)$

Result: sample from $\pi(\eta|x,D)$: steps 1-3; sample from $\pi(y|x,D)$: steps 1-4;

1. sample kth network uniformly from $k \in \{1,\ldots,K\}$;
2. sample network weights $\theta^{(k)}$ from $N(\widehat{\theta}^{(k)},\gamma_k \mathrm{I})$;
3. evaluate $\eta_{\theta^{(k)}}(x)$;
4. evaluate $\sigma^2_{\hat{\theta}^{(k)}}(x)$ and sample from $N\left(\eta_{\theta^{(k)}}(x),\sigma^2_{\hat{\theta}^{(k)}}(x)\mathrm{I}\right)$.

6.2 Experiments

In this section, the classical deep ensembles (DE) (Lakshminarayanan et al., 2017) are compared to the proposed extended approach (DE extended). Following Algorithm 1, the neural network training procedure is exactly the same for the classical and the extended deep ensembles. The

inferred regression function is identical in both cases, namely the average over the estimates of all ensemble members $\frac{1}{K}\sum_{k=1}^{K}\eta_{\hat{\theta}^{(k)}}(x)$ at the trained MAPs $\widehat{\theta}$. The only difference is the additional term of the extended covariance matrix in (6.15), which is related to the epistemic part of the uncertainty. Uncertainties are understood as the half width of the 95% credible intervals calculated in terms of an assumed Gaussian distribution using the approximate variance formulas (i.e. 1.96 times the standard deviation). Coverage probabilities refer to the number of test cases that lie within the estimated, symmetric around the inferred regression function, 95% credible interval. Training and test data are always disjoint and all obtained results are shown in respect to the test set.

In Subsection 6.2.1, the deep ensembles method and its proposed extension are compared on simulated toy regression problems with known ground truth. The implemented code, together with additional examples, is provided[2]. Then, the results of the two methods are compared on real data in Subsection 6.2.2. The architecture of all neural networks in the above mentioned experiments is fully connected with three hidden layers consisting of 128, 64, and 32 neurons, respectively. Eventually, in Subsection 6.2.3, we apply the proposed extension to deep ensembles for solving the inverse problem related to the TWI.

All implementations in this section are realized in Python (version 3.6.9) (Van Rossum and Drake, 2009) using pytorch (version 1.6.0) (Paszke et al., 2019). The weights of the employed neural networks are initialized by randomly drawing from the kaiming uniform distribution (He et al., 2015) which is currently the default initialization for convolutional and linear layers in pytorch. Dispersing the initial values of the individual neural networks before starting the model training is important to ensure unbiased solutions.

6.2.1 Toy regression problems

First, numerical experiments are carried out on some toy regression examples. Results are shown for two univariate regression tasks with a one and two dimensional input domain, respectively. Many more examples can be performed and parameter settings tested using the provided GitLab-repository[2]. All data of the output space are normalized by subtracting the mean and dividing the difference by the standard deviation of the corresponding training set. The input data are uniformly drawn from the elementwise support $[-1, 1]$. The training and test sets are disjoint. All presented results refer to the test data.

The hyperparameters and the network architecture are not fine-tuned, since the achieved prediction accuracies are the same for both methods and the focus lies on the quantification of the epistemic uncertainty. The following hyperparameter settings are equal for all presented results, while the individual parameter settings are described below. Training is performed over 60 epochs with a learning rate drop factor of 0.1 for the last five epochs and a mini batch size of 64.

Figure 6.3 shows the predicted regression function and the resulting total and epistemic uncertainties for the classical and extended deep ensembles together with the ground truth. The data $y(x) \in \mathbb{R},\ x \in [-1, 1]$, are normally distributed around the regression function $\eta(x) = \frac{1}{2}\left((4.5x)^4 - (18x)^2 + 22.5x\right)$ with a standard deviation of 10. An ensemble of 10 networks is trained using 200 data samples with a learning rate equal to 0.005, and the

[2]BayesianDeepEnsembles (https://gitlab1.ptb.de/hoffma31/bayesiandeepensembles)

regularization parameter λ (cf. equation (6.4)) is set to 0.01. The total uncertainty is much greater than the epistemic part of the uncertainty for the classical deep ensembles, when comparing the first two images of the first row. This means that the aleatoric part of the uncertainty is dominant there. The total uncertainty almost perfectly covers the underlying disturbed data and hence, the approximation of the posterior predictive distribution works well for both approaches. However, the second column shows that the uncertainty for the inferred regression function covers the ground truth in less than 60% of the considered test cases for the conventional deep ensembles, i.e. conventional deep ensembles underrate the uncertainty. In contrast, the extended deep ensembles cover the regression function completely, as shown in the second row (second and third image), and hence yield a more reliable, yet to some extent conservative, uncertainty quantification.

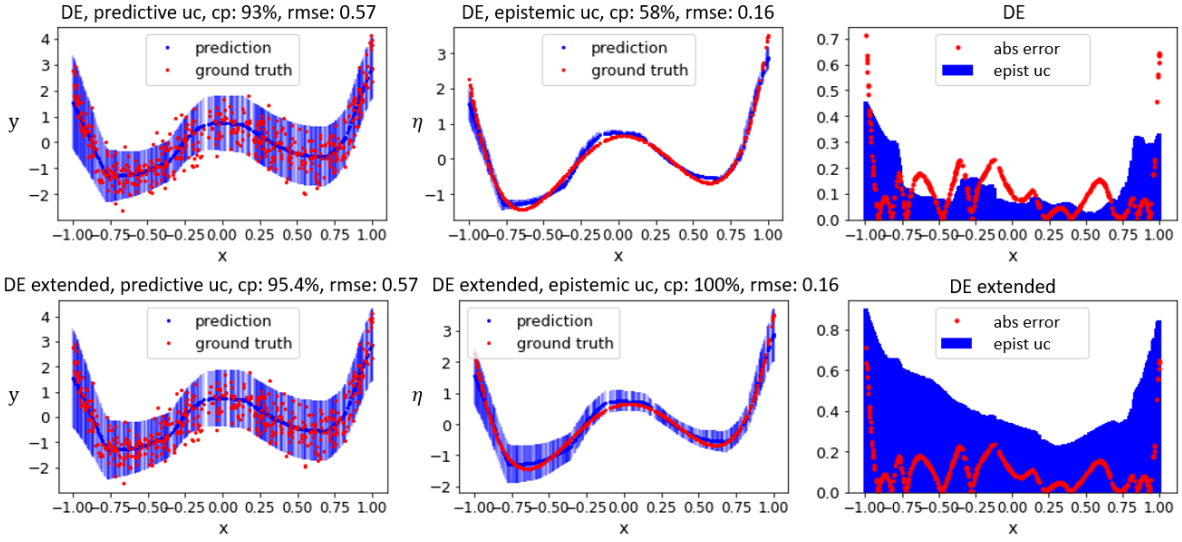

Figure 6.3: The results are plotted for the classical deep ensembles (first row) and the proposed extension (second row). The coverage probabilities (cp) and root mean squared errors (rmse) are estimated for the posterior predictive distribution (first column) and the regression function (second column). The estimates, predictions and corresponding 95% credible intervals are plotted (in blue) together with the underlying disturbed and undisturbed ground truth data (in red). In the third column, the epistemic uncertainties are plotted together with the absolute errors of the estimated regression function to the ground truth. The plots are based on Hoffmann and Elster (2021).

Another comparison between the classical and extended deep ensembles is provided in Figure 6.4, which analyzes the dependence of the amount of training data and the number of trained ensemble members on the epistemic uncertainty. A more complex regression function with a two dimensional input domain is chosen here as an example. Nonetheless, the regression function from the previous example leads to similar significant results which can be easily checked[2]. The data samples $y(x) \in \mathbb{R}$, $x \in [-1,1]^2$, are normally distributed around the regression function $\eta(x) = \sum_{i=1}^{2}(1.5x_i - 1)^2(1.3x_i + 1)^2$ with a standard deviation of 0.2. The results in Figure 6.4 are produced using a learning rate of 0.001 and a regularization parameter of 0.1. 10 neural networks are employed to generate the results in the first column, while the ensembles yielding the plots in the second column are trained using 600 data samples. Our findings show that the extended approach consistently outperforms the uncertainty quantification of the classical deep ensembles with a coverage probability of over 80% in almost

every setting. Even for a single network, the coverage of the extended method is about 60%, while there is no epistemic uncertainty at all in the classical method for this case. In this example, the uncertainty remains almost constant when the number of neural networks for the ensemble is increased beyond ten ensemble members. The uncertainties of the two approaches are expected to converge for an infinite amount of training data (cf. Theorem 1), which seems to be supported by the observed results. Finally, note that the root mean squared error (RMSE) of the predicted regression function is the same for the two approaches since the same estimates are reached. It decreases for an increasing amount of training data and is more stable for an increasing number of ensemble members.

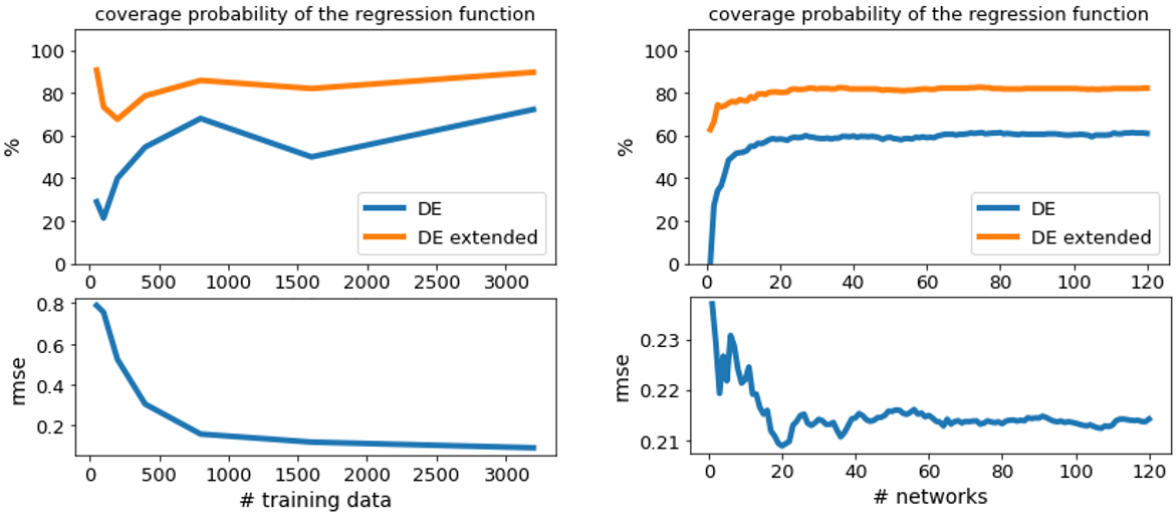

Figure 6.4: The coverage probabilities for the regression function of the classical deep ensembles approach and its suggested extension are plotted in the first row in dependence on an increasing amount of training data and an increasing number of ensemble members, respectively. The second row shows the corresponding root mean squared errors (which are the same for both approaches). The plots are based on Hoffmann and Elster (2021).

6.2.2 Real data

The proposed extension is tested on four real datasets downloaded from the UCI database (Dua and Graff, 2017). A summary of the datasets (e.g. amount of training and test data) and the chosen hyperparameters for the network training are given in Table 6.1. All data are normalized, i.e. the mean is subtracted and the difference is divided by its standard deviation. All networks are trained using the Adam optimizer (Kingma and Ba, 2015) and a constant learning rate up to the last five epochs where the learning rate drops by a factor of 0.5 in every epoch.

The results are shown in Table 6.2 for the conventional deep ensembles (Lakshminarayanan et al., 2017) and the proposed extension. All investigations are conducted for an ensemble of five and ten neural networks, respectively. Note that the RMSE of the predicted regression function based on the given (noisy) target values and the aleatoric uncertainty are the same for both methods since the proposed extension has only an effect on the epistemic uncertainty via the post-processing step. The epistemic and total coverages describe the relative amount of target data that lie in the 95% symmetric credible intervals. The epistemic coverage relates only to the coverage of the epistemic uncertainty, while the total coverage relates to the

Table 6.1: The table provides general information about the considered datasets taken from the UCI database (Dua and Graff, 2017) and specifies the hyperparameters used for the neural network training. λ is the L^2-regularization parameter for the neural network parameters (cf. equation (6.4)). The output dimension equals one for all datasets.

data information

name	QSAR aquatic toxicity	Yacht hydrodynamics	Blog feedback	Year prediction MSD
reference	Cassotti et al. (2014)	Gerritsma et al. (1981); Ortigosa et al. (2007)	Buza (2014)	Bertin-Mahieux et al. (2011)
# training data	437	246	$52,397$	$463,715$
# test data	109	62	$7,624$	$51,630$
input dimension	8	6	276	90
hyperparameters for neural network training				
learning rate	10^{-3}	10^{-3}	10^{-5}	$5 * 10^{-5}$
λ	10^{-2}	10^{-3}	10^{-5}	10^{-4}
mini batch size	128	8	64	128
epochs	60	100	50	40

coverage of both the epistemic and the aleatoric uncertainties, i.e. the total uncertainty. An automatic uncertainty estimation as proposed here is expected to have a total coverage of about 95%, which is roughly true for both variants in the examples considered here. The epistemic uncertainty, however, relates to the distance of the inferred regression function to the underlying ground truth, which is unknown. The epistemic coverage therefore refers to the same noisy target values as the total coverage, and is expected to be smaller than 95% and much smaller in cases where the aleatoric uncertainty strongly dominates.

The ratio between the epistemic and aleatoric variances describes the proportion of the two estimated uncertainty sources. Furthermore, this ratio shows how the epistemic part of the uncertainty changes for the proposed extension in comparison to the original method. In the considered examples, the aleatoric part dominates the uncertainty, because the ratio of the epistemic and aleatoric variances is always smaller than one. However, the proposed extension has a significant impact on the estimated epistemic uncertainties in the first two datasets (QSAR aquatic toxicity and yacht hydrodynamics) where only comparatively few data samples are available. The ratios show an increase of the mean epistemic variance up to a factor of three. Here, the increase of the epistemic uncertainty and the coverage of the epistemic uncertainty for the proposed extension of deep ensembles can be considered an improvement, because the coverage of the total uncertainty remains well calibrated without being further increased. In contrast, the proposed extension shows no effect on the last dataset (year prediction MSD), which means that the proposed approximation of the posterior of the

Table 6.2: The results are shown for the classical deep ensembles and the proposed extension for the four datasets from Table 6.1. The root mean squared error (RMSE) of the predicted regression function and the epistemic and total coverages, i.e. the relative amount of target data that lie in the 95% symmetric credible intervals, are based on the given (noisy) target values. The ratio describes the mean over the test set of the estimated epistemic variances divided by the estimated aleatoric variances.

method		**deep ensembles**		**proposed extension**	
# ensemble members		5	10	5	10
QSAR aquatic toxicity	RMSE	0.555	0.557	0.555	0.557
	epistemic coverage in %	24.8	28.4	42.2	47.7
	total coverage in %	95.4	95.4	95.4	95.4
	ratio epistemic-aleatoric var	0.030	0.036	0.088	0.098
Yacht hydro-dynamics	RMSE	0.085	0.078	0.085	0.078
	epistemic coverage in %	75.8	87.1	82.3	90.3
	total coverage in %	98.4	98.4	98.4	98.4
	ratio epistemic-aleatoric var	0.284	0.383	0.412	0.509
Blog feedback	RMSE	0.684	0.683	0.684	0.683
	epistemic coverage in %	41.0	44.7	41.6	45.0
	total coverage in %	96.8	96.8	96.8	96.8
	ratio epistemic-aleatoric var	0.093	0.092	0.094	0.092
Year prediction MSD	RMSE	0.795	0.795	0.795	0.795
	epistemic coverage in %	20.4	22.8	20.5	22.8
	total coverage in %	95.2	95.3	95.2	95.3
	ratio epistemic-aleatoric var	0.024	0.027	0.024	0.027

network parameters (6.11) through a family of normal distributions can safely be replaced with a family of delta distributions according to the classical deep ensembles (6.2). This finding is also in accordance with the expectation that widths of the normal distributions tend to zero when the amount of training data become large (cf. Theorem 1).

Figure 6.5 shows additional results for the third dataset (blog feedback). More precisely, coverages of epistemic uncertainties are plotted in the first row for an increasing amount of training data and an increasing amount of ensemble members, respectively, together with the corresponding root-mean-squared errors in the second row. In the first column, the results are provided for an ensemble of five neural networks, while 5240 data samples are used for training in the second column. Note that the x-axis in the first column shows the logarithm of the number of data samples used for training. More detailed information are provided in Tables 6.3 and 6.4.

The upper left plot in Figure 6.5 shows that the coverage of the epistemic uncertainty is much larger for the proposed extension than for the classical deep ensembles when only a small

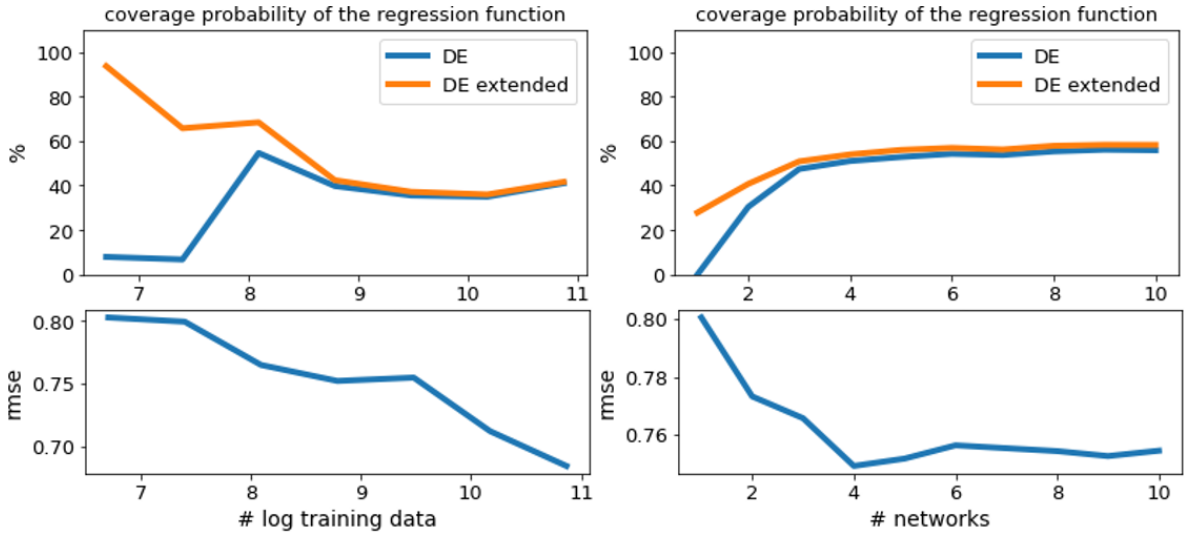

Figure 6.5: The plots are based on the blog feedback dataset. Coverage probabilities for the epistemic uncertainties obtained by classical deep ensembles and their suggested extension are plotted in the first row in dependence on an increasing amount of training data and an increasing number of ensemble members, respectively. The second row shows the corresponding RMSE (which are the same for both approaches). 5 ensemble members are used in the first column, while the neural network ensemble is trained on 5240 data samples in the second column. The figures are taken from Hoffmann and Elster (2021).

amount of training data are available, while converging to the same value for a growing number of data samples in the training data. This behaviour can be explained by considering the corresponding epistemic uncertainties for the two methods shown in Figure 6.6. The epistemic uncertainties of the proposed extension are much larger than of the classical deep ensembles for a small amount of training data. Also, they decrease significantly for a growing amount of training data. The epistemic uncertainties of the two methods converge to a common value for an increasing number of training data, as expected (cf. Theorem 1).

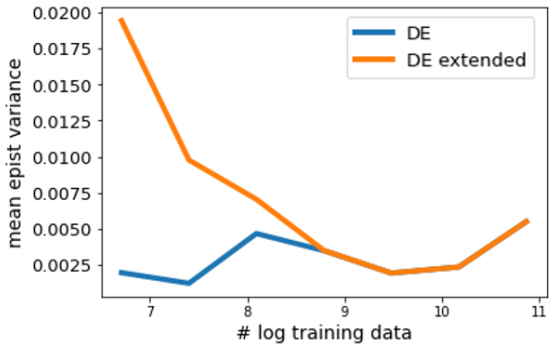

Figure 6.6: The mean of the epistemic variances predicted by deep ensembles and the proposed extension, respectively, are plotted together with the logarithm of the number of data samples used for training an ensemble consisting of 5 neural networks on the blog feedback dataset. The figure is taken from Hoffmann and Elster (2021).

Unfortunately, the ground truth is not known for real data which prevents a quantitative comparison of the results of the two approaches. Nonetheless, it can be assumed that the inferred regression function is less well approximated for a smaller number of training data, because of the larger root-mean-squared errors in Figure 6.5 (lower left plot). Therefore, a larger model uncertainty is more reliable than a small one. Furthermore, the coverage of the

Table 6.3: Results for the blog feedback dataset per amount of training data for 5 ensemble members, cf. first column of Figure 6.5. The table shows the number of training data, the root-mean squared errors, and the coverage probabilities for the epistemic uncertainties and total uncertainties, respectively.

# training data		819	1,638	3,275	6,550	13,099	26,198	52,397
RMSE		0.803	0.8	0.765	0.752	0.755	0.712	0.684
epist. coverage (in %)	DE	8	6.8	54.6	39.8	35.6	35	41
	extension	93.6	65.7	68.4	42.4	37.2	36	41.6
total coverage (in %)	DE	98	98	98.1	97.1	96.9	97	96.8
	extension	98.1	98.1	98.1	97.1	96.9	97	96.8

Table 6.4: Results for the blog feedback dataset per amount of ensemble members for 5240 training samples, cf. second column of Figure 6.5. The table shows the number of neural networks in the deep ensemble, the root-mean squared errors, and the coverage probabilities for the epistemic uncertainties and total uncertainties, respectively.

# ensemble members		1	2	3	4	5	8	10
RMSE		0.801	0.773	0.766	0.749	0.752	0.754	0.754
epist. coverage (in %)	DE	0	30.4	47.4	51	52.8	55.3	55.9
	extension	27.7	40.7	50.8	54	56	57.8	58.2
total coverage (in %)	DE	97.3	97.3	97.6	97.6	97.6	97.6	97.5
	extension	97.3	97.3	97.6	97.6	97.6	97.6	97.5

total uncertainty remains well calibrated in the sense that it is the same for both methods, as shown in Table 6.3. A possible explanation for the behavior of the estimated epistemic uncertainty of the classical deep ensembles is the following. If the neural networks are updated in the same general direction during training and have not converged towards individual local minima yet, then the predictions of the different ensemble members might not vary much and the delta distributions are insufficient to approximate the posterior and hence to describe the complete epistemic uncertainty. Of course, even if local solutions have been found correctly, the widths around the point estimates might actually be significantly greater than zero.

The upper right plot in Figure 6.5 shows that it is an advantage to apply the proposed extension also for a single trained neural network, even if no ensembling approach is pursued. Altogether, it can be concluded that in the considered cases the proposed extension to deep ensembles has the most significant impact on the reliability of the epistemic uncertainty quantification when only limited data samples are available for training or the ensembles consist of a few neural networks only, e.g. due to limited computational capacity and time.

6.2.3 Tilted-wave interferometer

The proposed extension will now be applied for solving the inverse problem related to the TWI on simulated data (cf. Section 3.2). In analogy to Chapter 4, the goal is to reconstruct the corresponding difference topography ΔT for given differences ΔL of optical path length differences (OPDs) for a fixed design topography. Again, we select an asphere (cf. Appendix A.1) as design topography, up to $40,000$ data samples are available for training the neural networks (cf. Subsections 3.2.2 and 4.2.1), and the test data are the 30 samples shown in Figure 4.4.

To analytically compute the variances around the (local) MAPs of the model parameters, only the weights of the last layer are treated as random variables and the last layer is taken to be linear allowing only linear transformations to the model output (cf. Theorem 1). Therefore, in contrast to Chapter 4, a different type of deep learning model is chosen here, as presented in Figure 6.7. Instead of directly predicting the height values of the topography, the neural networks learn to predict Zernike coefficients $c_\theta(x) \in \mathbb{R}^Q$, providing a topography parameterized by Zernike polynomial functions, i.e. $\hat{y} = \sum_{j=1}^{Q} c_\theta(x)_j Z_j$ (cf. Subsection 3.1.2). Remember that Zernike coefficients are drawn randomly for up to the first 136 Zernike polynomial functions (i.e. $n, m = 15$) to generate the training data. The Zernike coefficients of the first 153 Zernike polynomial functions (i.e. $n, m = 16$) are predicted in order to ensure sufficient complexity of the model class to cover the ground truth.

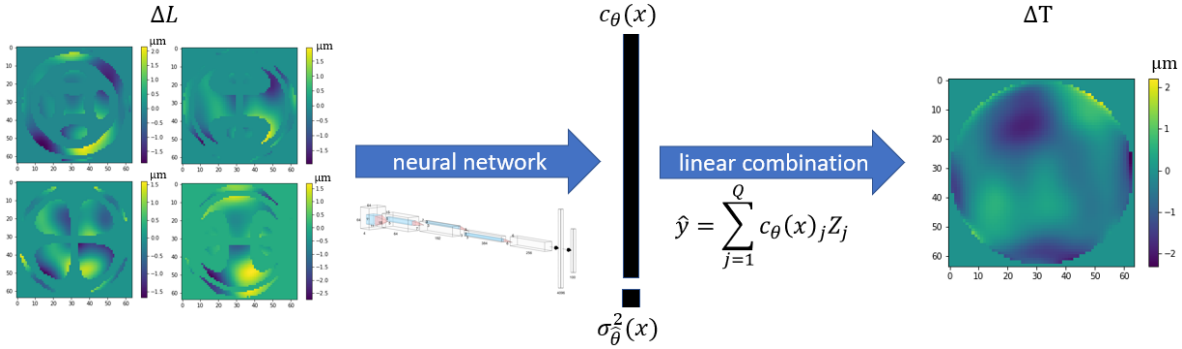

Figure 6.7: The individual ensemble members learn to predict Zernike coefficients $c_\theta(x) \in \mathbb{R}^Q$ and a scalar valued heteroscedastic variance according to the assumed regression model in (6.1) for a four-channel input of differences of OPDs, i.e. $x = \Delta L$. The neural network architecture is presented in more details in Figure 6.8. The output $\hat{y} = \Delta T$ is the estimated difference topography provided by a parameterization by Zernike polynomials using the estimated Zernike coefficients.

Figure 6.8 shows the architecture of the neural networks in more detail. In addition to the $Q = 153$ Zernike coefficients $c_\theta(x)$, the neural networks learn to predict the scalar valued variance $\sigma^2_\theta(x)$ according to the assumed regression model in (6.1) using the loss function from (6.4). Including the additional linear combination between the Zernike coefficients and their polynomial functions, the corresponding formulas are provided in Appendix B.5. Here, the extended epistemic part of the uncertainty is computed by sampling 100 times from the posterior of the inferred regression function (cf. steps 1-3 in Algorithm 2). This is done because of occurring memory issues due to the high dimensionality of the Jacobian when calculating the covariance matrix of the inferred regression function (6.15) from Theorem 2 (cf. Appendix B.5, equation (B.32)). Recall that the variance γ around the (local) MAPs of the neural network parameters is taken as a scalar and therefore, sampling 100 times is more than sufficient.

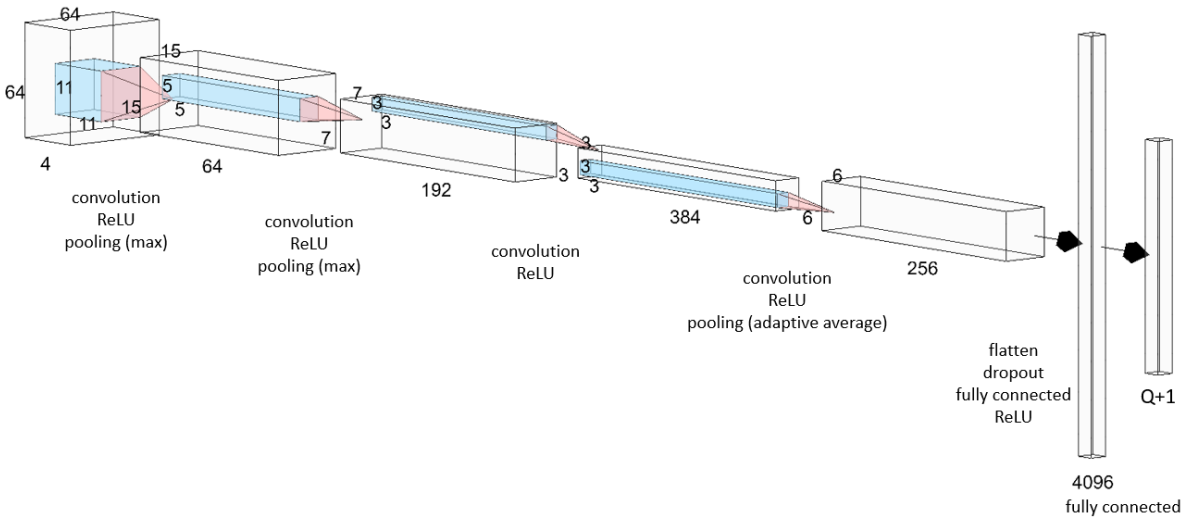

Figure 6.8: The architecture of the neural networks is based on the AlexNet (Krizhevsky et al., 2012). The chosen heights, widths, depths, layer structures and other components are presented (Paszke et al., 2019). The plot is produced using LeNail (2019).

The individual ensemble members are trained for 40 epochs using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 10^{-4} that decreases with a learning rate drop factor of 0.5 in every epoch for the last five epochs. The mini batch size equals 32 and the regularization parameter is set to 10^{-4}. All data are normalized for the neural network training and prediction in the sense that they are divided by the standard deviation over all pixels in the training data in the input and output domain, respectively.

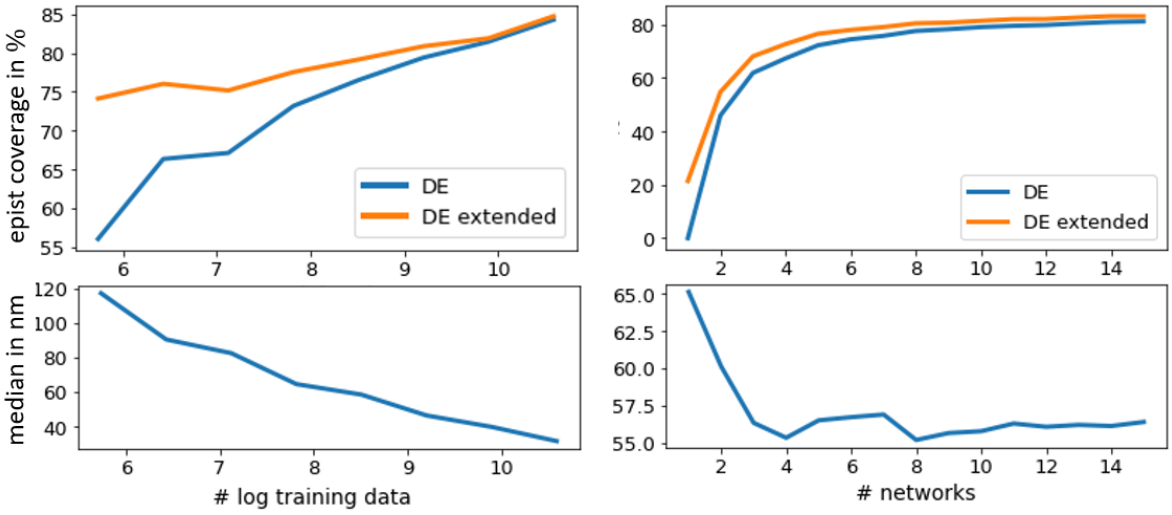

Figure 6.9: The coverage probabilities of the epistemic uncertainties are plotted for the conventional deep ensembles and the proposed extension (first row) together with the median absolute errors averaged over the test samples (second row). The results are compared for an ensemble of 8 neural networks and an increasing number of training data (first column) as well as for an increasing number of ensemble members trained on almost 4000 data samples (second column).

Results are shown in Figure 6.9. The first row shows the coverage probabilities for the epistemic uncertainties of the conventional and extended deep ensembles, while the second row presents the corresponding averages over the test set of the median absolute errors between the estimated topographies and their ground truth. Note that the average of the median absolute

deviations of the topographies in the test set is 447 nm. The first column of Figure 6.9 shows the results for an ensemble of 8 neural networks in dependence of the amount of training data. The second column refers to an increasing number of ensemble members trained on about 4000 data samples.

In summary, our results imply that the uncertainty quantification of the inferred regression function is sufficiently reliable using classical deep ensembles for the inverse form reconstruction problem associated with the TWI. The approximation of the posterior of the model parameters by a mixture of delta distributions captures the epistemic uncertainty well enough for the amount of available training data used here. However, this application can benefit from the proposed extension to a mixture of normal distributions when very few training data are used or when only a single deep neural network is trained, cf. Figures 6.9 and 6.10.

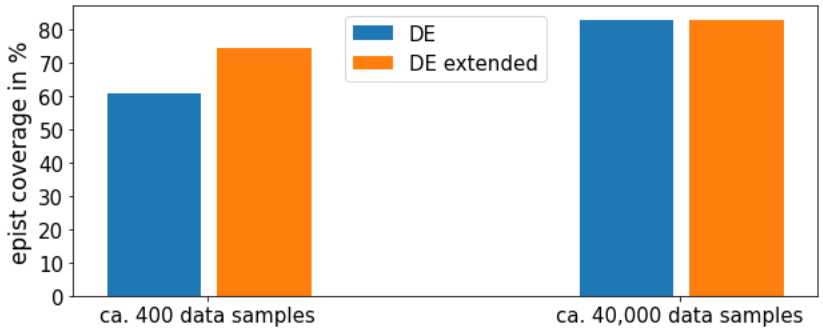

Figure 6.10: Comparison between the coverage probabilities of the epistemic uncertainties of the classical deep ensembles (blue) and the proposed extension (orange) for few vs many data.

6.3 Discussion and summary

Deep ensembles (Lakshminarayanan et al., 2017) represent the state-of-the-art for the uncertainty quantification in deep learning. However, they can significantly underrate the epistemic uncertainty, as the numerical experiments have shown. We provide a remedy by developing an extension to deep ensembles that improves its uncertainty quantification in a post-processing step without losing any of the advantages of the approach, such as its prediction accuracy, and without altering its training procedure or inferred regression function. Numerical experiments on both simulated and real data illustrate the effect of the extended approach, leading to a more appropriate uncertainty quantification for the inferred regression function than classical deep ensembles. The observed effects are especially significant when limited training data are available or the neural network ensemble consists of few members, e.g. due to limited computational capacities. Moreover, the introduced method can also be employed to obtain an epistemic uncertainty for a single trained neural network.

The proposed approach is based on viewing deep ensembles from a Bayesian perspective. More precisely, the delta distributions that serve as an approximation of the posterior for the neural network parameters in the conventional deep ensembles are replaced by normal distributions with finite variances. The corresponding variances can be computed in a post-processing step by minimizing the Kullback-Leibler divergence between the approximate posterior and the true posterior. Analytical calculations show that the proposed extension

of deep ensembles yields an additional term to the covariance matrix of the estimated regression function that results in an enlarged epistemic uncertainty. The presented theoretical formulations in this chapter range from explicitly deriving deep ensembles as an approximation of a conventional Bayesian inference by stating the underlying approximations and assumptions, including the required statistical model and choice of prior, to new formulations related to the developed extension.

In order to apply the proposed extension to the inverse problem in the context of the TWI, we choose the model architecture and loss function to be different from the previous chapters. In particular, a simpler neural network architecture is chosen here with a fully connected last layer (without bias), instead of the previously used U-Net having a convolutional layer as the last layer. In contrast to directly predicting the height values of the seeked topography, Zernike polynomials are used in this chapter to parameterize the estimated topographies and the neural networks predict the Zernike coefficients. Thus, an additional linear transformation is performed to the output of the neural networks and the posterior of the neural network parameters can be analytically approximated around the learned (local) MAPs. The results are similar to the results from both the toy regression problems and the real data. Namely, that the improved uncertainty estimation of the proposed extension is most significant when limited training data are available or only few neural networks are trained, for instance due to limited computational capacities. It can be concluded that approximating the posterior of the model parameters by delta distributions is sufficient for the TWI application when enough training data and ensemble members are available.

The epistemic uncertainty is the crucial part of the uncertainty when the goal is to infer the underlying regression function. Therefore, it should be given more relevance in future work, e.g. when dealing with the black-box character of deep neural networks. Another observation is that the prediction capacity as a function of the number of trained neural networks converged in all performed experiments for less than 10 ensemble members.

The basis of the proposed extension is similar to ideas that have already been put forward in related work (Maddox et al., 2019; Wilson and Izmailov, 2020). In contrast to those approaches, however, the extension of deep ensembles presented here does not alter the actual method, but simply adds a post-processing step. Future work could address the adaptation to classification problems, replacing the linear unit of the neural network output with a convolution layer for imaging tasks, and relaxing the uniform weights $1/K$ of the Gaussian mixture distributions. Combining the proposed extension with other methods that improve the accuracy of deep ensembles (Wenzel et al., 2020) or their extrapolation power (Pearce et al., 2020) could also be of interest.

7

Conclusion and outlook

The goal of this thesis was to explore the potential benefit of data-driven deep learning techniques for solving the high-dimensional nonlinear inverse problem associated to the form reconstruction problem of the TWI, which has so far been treated using conventional data analysis methods. The TWI is a state-of-the-art method for the accurate form measurement of optical aspheres and freeform surfaces. Deep learning algorithms have been successfully applied in a wide range of applications in science and industry, including inverse problems and optical imaging, thanks to their ability to learn complex relationships from given data. Potential benefits for the TWI application might be shorter calculation times, since neural networks produce instant results once they are trained, improved accuracies of the measured surface, as conventional methods use local linearizations for the reconstruction procedure, or a quantification of the uncertainty of the reconstructed topography to obtain more reliable results.

Five objectives were defined at the beginning of this work and will be discussed subsequently. The first objective concerns the generation of a database to train, test and compare deep learning approaches for solving the inverse problem related to the TWI. To this end, a virtual TWI is introduced in Chapter 3, which is used as numerical forward model for simulating experimental TWI data. We explain how to generate a database for the neural network training via virtual experiments and how to simulate out-of-distribution data for the model validation by systematically modifying the virtual TWI used to generate the test data. In this work, the data are generated for two different design topographies (a two-radii and an aspherical surface) at fixed positions of the specimens in the measurement setup.

Chapter 4 deals with the second and third objectives that refer to the actual development of a deep learning approach for the inverse problem associated with the form reconstruction problem of the TWI incorporating at the same time an uncertainty quantification of the prediction. A first proof-of-principle study shows that the inverse problem related to the TWI can be solved using deep neural networks with an achieved accuracy of about 20 nm median absolute error (compared to a 470 nm median absolute deviation of the test set). Based on a U-Net architecture, the pixelwise height-values of the topography are predicted directly. It is difficult to compare these results to the classical TWI evaluation, because there does not

exist a unique TWI but several implementations from different research sites and companies, and various possible hyperparameter settings. For instance, the TWI at PTB is continuously adapted and further improved as part of ongoing research. Nonetheless, we conclude from our experience that the conventional TWI outperforms the developed deep learning approach in terms of accuracy for perfectly simulated data, while the results of both methods approach each other when the numerical model is disturbed.The deep learning approach turns out to be beneficial by reducing the calculation time of the evaluation procedure by more than an order of magnitude, since predictions are provided instantly once the model is trained. Moreover, it appears robust towards noisy input data. Our findings also show that the achieved accuracy highly depends on the number of available training data and that an ensemble of trained neural networks provides higher accuracies than a single model.

To incorporate an uncertainty quantification of the prediction into the deep learning approach, we developed an ensembling technique based on deep ensembles. Deep ensembles scale well to high-dimensional data, outperform other modern uncertainty methods in recent comparison studies, and are straightforward to implement. In general, the uncertainty quantification in deep learning is composed of an irreducible aleatoric part related to noisy data and a reducible epistemic part. Since the goal is to infer the underlying regression function when reconstructing the specimen topographies, we focus on the epistemic part of the uncertainty. Furthermore, the results show that the uncertainty is mainly of epistemic nature, because disturbing the input data by white noise has almost no effect on the predictions. Noisy input data can be considered to be the main source of aleatoric uncertainty here, since the problem at hand is inverse and the noisy measurement data constitute the input to the deep learning model. Our findings show that the proposed deep neural network ensemble provides reliable predictions and a calibrated uncertainty quantification of the prediction not only for in-domain but also for out-of-distribution data simulated by a systematically disturbed numerical forward model generating the test data. However, the epistemic uncertainty quantification by deep ensembles can significantly underrate the uncertainty when less training data are available.

Chapter 5 tackles the fourth objective, i.e. to eventually apply and validate the developed deep learning approach on measurement data from a real specimen. For real data the ground truth is unknown. Therefore, we compare our results to the results obtained by a state-of-the-art TWI measurement system from industry. The reconstructed topographies are quite similar, and differences lie well within the uncertainty quantification of the solution of the proposed deep learning method. This shows (in the test cases considered here) that the developed deep learning approach generalizes well from simulated to measurement data.

The success of deep ensembles in a variety of different applications, such as the inverse problem related to the TWI considered in this thesis, motivates the last objective, namely to investigate and improve its uncertainty quantification for regression problems. Deep ensembles were originally introduced as a heuristic method, but arguments towards a Bayesian footing have been put forward as well, albeit more recently. We investigate and further develop deep ensembles from a Bayesian perspective in Chapter 6. Starting with a theoretical analysis, we specify the statistical model, the prior distribution and the type of approximation for the posterior of the model parameters required to understand deep ensembles as an approximate Bayesian method. These considerations lead to new insights into the role of deep ensembles

and help us to develop an extension of deep ensembles, in which an improved approximation of the posterior of the model parameters is achieved by expanding a mixture of delta distributions to normal distributions. The additional variances related to the spread of the neural network parameters around the (local) MAPs can be computed in a post-processing step by maximizing the ELBO, which is a popular approach from variational inference to minimize the distance between two distributions, without altering the training and the prediction accuracy of deep ensembles. If only the weights from the last layer are considered to be random and the last layer is linear, the corresponding variances can be calculated analytically and the variances of the network parameters tend to zero for an infinite amount of training data, cf. Theorem 1. Including the spread of the random neural network parameters around the (local) MAPs leads to an enlarged, and more realistic part of the epistemic uncertainty, cf. Theorem 2.

In addition to providing the theoretical formulas and analytical derivations, we perform numerical experiments to analyze the impact of the developed extension to deep ensembles. Our findings show that conventional deep ensembles sometimes significantly underestimate the epistemic part of the uncertainty, where our proposed method can provide a remedy. The highest impact of the proposed extension arises when limited data are available or only very few ensemble members are trained, e.g. due to limited computational capacities. Also, the proposed extension already provides an epistemic uncertainty if only a single network is trained, which is non-existent in conventional deep ensembles. Some general conclusions of the experiments performed in this thesis are that the aleatoric part of the uncertainty often dominates the total uncertainty of the prediction and we recommend future research to put more emphasis on the epistemic part of the uncertainty, which is the relevant part of the uncertainty when the goal is to infer a regression function.

To apply the proposed extension of deep ensembles to the inverse problem related to the TWI, we choose a different neural network architecture which is simpler and, in particular, has a fully connected linear last layer instead of a convolutional layer to ensure analytical computations. The results suggest that using a mixture of delta distributions for the approximation of the posterior of the model parameters is sufficiently accurate, provided that enough training data are available. However, the proposed extension should be applied instead when limited data are available or when the ensemble consists of less than five neural networks. A general remark is that in the experiments conducted here, an ensemble of less than ten neural networks was usually sufficient to exploit the potential of the prediction accuracy and uncertainty estimation of deep ensembles.

Limitations of the work presented here are discussed in the following. The proposed Bayesian treatment of deep ensembles is a variational inference method without knowing how good the approximation to the real posterior really is. Considering the deep learning approach developed to solve the inverse problem related to the TWI, the model needs to be retrained anytime the design topography or the measurement position changes. The reconstruction accuracies do not improve compared to conventional methods on perfectly simulated data. Also, the presented results do not have the full resolution provided by the conventional TWI in order to save memory, computational capacities and to fasten the data generation process. It should be no problem to adapt the resolution in future work. Another limitation from the metrological perspective is that the numerical forward model employed to generate the

training data is not perfect. Systematic inconsistencies in the training data as well as further neglected error sources of the measurement process might not be considered in the data-driven uncertainty quantification of the reconstructed surface form. More research needs to be carried out to provide a full measurement uncertainty.

In summary, the application and development of deep learning methods to the high-dimensional inverse problem from computational optical form measurement related to the form reconstruction problem of the TWI was explored for the first time in this thesis. The developed deep learning approach based on deep ensembles might be beneficial for mass production, where many specimens of a fixed design topography are fabricated, thanks to shorter calculation times and the included instant uncertainty quantification of the predicted surface form.

Next steps could be to extend the database to more design topographies and include more error sources, such as misalignment errors or high-frequency errors. Future work could also include the deep learning approach in a larger comparison study. The trustworthiness of the developed approach might be further improved using explainability methods. It would be desirable to release a public database for the machine learning community in the near future to compare, for instance, scalable uncertainty methods on a high-dimensional benchmark dataset from optical form measurement based on a real-world application. Training the deep neural networks might be further improved by investigating techniques such as curriculum learning (Bengio et al., 2009; Weinshall et al., 2018) for faster convergence and better generalization ability, active learning (Ren et al., 2021) for less required training data and faster convergence, or pruning (LeCun et al., 1989a; Yeom et al., 2021) to eliminate irrelevant units of the deep neural network resulting in smaller architectures.

Future work should also include a thorough comparison of uncertainty methods in deep learning with respect to the epistemic uncertainty for deep regression tasks. Furthermore, it would be of interest to expand the proposed extension of deep ensembles to a convolutional last layer, or to reconsider the uniform weighting of the mixture distributions, or to investigate if a similar approach could incorporate an epistemic uncertainty that is applicable in a meaningful way also to inverse problems with a non-unique solution. Another idea derived from the developed extension could be to compute the variances of the model parameters via a Laplace approximation instead of maximizing the ELBO.

A

Supplementary material

A.1 Aspherical specimen

The asphere, used as the design topography for the data generation as well as for the application to measurement data, has the following design parameters. The aspherical coefficients are $A_4 = 5.4145 * 10^3$ m^{-3}, $A_6 = -8.0413 * 10^5$ m^{-5}, $A_8 = -2.9871 * 10^9$ m^{-7}, $A_{10} = -1.4918 * 10^{12}$ m^{-9}, $A_{12} = 1.3777 * 10^{15}$ m^{-11}, $A_{14} = 4.4258 * 10^{18}$ m^{-13}, and $A_{16} = -3.4928 * 10^{21}$ m^{-15}. The aspherical equation is given in equation (2.1). The conic constant and the paraxial surface radius characterising the asphere are $\kappa = -1$, and $R = 0.0202$ m.

The asphere is plotted in Figure A.1a. Figure A.1b shows the deviation of the asphere to a sphere, i.e. the difference between the topography of the asphere and its BFS.

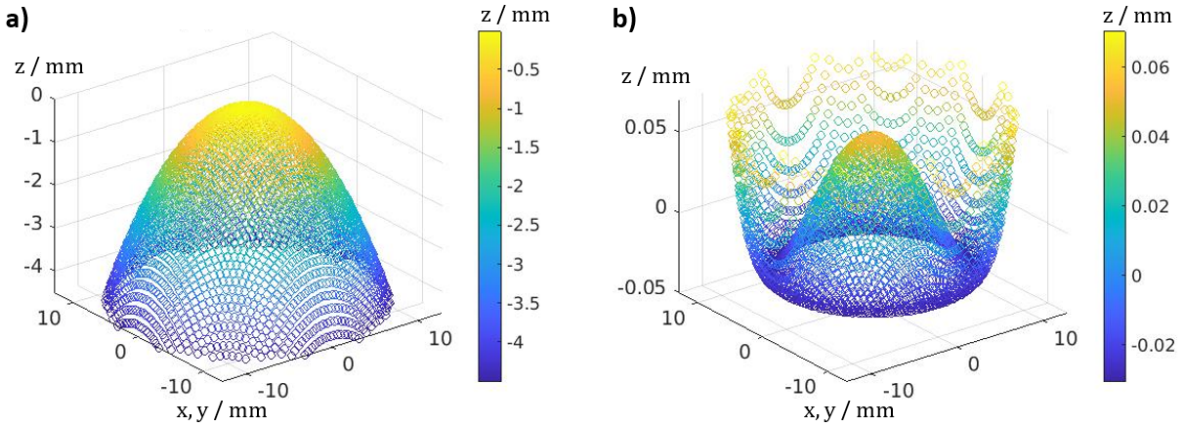

Figure A.1: a) Topography of the aspherical specimen. **b)** Deviation of the asphere to its BFS.

A.2 Additional plots

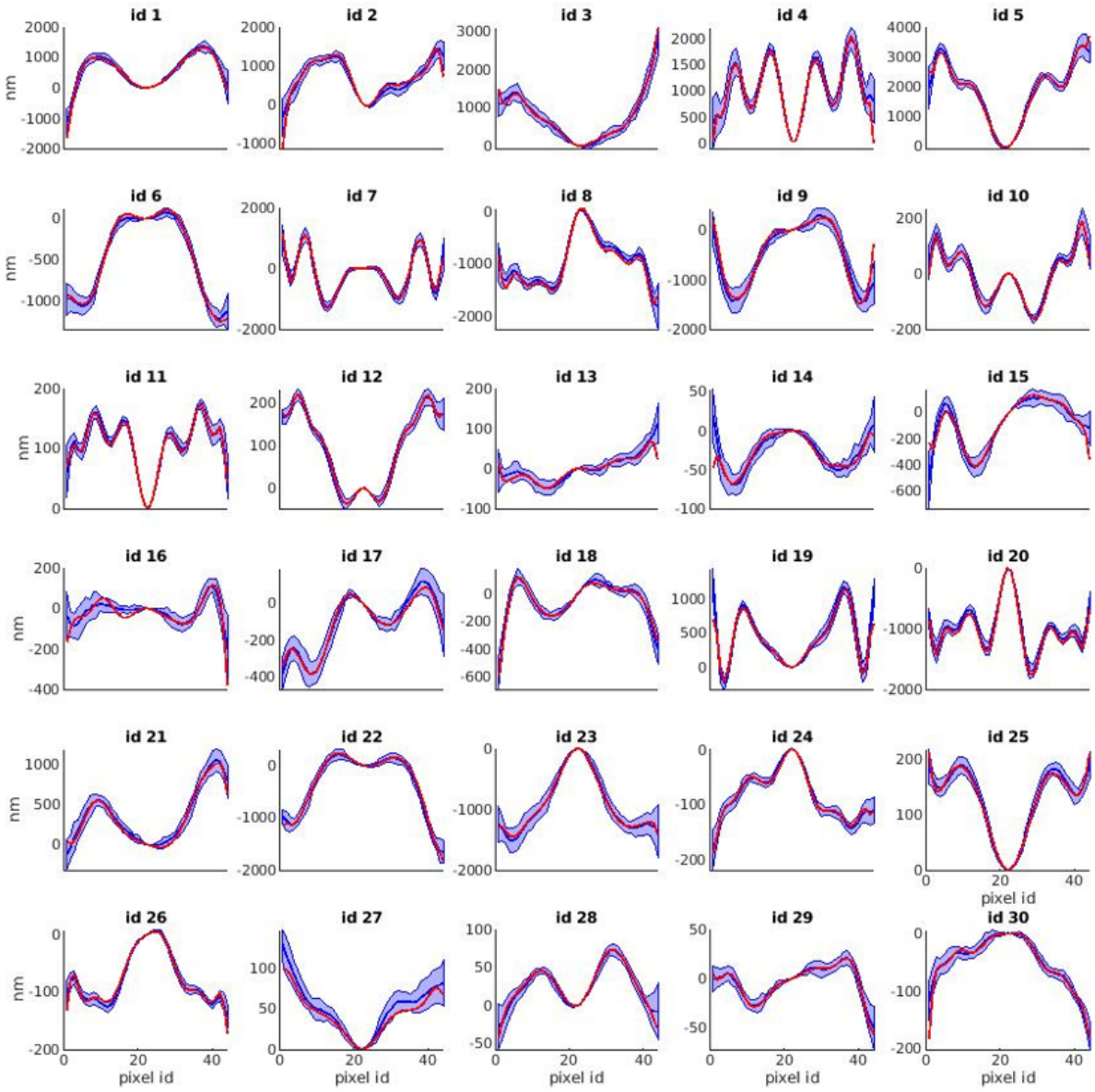

Figure A.2: The profile plots of all difference topographies from Figure 4.4 are plotted in red, while the ensemble predictions and the estimated uncertainties are plotted in blue for the perfectly calibrated system (cf. Subsection 4.2.3). The plots are taken from Hoffmann et al. (2021b).

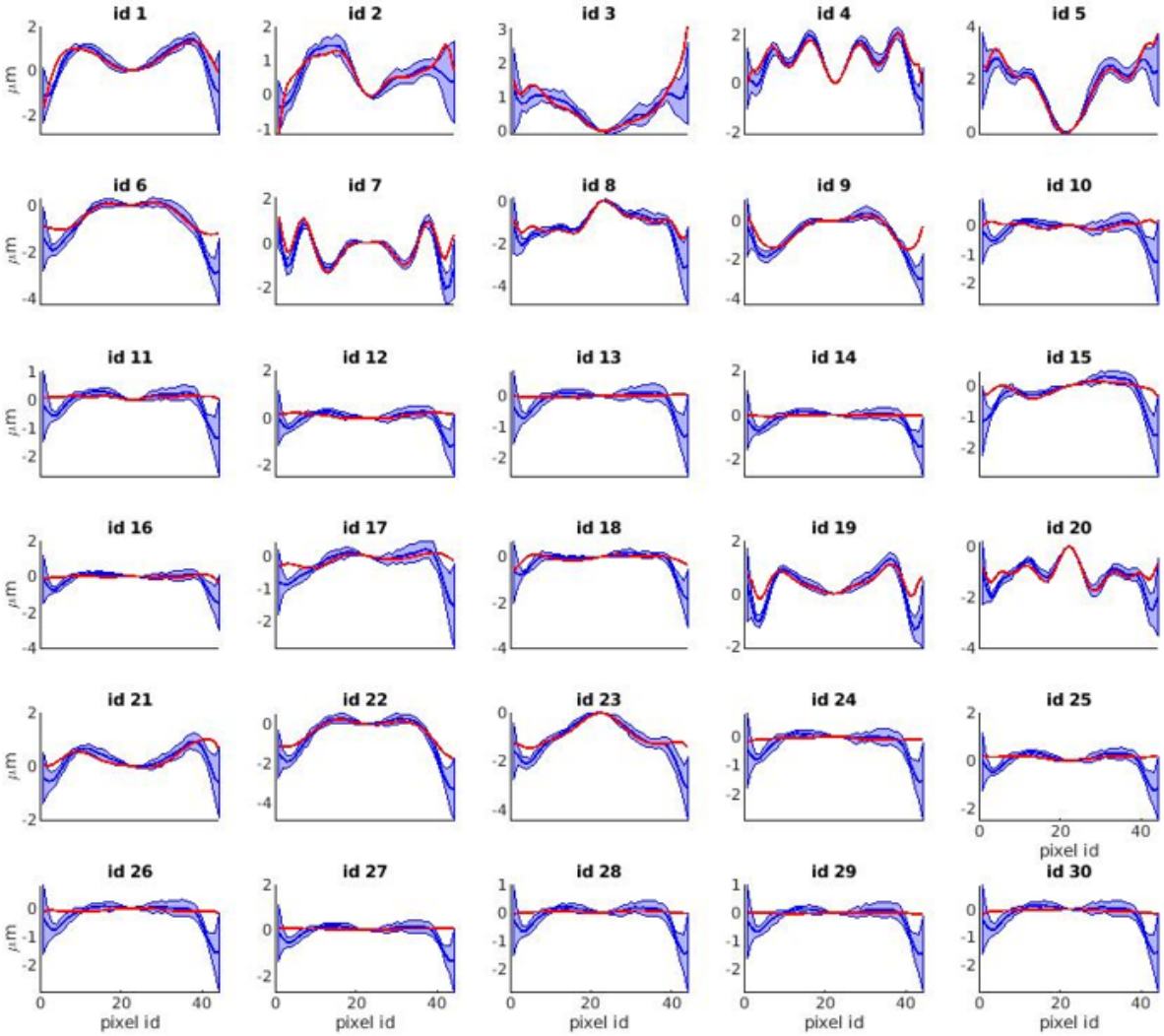

Figure A.3: The profile plots of all difference topographies from Figure 4.4 are plotted in red, while the ensemble predictions and the estimated uncertainties are plotted in blue for the full calibration error (cf. Subsection 4.2.3). The plots are taken from Hoffmann et al. (2021b).

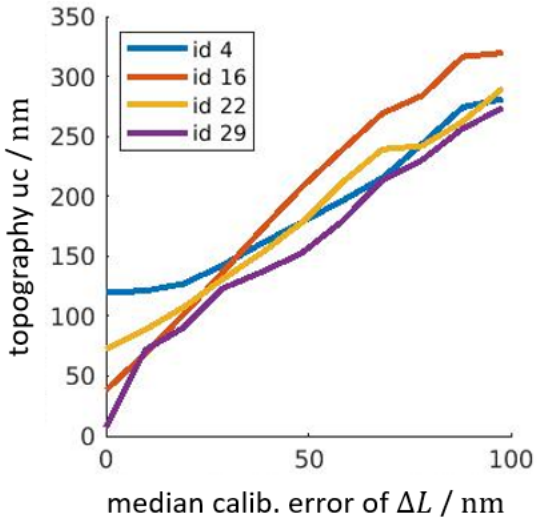

Figure A.4: The topography uncertainty is plotted against the increasing calibration error for four individual difference topographies (cf. Figures 4.4, 4.6a and 4.7a). The figure is taken from Hoffmann et al. (2021b).

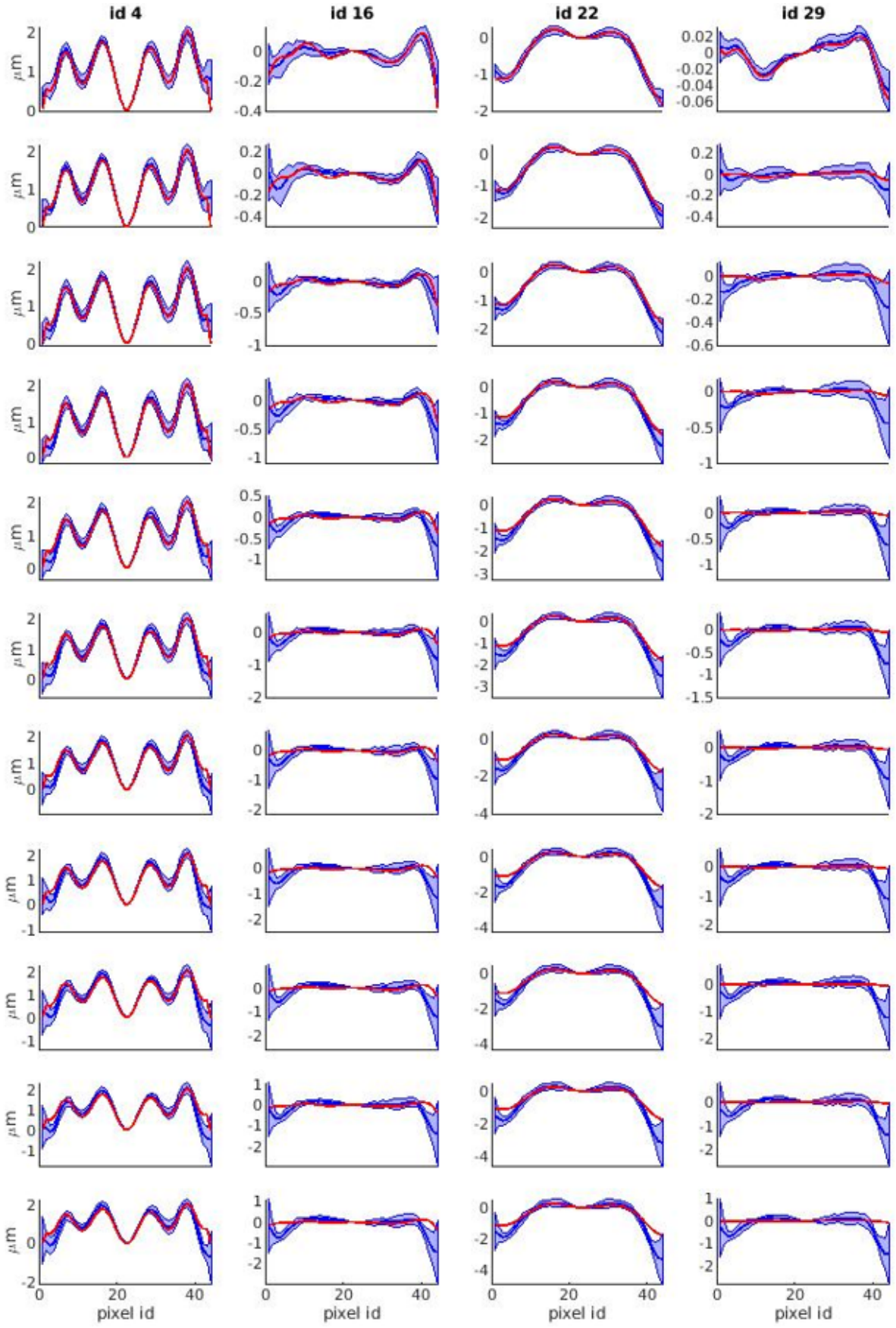

Figure A.5: The profiles of the ensemble prediction and its estimated uncertainty tube (blue) are plotted together with the ground truth (red) for four difference topographies. The first row shows the results for the perfectly calibrated system. The calibration error is then increased by 10% in each row (cf. Figures 4.4, 4.6a, 4.7a and A.4). The plots are taken from Hoffmann et al. (2021b).

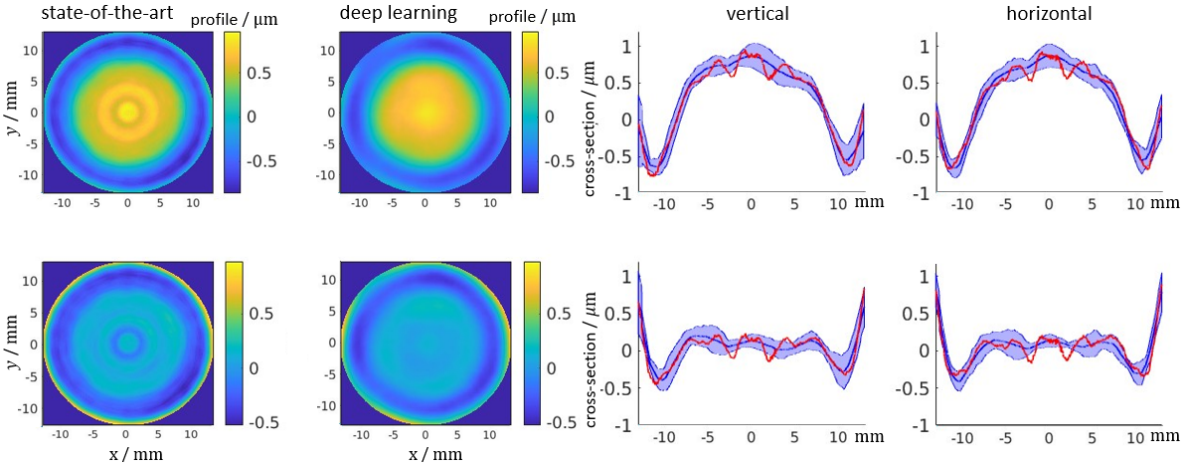

Figure A.6: Comparison of the measurement results between the state-of-the-art method and the proposed deep learning hybrid approach discussed in Chapter 5 with a resolution of 1000×1000 pixels. The topographies are linearly interpolated on the equidistant grid from their respective pointclouds. Here, the high-frequency parts are included as well and no smoothing occurs through fitted Zernike polynomials. A misalignment correction is carried out prior to the deep learning prediction. The first row shows the reconstructed difference topographies, while the second row shows the results after subtracting the corresponding best-fit-spheres. The profiles are plotted together with their vertical and horizontal cross-sections, where the cross-section of the predicted topography of the deep learning hybrid approach is plotted with its 95% credible interval in blue and the red curve corresponds to the state-of-the-art method.

B

Supplementary formulas and derivations

B.1 Mean and covariance matrix of the posterior predictive distribution for the original deep ensembles approach

The first (6.6) and central second (6.7) moments of the approximation (6.5) of the posterior predictive distribution are calculated as follows:

$$
\begin{aligned}
E\left(y|x,D\right) &= \int y\pi(y|x,D)\mathrm{d}y \\
&\approx \frac{1}{K}\sum_{k=1}^{K}\int yN\left(y;\eta_{\widehat{\theta}^{(k)}}(x),\sigma^2_{\widehat{\theta}^{(k)}}(x)\mathrm{I}\right)\mathrm{d}y \\
&= \frac{1}{K}\sum_{k=1}^{K}\eta_{\widehat{\theta}^{(k)}}(x)\,, \\
Cov\left(y|x,D\right) &= \int\left[\left(y-E\left(y|x,D\right)\right)\left(y-E\left(y|x,D\right)\right)^{\top}\right]\pi(y|x,D)\mathrm{d}y \\
&\approx \frac{1}{K}\sum_{k=1}^{K}\int[\left(\eta_{\widehat{\theta}^{(k)}}(x)-E\left(y|x,D\right)\right)\left(\eta_{\widehat{\theta}^{(k)}}(x)-E\left(y|x,D\right)\right)^{\top} \\
&+ \left(y-\eta_{\widehat{\theta}^{(k)}}(x)\right)\left(\eta_{\widehat{\theta}^{(k)}}(x)-E\left(y|x,D\right)\right)^{\top} \\
&+ \left(\eta_{\widehat{\theta}^{(k)}}(x)-E\left(y|x,D\right)\right)\left(y-\eta_{\widehat{\theta}^{(k)}}(x)\right)^{\top} \\
&+ \left(y-\eta_{\widehat{\theta}^{(k)}}(x)\right)\left(y-\eta_{\widehat{\theta}^{(k)}}(x)\right)^{\top}]\;N\left(y;\eta_{\widehat{\theta}^{(k)}}(x),\sigma^2_{\widehat{\theta}^{(k)}}(x)\mathrm{I}\right)\mathrm{d}y \\
&= \frac{1}{K}\sum_{k=1}^{K}\left[\left(\eta_{\widehat{\theta}^{(k)}}(x)-E\left(y|x,D\right)\right)\left(\eta_{\widehat{\theta}^{(k)}}(x)-E\left(y|x,D\right)\right)^{\top}+\sigma^2_{\widehat{\theta}^{(k)}}(x)\mathrm{I}\right].
\end{aligned}
$$

B.2 Proof of Lemma 1

Application of the change-of-variables formula and the approximation (6.2) of the posterior $\pi(\theta|D)$ yield the following for the regression function $\eta \equiv \eta_\theta(x)$:

$$\begin{aligned}
\pi(\eta|x,D) &= \int \delta(\eta - \eta_\theta(x))\pi(\theta|D)\mathrm{d}\theta \\
&\approx \int \delta(\eta - \eta_\theta(x))\frac{1}{K}\sum_{k=1}^{K}\delta(\theta - \widehat{\theta}^{(k)})\mathrm{d}\theta \\
&= \frac{1}{K}\sum_{k=1}^{K}\delta(\eta - \eta_{\widehat{\theta}^{(k)}}(x))\,.
\end{aligned}$$

The expectation and covariance matrix then follow immediately:

$$\begin{aligned}
E\left(\eta|x,D\right) &= \int \eta\pi(\eta|x,D)\mathrm{d}\eta \\
&\approx \frac{1}{K}\sum_{k=1}^{K}\int \eta\delta(\eta - \eta_{\widehat{\theta}^{(k)}}(x))\mathrm{d}\eta \\
&= \frac{1}{K}\sum_{k=1}^{K}\eta_{\widehat{\theta}^{(k)}}(x)\,,
\end{aligned}$$

$$\begin{aligned}
Cov\left(\eta|x,D\right) &= \int \left[\left(\eta - E\left(\eta|x,D\right)\right)\left(\eta - E\left(\eta|x,D\right)\right)^\top\right]\pi(\eta|x,D)\mathrm{d}\eta \\
&\approx \frac{1}{K}\sum_{k=1}^{K}\int [\left(\eta_{\widehat{\theta}^{(k)}}(x) - E\left(\eta|x,D\right)\right)\left(\eta_{\widehat{\theta}^{(k)}}(x) - E\left(\eta|x,D\right)\right)^\top \\
&+ \left(\eta - \eta_{\widehat{\theta}^{(k)}}(x)\right)\left(\eta_{\widehat{\theta}^{(k)}}(x) - E\left(\eta|x,D\right)\right)^\top \\
&+ \left(\eta_{\widehat{\theta}^{(k)}}(x) - E\left(\eta|x,D\right)\right)\left(\eta - \eta_{\widehat{\theta}^{(k)}}(x)\right)^\top \\
&+ \left(\eta - \eta_{\widehat{\theta}^{(k)}}(x)\right)\left(\eta - \eta_{\widehat{\theta}^{(k)}}(x)\right)^\top] \,\delta(\eta - \eta_{\widehat{\theta}^{(k)}}(x))\mathrm{d}\eta \\
&= \frac{1}{K}\sum_{k=1}^{K}\left[\left(\eta_{\widehat{\theta}^{(k)}}(x) - E\left(\eta|x,D\right)\right)\left(\eta_{\widehat{\theta}^{(k)}}(x) - E\left(\eta|x,D\right)\right)^\top\right]. \quad \square
\end{aligned}$$

B.3 Proof of Theorem 1

The objective function to be maximized is the ELBO from equation (6.12):

$$ELBO = E_q \log p(D|\theta) - KL\left(q(\theta)\|\pi(\theta)\right)\,.$$

The first term of equation (6.12) is the expectation of the log likelihood (up to a constant):

$$E_q \log p(D|\theta) = -\frac{1}{2K}\sum_{k=1}^{K}\sum_{i=1}^{N}\int N\left(\theta;\widehat{\theta}^{(k)},\gamma_k \mathrm{I}\right)\left(\frac{\|y_i - \eta_\theta(x_i)\|^2}{\sigma^2_{\hat{\theta}^{(k)}}(x_i)} + d_y \log\left(\sigma^2_{\hat{\theta}^{(k)}}(x_i)\right)\right)\mathrm{d}\theta, \tag{B.1}$$

where $\|.\|$ refers to the L^2-norm. Note that the parameters of the last layer related to the variances $\sigma^2_{\hat{\theta}^{(k)}}$ of the statistical model (6.1) are fixed after the network training. The parameters θ of the corresponding mean η_θ are drawn only from the last layer to the output without any nonlinearity. Hence, the network output of the kth ensemble member can be written as

$$\eta_\theta(x) = W_\theta \, \widehat{\eta}^{(k)}(x), \; W_\theta = W_{\hat{\theta}^{(k)}} + \sqrt{\gamma_k}\mathcal{E} \; \in \mathbb{R}^{d_y \times d_{\hat{\eta}}}, \; \mathcal{E}_{i,j} \sim N(0,1), i = 1, \ldots, d_y, j = 1, \ldots, d_{\hat{\eta}}, \tag{B.2}$$

where $\widehat{\eta}^{(k)}(x)$ is the output of the next-to-last layer of the trained network. The dependence of $\widehat{\eta}^{(k)}(x)$ on θ is suppressed, assuming that the corresponding MAP estimates for θ are taken. W_θ contains the parameters of the linear unit drawn from a normal distribution around the MAPs $W_{\hat{\theta}^{(k)}}$ with the same elementwise variance γ_k for each parameter. $\mathcal{E}$ is a matrix with the same dimensionality as $W_{\hat{\theta}^{(k)}}$, i.e. the output dimension d_y times the dimension of the last layer $d_{\hat{\eta}}$.

Using (B.2), the following equations hold:

$$\begin{aligned}
\int N\left(\theta; \widehat{\theta}^{(k)}, \gamma_k \mathrm{I}\right) \|y - \eta_\theta(x)\|^2 \mathrm{d}\theta &= E_{\mathcal{E}}\left[\|y - (W_{\hat{\theta}^{(k)}} + \sqrt{\gamma_k}\mathcal{E})\hat{\eta}^{(k)}(x)\|^2\right] && \text{(B.3)} \\
&= E_{\mathcal{E}}\left[\|y - W_{\hat{\theta}^{(k)}}\hat{\eta}^{(k)}(x)\|^2 - 2(y - W_{\hat{\theta}^{(k)}}\hat{\eta}^{(k)}(x))^\top(\sqrt{\gamma_k}\mathcal{E}\hat{\eta}^{(k)}(x)) + \|\sqrt{\gamma_k}\mathcal{E}\hat{\eta}^{(k)}(x)\|^2\right] \\
&= \|y - \eta_{\hat{\theta}^{(k)}}(x)\|^2 + \gamma_k E_{\mathcal{E}}\left[\|\mathcal{E}\hat{\eta}^{(k)}(x)\|^2\right], && \text{(B.4)}
\end{aligned}$$

where $\eta_{\hat{\theta}^{(k)}}$ is the trained network with MAP estimates. Furthermore, the following holds:

$$\begin{aligned}
E_{\mathcal{E}}\left[\|\mathcal{E}\widehat{\eta}^{(k)}(x)\|^2\right] &= E_{\mathcal{E}}\left[(\mathcal{E}\widehat{\eta}^{(k)}(x))^\top(\mathcal{E}\widehat{\eta}^{(k)}(x))\right] && \text{(B.5)} \\
&= \sum_{r=1}^{d_y}\sum_{\alpha,\beta=1}^{d_{\hat{\eta}}} \widehat{\eta}^{(k)}(x)_\alpha E\left[\mathcal{E}_{r,\alpha}\mathcal{E}_{r,\beta}\right]\widehat{\eta}^{(k)}(x)_\beta && \text{(B.6)} \\
&= \sum_{r=1}^{d_y}\sum_{\alpha=1}^{d_{\hat{\eta}}} \widehat{\eta}^{(k)}(x)_\alpha E\left[\mathcal{E}^2_{r,\alpha}\right]\widehat{\eta}^{(k)}(x)_\alpha && \text{(B.7)} \\
&= \sum_{r=1}^{d_y}\sum_{\alpha=1}^{d_{\hat{\eta}}} \widehat{\eta}^{(k)}(x)_\alpha \left(\mathrm{Var}(\mathcal{E}_{r,\alpha}) + E\left[\mathcal{E}_{r,\alpha}\right]^2\right)\widehat{\eta}^{(k)}(x)_\alpha && \text{(B.8)} \\
&= d_y\|\widehat{\eta}^{(k)}(x)\|^2. && \text{(B.9)}
\end{aligned}$$

Inserting (B.4) and (B.9) into equation (B.1) yields:

$$E_q \log p(D|\theta) = -\frac{1}{2K}\sum_{k=1}^{K}\sum_{i=1}^{N}\left(\frac{\|y_i - \eta_{\hat{\theta}^{(k)}}(x)\|^2 + \gamma_k d_y\|\widehat{\eta}^{(k)}(x)\|^2}{\sigma^2_{\hat{\theta}^{(k)}}(x_i)} + d_y \log\left(\sigma^2_{\hat{\theta}^{(k)}}(x_i)\right)\right). \tag{B.10}$$

The second term of the ELBO (6.12) is the Kullback-Leibler divergence between the approximation of the posterior $q(\theta)$ and the chosen prior $\pi(\theta)$. The following holds:

$$\begin{aligned} KL\left(q(\theta)\|\pi(\theta)\right) &= \int q(\theta)\log\left(\frac{q(\theta)}{\pi(\theta)}\right)\mathrm{d}\theta && \text{(B.11)}\\ &= \int \frac{1}{K}\sum_{k=1}^{K} N\left(\theta;\widehat{\theta}^{(k)},\gamma_k\mathrm{I}\right)\log\left(\frac{\frac{1}{K}\sum_{r=1}^{K} N\left(\theta;\widehat{\theta}^{(r)},\gamma_r\mathrm{I}\right)}{\pi(\theta)}\right)\mathrm{d}\theta && \text{(B.12)}\\ &\approx \frac{1}{K}\sum_{k=1}^{K}\int N\left(\theta;\widehat{\theta}^{(k)},\gamma_k\mathrm{I}\right)\log\left(\frac{\frac{1}{K}N\left(\theta;\widehat{\theta}^{(k)},\gamma_k\mathrm{I}\right)}{\pi(\theta)}\right)\mathrm{d}\theta && \text{(B.13)}\\ &= \frac{1}{K}\sum_{k=1}^{K} KL\left(N(\theta;\widehat{\theta}^{(k)},\gamma_k\mathrm{I})\|\pi(\theta)\right) - \frac{1}{K}\sum_{k=1}^{K}\log(K)\,. && \text{(B.14)} \end{aligned}$$

The approximation

$$\int N\left(\theta;\widehat{\theta}^{(k)},\gamma_k\mathrm{I}\right)\log\left(\sum_{r=1}^{K} N\left(\theta;\widehat{\theta}^{(r)},\gamma_r\mathrm{I}\right)\right)\mathrm{d}\theta \approx \int N\left(\theta;\widehat{\theta}^{(k)},\gamma_k\mathrm{I}\right)\log\left(N\left(\theta;\widehat{\theta}^{(k)},\gamma_k\mathrm{I}\right)\right)\mathrm{d}\theta$$

is valid under the assumption that the between-variability of the MAP estimates is large compared to the within-variability, since then the first term of the integrand (the normal distribution around $\widehat{\theta}^{(k)}$) goes sufficiently fast to zero for distant $\widehat{\theta}^{(r)}, r \neq k$.

It is well known that the Kullback-Leibler divergence between two Gaussians can be calculated analytically. It follows that:

$$\begin{aligned} KL\left(N(\theta;\widehat{\theta}^{(k)},\gamma_k\mathrm{I})\|\pi(\theta)\right) &= KL\left(N(\theta;\widehat{\theta}^{(k)},\gamma_k\mathrm{I})\|N(\theta;0,\lambda^{-1}I)\right) && \text{(B.15)}\\ &= \frac{1}{2}\left(d_\theta\gamma_k\lambda + \lambda\|\widehat{\theta}^{(k)}\|^2 - d_\theta - d_\theta\log(\gamma_k\lambda)\right), && \text{(B.16)} \end{aligned}$$

where $d_\theta := dim(\widehat{\theta}^{(k)}),\ k = 1,\ldots,K$. Here, $d_\theta = d_y d_{\hat{\eta}}$, as only the weights W_θ of the last linear unit of the network are random. It follows with (B.14) that

$$KL\left(q(\theta)\|\pi(\theta)\right) = \frac{1}{2K}\sum_{k=1}^{K}\left(d_y d_{\hat{\eta}}\gamma_k\lambda + \lambda\|W_{\hat{\theta}^{(k)}}\|_F^2 - d_y d_{\hat{\eta}} - d_y d_{\hat{\eta}}\log(\gamma_k\lambda)\right) - \frac{1}{K}\sum_{k=1}^{K}\log(K)\,, \tag{B.17}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

Now, the ELBO is given by combining (B.10) and (B.17):

$$\begin{aligned}
ELBO &= E_q \log p(D|\theta) - KL\left(q(\theta)\|\pi(\theta)\right) \\
&= -\frac{1}{2K}\sum_{k=1}^{K}\sum_{i=1}^{N}\left(\frac{\|y_i - \eta_{\hat{\theta}^{(k)}}(x_i)\|^2 + \gamma_k d_y \|\widehat{\eta}^{(k)}(x_i)\|^2}{\sigma^2_{\hat{\theta}^{(k)}}(x_i)} + d_y \log\left(\sigma^2_{\hat{\theta}^{(k)}}(x_i)\right)\right) \\
&\quad - \left(\frac{1}{2K}\sum_{k=1}^{K}\left(d_y d_{\hat{\eta}}\gamma_k\lambda + \lambda\|W_{\hat{\theta}^{(k)}}\|_F^2 - d_y d_{\hat{\eta}} - d_y d_{\hat{\eta}}\log(\gamma_k\lambda)\right) - \frac{1}{K}\sum_{k=1}^{K}\log(K)\right) \\
&= \frac{1}{K}\sum_{k=1}^{K}[\log(K) - \frac{1}{2}\{\sum_{i=1}^{N}\left(\frac{\|y_i - \eta_{\hat{\theta}^{(k)}}(x_i)\|^2}{\sigma^2_{\hat{\theta}^{(k)}}(x_i)} + d_y \log\left(\sigma^2_{\hat{\theta}^{(k)}}(x_i)\right)\right) + \lambda\|W_{\hat{\theta}^{(k)}}\|_F^2 \\
&\quad - d_y d_{\hat{\eta}} - d_y d_{\hat{\eta}}\log(\lambda)\} - \frac{1}{2}\left(\sum_{i=1}^{N}\frac{d_y\|\widehat{\eta}^{(k)}(x_i)\|^2}{\sigma^2_{\hat{\theta}^{(k)}}(x_i)} + d_y d_{\hat{\eta}}\lambda\right)\gamma_k + \frac{1}{2}d_y d_{\hat{\eta}}\log(\gamma_k)].
\end{aligned} \tag{B.18}$$

In total, the ELBO takes the form:

$$ELBO = \frac{1}{K}\sum_{k=1}^{K}\left(a_k + b_k\gamma_k + c_k\log(\gamma_k)\right), \tag{B.19}$$

where a_k, b_k, c_k are given analytically in dependence on the (local) MAP estimates $\widehat{\theta}^{(k)}, k = 1, \ldots, K$. The maximizer of (B.19) w.r.t. the γ_k is given by

$$\gamma_k = -\frac{c_k}{b_k} = \frac{d_{\hat{\eta}}}{\sum_{i=1}^{N}\frac{\|\widehat{\eta}^{(k)}(x_i)\|^2}{\sigma^2_{\hat{\theta}^{(k)}}(x_i)} + d_{\hat{\eta}}\lambda}, \ k = 1, \ldots, K. \tag{B.20}$$

Note that $b_k < 0$ and $c_k > 0$ holds. The solution for γ_k in equation (B.20) is indeed a maximum, since the ELBO in equation (B.19) goes to minus infinity for the limits of γ_k (i.e. $\gamma_k \to 0$ and $\gamma_k \to \infty$, and (B.20) is the only stationary solution. It can be expected that $\sum_{i=1}^{N}\frac{\|\widehat{\eta}^{(k)}(x_i)\|^2}{\sigma^2_{\hat{\theta}^{(k)}}(x_i)}$ is unbounded as N grows, since the individual networks are trained independently. Under this assumption, it follows that $\gamma_k \to 0$ for $N \to \infty$. □

B.4 Proof of Theorem 2

The proof is straightforward in analogy to the proof of Theorem 1 (cf. Appendix B.2), with the following addition:

$$\int\left(\eta - \eta_{\widehat{\theta}^{(k)}}(x)\right)\left(\eta - \eta_{\widehat{\theta}^{(k)}}(x)\right)^{\top}\ \delta(\eta - \eta_\theta(x))N\left(\theta; \widehat{\theta}^{(k)}, \gamma_k \mathrm{I}\right)\mathrm{d}\theta\mathrm{d}\eta \approx \gamma_k J_k J_k^{\top}, \ k = 1, \ldots, K, \tag{B.21}$$

using the first-order Taylor approximation $\eta_\theta(x) \approx \eta_{\widehat{\theta}^{(k)}}(x) + J_k(\theta - \widehat{\theta}^{(k)})$ of η_θ for θ in the vicinity of $\widehat{\theta}^{(k)}$.

The derivative J_k can be easily calculated based on the output of the next-to-last layer of the trained networks (cf. Appendix B.3 (B.2)) and equation (6.14). □

B.5 Proposed extension applied to the TWI in 6.2.3

Applying the proposed extension to the inverse problem related to the TWI requires an additional linear transformation after the fully connected last layer (cf. Figure 6.7). The adapted formulas are given below.

The following equations are modified when considering the proof of Theorem 1 in Appendix B.3. Adapted formulas (B.2):

$$\eta_\theta(x) = Z\ W_\theta\ \hat{\eta}(x),\ W_\theta = W_{\hat{\theta}} + \sqrt{\gamma}\mathcal{E},\ \mathcal{E}_{i,j} \sim N(0,1), \tag{B.22}$$

with

$$Z \in \mathbb{R}^{d_y \times d_z},\ W_\theta \in \mathbb{R}^{d_z \times d_{\hat{\eta}}},\ \hat{\eta}(x) \in \mathbb{R}^{d_{\hat{\eta}} \times 1},\ i = 1, \ldots, d_z, j = 1, \ldots, d_{\hat{\eta}}, \tag{B.23}$$

where d_z is the number of estimated Zernike coefficients, i.e. the number of employed Zernike polynomial functions, d_y is the output dimension of the resulting topographies understood as a vector, and $d_{\hat{\eta}}$ is the dimension of the next-to-last layer, i.e. $\hat{\eta}(x)$ is the input to the layer estimating the Zernike coefficients. Adapted equations (B.5-B.9):

$$\begin{aligned}
E_\mathcal{E}\left[\|Z\mathcal{E}\hat{\eta}(x)\|^2\right] &= E_\mathcal{E}\left[(Z\mathcal{E}\hat{\eta}(x))^\top (Z\mathcal{E}\hat{\eta}(x))\right] && \text{(B.24)}\\
&= \sum_{r=1}^{d_y}\sum_{i,j=1}^{d_z}\sum_{\alpha,\beta=1}^{d_{\hat{\eta}}} \hat{\eta}(x)_\alpha z_{r,i} E\left[\mathcal{E}_{i,\alpha}\mathcal{E}_{j,\beta}\right] z_{r,j}\hat{\eta}(x)_\beta && \text{(B.25)}\\
&= \sum_{r=1}^{d_y}\sum_{i=1}^{d_z}\sum_{\alpha=1}^{d_{\hat{\eta}}} \hat{\eta}(x)_\alpha z_{r,i} E\left[\mathcal{E}_{i,\alpha}^2\right] z_{r,i}\hat{\eta}(x)_\alpha && \text{(B.26)}\\
&= \sum_{r=1}^{d_y}\sum_{i=1}^{d_z} z_{r,i}^2 \|\hat{\eta}(x)\|^2 . && \text{(B.27)}
\end{aligned}$$

It follows for the updated variances from equation (B.20):

$$\gamma_k = \frac{d_z d_{\hat{\eta}}}{\sum_{i=1}^N \sum_{r=1}^{d_y}\sum_{k=1}^{d_z} z_{r,k}^2 \frac{\|\widehat{\eta}^{(k)}(x_i)\|^2}{\sigma^2_{\hat{\theta}^{(k)}}(x_i)} + d_z d_{\hat{\eta}}\lambda},\ k = 1, \ldots, K . \tag{B.28}$$

In Theorem 2, the additional part of the covariance matrix in equation (6.15) can be computed as follows:

$$\begin{aligned}
\eta_\theta(x) &= Z\ W_\theta\ \hat{\eta}(x) = \sum_{i=1}^{d_z}\sum_{j=1}^{d_{\hat{\eta}}} \hat{\eta}(x)_j W_{\theta_{i,j}} Z^i\ \in \mathbb{R}^{d_y}, && \text{(B.29)}\\
\frac{\mathrm{d}\eta_\theta(x)}{\mathrm{d}W_{\theta_{i,j}}} &= \hat{\eta}(x)_j Z^i\ \in \mathbb{R}^{d_y}, && \text{(B.30)}\\
J &:= \frac{\mathrm{d}\eta_\theta(x)}{\mathrm{d}W_\theta}, && \text{(B.31)}\\
dim(J) &= d_y \times (d_z \times d_{\hat{\eta}}),\ dim(JJ^\top) = d_y \times d_y. && \text{(B.32)}
\end{aligned}$$

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297.

Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160.

Akhtar, N. and Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430.

ANSI Z80.28-2017 (2017). American National Standard for Ophthalmics - Methods for Reporting Optical Aberrations of Eyes. Standard, American National Standards Institute, Inc.

Ashukha, A., Lyzhov, A., Molchanov, D., and Vetrov, D. (2020). Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *Published as a conference paper at ICLR 2020*.

Aster, R. C., Borchers, B., and Thurber, C. H. (2018). *Parameter estimation and inverse problems*. Elsevier.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. (2010). How to explain individual classification decisions. *JMLR*, 11:1803–1831.

Baer, G., Schindler, J., Pruss, C., and Osten, W. (2013a). Correction of misalignment introduced aberration in non-null test measurements of free-form surfaces. *JEOS:RP*, 8(0).

Baer, G., Schindler, J., Pruss, C., Siepmann, J., and Osten, W. (2014a). Calibration of a non-null test interferometer for the measurement of aspheres and free-form surfaces. *Optics express*, 22(25):31200–31211.

Baer, G., Schindler, J., Pruss, C., Siepmann, J., and Osten, W. (2014b). Fast and flexible non-null testing of aspheres and free-form surfaces with the tilted-wave-interferometer. *International Journal of Optomechatronics*, 8(4):242–250.

Baer, G., Schindler, J., Siepmann, J., Pruß, C., Osten, W., and Schulz, M. (2013b). Measurement of aspheres and free-form surfaces in a non-null test interferometer: reconstruction of high-frequency errors. In *Optical Measurement Systems for Industrial Inspection VIII*, volume 8788, page 878818. International Society for Optics and Photonics.

Baer, G. B. (2016). *Ein Beitrag zur Kalibrierung von Nicht-Null-Interferometern zur Vermessung von Asphären und Freiformflächen*. Institut für Technische Optik, Universität Stuttgart.

Baer, G. B., Schindler, J., Pruss, C., and Osten, W. (2014c). Measurement of aspheres and free-form surfaces with the tilted-wave-interferometer. In *Fringe 2013*, pages 87–95. Springer.

Bai, Y., Chen, W., Chen, J., and Guo, W. (2020). Deep learning methods for solving linear inverse problems: Research directions and paradigms. *Signal Processing*, 177:107729.

Balas, V. E., Roy, S. S., Sharma, D., and Samui, P. (2019). *Handbook of deep learning applications*, volume 136. Springer.

Baldi, P. and Sadowski, P. J. (2013). Understanding dropout. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 26, pages 2814–2822. Curran Associates, Inc.

Barbastathis, G., Ozcan, A., and Situ, G. (2019). On the use of deep learning for computational imaging. *Optica*, 6(8):921–943.

Beisswanger, R., Pruss, C., Schober, C., Harsch, A., and Osten, W. (2019). Tilted wave interferometer in common path configuration: challenges and realization. In *Optical Measurement Systems for Industrial Inspection XI*, volume 11056, page 110561G. International Society for Optics and Photonics.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Beluch, W. H., Genewein, T., Nürnberger, A., and Köhler, J. M. (2018). The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Bergmann, R. B., Falldorf, C., Klein, T., and Agour, M. (2016). Computational optical metrology. `https://repositorium.sdum.uminho.pt/bitstream/1822/46930/1/plenaryIcopen2016_mfcosta.pdf`. icOPEN (Keynote Speech).

Bergmans, R., Nieuwenkamp, H., Kok, G., Blobel, G., Nouira, H., Küng, A., Baas, M., Tevoert, M., Baer, G., and Stuerwald, S. (2015). Comparison of asphere measurements by tactile and optical metrological instruments. *Measurement Science and Technology*, 26(10):105004.

Bernal, J., Kushibar, K., Asfaw, D. S., Valverde, S., Oliver, A., Martí, R., and Lladó, X. (2019). Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review. *Artificial Intelligence in Medicine*, 95:64–81.

Bertero, M., Boccacci, P., and De Mol, C. (2021). *Introduction to inverse problems in imaging*. CRC press.

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. (2011). The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

Binder, A., Bach, S., Montavon, G., Müller, K.-R., and Samek, W. (2016). Layer-wise relevance propagation for deep neural network architectures. In Kim, J. K. and Joukov, N., editors, *Information Science and Applications (ICISA) 2016*, pages 913–922, Singapore. Springer Singapore.

BIPM (2004). What is metrology? `https://web.archive.org/web/20110927012931/http://www.bipm.org/en/convention/wmd/2004/#`. Accessed: 2021-12-22.

BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML (2008a). Evaluation of measurement data — Supplement 1 to the "Guide to the expression of uncertainty in measurement" — Propagation of distributions using a Monte Carlo method. Joint Committee for Guides in Metrology, JCGM 101:2008. `https://www.bipm.org/en/committees/jc/jcgm/publications`. Accessed: 2022-06-14.

BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML (2008b). Evaluation of measurement data — Supplement 2 to the "Guide to the expression of uncertainty in measurement" — Extension to any number of output quantities. Joint Committee for Guides in Metrology, JCGM 102:2008. `https://www.bipm.org/en/committees/jc/jcgm/publications`. Accessed: 2022-06-14.

BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML (2008c). Evaluation of measurement data—guide to the expression of uncertainty in measurement. Joint Committee for Guides in Metrology, JCGM 100: 2008. `https://www.bipm.org/en/committees/jc/jcgm/publications`. Accessed: 2022-06-14.

BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML (2020). Guide to the expression of uncertainty in measurement — Part 6: Developing and using measurement models. Joint Committee for Guides in Metrology, JCGM 102:2008. `https://www.bipm.org/en/committees/jc/jcgm/publications`. Accessed: 2022-06-14.

BIPM, I., IFCC, I., IUPAC, I., and ISO, O. (2012). The international vocabulary of metrology—basic and general concepts and associated terms (VIM). *JCGM*, 200:2012.

Bishop, C. M. (1994). Mixture density networks. Technical report, NCRG/94/004.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blalock, T., Medicus, K., and Nelson, J. D. (2015). Fabrication of freeform optics. In Fähnle, O. W., Williamson, R., and Kim, D. W., editors, *Optical Manufacturing and Testing XI*, volume 9575, pages 74 – 83. International Society for Optics and Photonics, SPIE.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1987). Occam's razor. *Information Processing Letters*, 24(6):377–380.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France. PMLR.

Braunecker, B., Hentschel, R., and Tiziani, H. J. (2008). *Advanced optics using aspherical elements*, volume 173. Spie Press.

Brosse, N., Riquelme, C., Martin, A., Gelly, S., and Moulines, É. (2020). On last-layer algorithms for classification: Decoupling representation from uncertainty estimation. *arXiv preprint arXiv:2001.08049*.

Buza, K. (2014). Feedback prediction for blogs. In *Data analysis, machine learning and knowledge discovery*, pages 145–152. Springer.

Caldeira, J. and Nord, B. (2020). Deeply uncertain: comparing methods of uncertainty quantification in deep learning algorithms. *Machine Learning: Science and Technology*, 2(1):015002.

Cao, C., Liu, F., Tan, H., Song, D., Shu, W., Li, W., Zhou, Y., Bo, X., and Xie, Z. (2018). Deep learning and its applications in biomedicine. *Genomics, Proteomics & Bioinformatics*, 16(1):17–32.

Carney, J. G., Cunningham, P., and Bhagwan, U. (1999). Confidence and prediction intervals for neural network ensembles. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, volume 2, pages 1215–1218. IEEE.

Caruana, R., Lawrence, S., and Giles, L. (2001). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. *Advances in neural information processing systems*, pages 402–408.

Cassotti, M., Ballabio, D., Consonni, V., Mauri, A., Tetko, I. V., and Todeschini, R. (2014). Prediction of acute aquatic toxicity toward *Daphnia Magna* by using the GA-*k*NN method. *Alternatives to Laboratory Animals*, 42(1):31–41.

Cauchy, A. et al. (1847). Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538.

Chung, B.-M. (2018). Neural-network model for compensation of lens distortion in camera calibration. *International Journal of Precision Engineering and Manufacturing*, 19(7):959–966.

Cooper, J., Vik, J. O., and Waltemath, D. (2015). A call for virtual experiments: Accelerating the scientific process. *Progress in Biophysics and Molecular Biology*, 117(1):99–106. Multi-scale Systems Biology.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Dargan, S., Kumar, M., Ayyagari, M. R., and Kumar, G. (2020). A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 27(4):1071–1092.

de Bézenac, E., Pajot, A., and Gallinari, P. (2019). Deep learning for physical processes: incorporating prior scientific knowledge. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124009.

DeVries, G. M., Fleig, J. F., and Murphy, P. E. (2006). Challenges with interferometric non-null measurements. In *Optical Fabrication and Testing*, page OFWD5. Optical Society of America.

Dhillon, A. and Verma, G. K. (2020). Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence*, 9(2):85–112.

Dieck, R. H. (2007). *Measurement uncertainty: methods and applications*. ISA.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern classification*. A Wiley-interscience publication. Wiley, New York u.a., 2. ed. edition.

Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.

Duvenaud, D., Maclaurin, D., and Adams, R. (2016). Early stopping as nonparametric variational inference. In *Artificial Intelligence and Statistics*, pages 1070–1077. PMLR.

Elster, C. (2014). Bayesian uncertainty analysis compared with the application of the GUM and its supplements. *Metrologia*, 51(4):S159.

Engl, H. W., Hanke, M., and Neubauer, A. (1996). *Regularization of inverse problems*, volume 375. Springer Science & Business Media.

Esser, P. and Sutter, E. (2018). A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, Los Alamitos, CA, USA. IEEE Computer Society.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., and Dean, J. (2019). A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29.

EURAMET (2011-2014). Optical and tactile metrology for absolute form characterisation, Project Number: IND10. `https://www.euramet.org/research-innovation/search-research-projects/details/project/optical-and-tactile-metrology-for-absolute-form-characterisation/`. Accessed: 2021-10-20.

EURAMET (2016-2019). Reference algorithms and metrology on aspherical and freeform lenses, Project Number: 15SIB01. `https://www.euramet.org/research-innovation/search-research-projects/details/project/reference-algorithms-and-metrology-on-aspherical-and-freeform-lenses/`. Accessed: 2021-10-20.

Fang, F., Cheng, Y., and Zhang, X. (2013a). Design of freeform optics. *Advanced Optical Technologies*, 2(5-6):445–453.

Fang, F., Zhang, X., Weckenmann, A., Zhang, G., and Evans, C. (2013b). Manufacturing and measurement of freeform optics. *CIRP Annals*, 62(2):823–846.

Fischedick, M., Stavridis, M., Bartl, G., and Elster, C. (2021). Investigation of the uncertainty contributions of the alignment of PTB's double-ended interferometer by virtual experiments. *Metrologia*, 58(6):064001.

Flack, D. (2014). Measurement good practice guide no. 41: Cmm measurement strategies. *National Physical Laboratory, London, UK*.

Fort, S., Hu, H., and Lakshminarayanan, B. (2019). Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*.

Fortmeier, I. (2016). *Zur Optimierung von Auswerteverfahren für Tilted-Wave Interferometer*. Institut für Technische Optik, Universität Stuttgart.

Fortmeier, I., Schachtschneider, R., Ledl, V., Matousek, O., Siepmann, J., Harsch, A., Beisswanger, R., Bitou, Y., Kondo, Y., Schulz, M., et al. (2020). Round robin comparison study on the form measurement of optical freeform surfaces. *Journal of the European Optical Society-Rapid Publications*, 16(1):1–15.

Fortmeier, I. and Schulz, M. (2022). Comparison of form measurement results for optical aspheres and freeform surfaces. *Measurement Science and Technology*.

Fortmeier, I., Schulz, M., and Meeß, R. (2019). Traceability of form measurements of freeform surfaces: metrological reference surfaces. *Optical Engineering*, 58(9):1 – 7.

Fortmeier, I., Stavridis, M., Elster, C., and Schulz, M. (2017). Steps towards traceability for an asphere interferometer. In Lehmann, P., Osten, W., and Jr., A. A. G., editors, *Optical Measurement Systems for Industrial Inspection X*, volume 10329, pages 790 – 798. International Society for Optics and Photonics, SPIE.

Fortmeier, I., Stavridis, M., Schulz, M., and Elster, C. (2022). Development of a metrological reference system for the form measurement of aspheres and freeform surfaces based on a tilted-wave interferometer. *Measurement Science and Technology*.

Fortmeier, I., Stavridis, M., Wiegmann, A., Schulz, M., Baer, G., Pruss, C., Osten, W., and Elster, C. (2013). Sensitivity analysis of tilted-wave interferometer asphere measurements using virtual experiments. In *Modeling Aspects in Optical Metrology IV*, volume 8789, page 878907. International Society for Optics and Photonics.

Fortmeier, I., Stavridis, M., Wiegmann, A., Schulz, M., Osten, W., and Elster, C. (2014). Analytical Jacobian and its application to tilted-wave interferometry. *Optics express*, 22(18):21313–21325.

Fortmeier, I., Stavridis, M., Wiegmann, A., Schulz, M., Osten, W., and Elster, C. (2016). Evaluation of absolute form measurements using a tilted-wave interferometer. *Opt. Express*, 24(4):3393–3404.

Free Dictionary (2003). Mcgraw-hill dictionary of scientific & technical terms, 6e. `https://encyclopedia2.thefreedictionary.com/optical+surface`. Accessed: 2021-10-14.

Gal, Y. (2016). *Uncertainty in deep learning*. PhD thesis, University of Cambridge.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Gal, Y., Hron, J., and Kendall, A. (2017). Concrete dropout. In *Advances in neural information processing systems*, volume 30, pages 3581–3590. Curran Associates, Inc.

Ganaie, M., Hu, M., et al. (2021). Ensemble deep learning: A review. *arXiv preprint arXiv:2104.02395*.

Garbusi, E., Baer, G., and Osten, W. (2011). Advanced studies on the measurement of aspheres and freeform surfaces with the tilted-wave interferometer. In Lehmann, P. H., Osten, W., and Gastinger, K., editors, *Optical Measurement Systems for Industrial Inspection VII*, volume 8082, pages 446 – 456. International Society for Optics and Photonics, SPIE.

Garbusi, E., Pruss, C., and Osten, W. (2008). Interferometer for precise and flexible asphere testing. *Optics Letters*, 33(24):2973–2975.

Gardner, M. and Dorling, S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14):2627–2636.

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., et al. (2021). A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995). *Bayesian data analysis*. Chapman and Hall/CRC.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58.

Gerritsma, J., Onnink, R., and Versluis, A. (1981). Geometry, resistance and stability of the delft systematic yacht hull series. *International shipbuilding progress*, 28(328):276–297.

Ghanem, R., Higdon, D., Owhadi, H., et al. (2017). *Handbook of uncertainty quantification*, volume 6. Springer.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Goan, E. and Fookes, C. (2020). Bayesian neural networks: An introduction and survey. In *Case Studies in Applied Bayesian Data Science*, pages 45–87. Springer.

Goldstein, R. M., Zebker, H. A., and Werner, C. L. (1988). Satellite radar interferometry: Two-dimensional phase unwrapping. *Radio Science*, 23(4):713–720.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386.

Gronle, A., Pruss, C., and Herkommer, A. (2022). Misalignment of spheres, aspheres and freeforms in optical measurement systems. *Opt. Express*, 30(2):797–814.

Gustafsson, F. K., Danelljan, M., and Schon, T. B. (2020). Evaluating scalable Bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 318–319.

Hadamard, J. (1902). Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton university bulletin*, pages 49–52.

Hagemann, P. L. and Neumayer, S. (2021). Stabilizing invertible neural networks using mixture models. *Inverse Problems*.

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.

Hariharan, P. (2010). *Basics of interferometry*. Elsevier.

Hariharan, P., Oreb, B. F., and Eiju, T. (1987). Digital phase-shifting interferometry: a simple error-compensating phase calculation algorithm. *Applied optics*, 26(13):2504–2506.

Harsch, A., Pruss, C., Baer, G., and Osten, W. (2019). Monte Carlo simulations: a tool to assess complex measurement systems. In Rascher, R. and Schopf, C., editors, *Sixth European Seminar on Precision Optics Manufacturing*, volume 11171, pages 66 – 72. International Society for Optics and Photonics, SPIE.

Hashemi, A., Gao, Y., Cai, C., Ghosh, S., Müller, K. R., Nagarajan, S. S., and Haufe, S. (2021). Efficient hierarchical Bayesian inference for spatio-temporal regression models in neuroimaging. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

Haufe, S., Nikulin, V. V., Ziehe, A., Müller, K.-R., and Nolte, G. (2008). Estimating vector fields using sparse basis field expansions. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 21*, pages 617–624, Cambridge, MA. MIT Press.

Häusler, G., Faber, C., Olesch, E., and Ettl, S. (2013). Deflectometry vs. interferometry. In Lehmann, P. H., Osten, W., and Albertazzi, A., editors, *Optical Measurement Systems for Industrial Inspection VIII*, volume 8788, pages 367 – 377. International Society for Optics and Photonics, SPIE.

He, B., Lakshminarayanan, B., and Teh, Y. W. (2020). Bayesian deep ensembles via the neural tangent kernel. *arXiv preprint arXiv:2007.05864*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hecht, E. (2018). *Optik*. Walter de Gruyter GmbH & Co KG.

Henne, M., Schwaiger, A., Roscher, K., and Weiss, G. (2020). Benchmarking uncertainty estimation methods for deep learning with safety-related metrics. In *SafeAI@ AAAI*, pages 83–90.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–401.

Hoffmann, L. and Elster, C. (2020). Deep neural networks for computational optical form measurements. *Journal of Sensors and Sensor Systems*, 9(2):301–307.

Hoffmann, L. and Elster, C. (2021). Deep ensembles from a Bayesian perspective. *arXiv preprint arXiv:2105.13283*.

Hoffmann, L., Fortmeier, I., and Elster, C. (2021a). Deep learning for tilted-wave interferometry. *tm - Technisches Messen*, 89(1):33–42.

Hoffmann, L., Fortmeier, I., and Elster, C. (2021b). Uncertainty quantification by ensemble learning for computational optical form measurements. *Machine Learning: Science and Technology*.

Hopp, T. H. et al. (1993). Computational metrology. *Manufacturing Review*, 6:295–295.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Hu, S., Pezzotti, N., and Welling, M. (2020). A new perspective on uncertainty quantification of deep ensembles. *arXiv e-prints*, pages arXiv–2002.

Huang, L., Idir, M., Zuo, C., and Asundi, A. (2018). Review of phase measuring deflectometry. *Optics and Lasers in Engineering*, 107:247–257.

Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

Işıl, Ç., Oktem, F. S., and Koç, A. (2019). Deep iterative reconstruction for phase retrieval. *Applied Optics*, 58(20):5422–5431.

ISO 10110-12:2019 (2019). Optics and photonics — Preparation of drawings for optical elements and systems — Part 12: Aspheric surfaces. Technical report, ISO.

ISO 10110-8:2019 (2019). Optics and photonics — Preparation of drawings for optical elements and systems — Part 8: Surface texture. Technical report, ISO.

Jospin, L. V., Buntine, W., Boussaid, F., Laga, H., and Bennamoun, M. (2020). Hands-on Bayesian neural networks–a tutorial for deep learning users. *arXiv preprint arXiv:2007.06823*.

Kando, D., Tomioka, S., Miyamoto, N., and Ueda, R. (2019). Phase extraction from single interferogram including closed-fringe using deep learning. *Applied Sciences*, 9(17):3529.

Karpatne, A., Atluri, G., Faghmous, J. H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., and Kumar, V. (2017). Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering*, 29(10):2318–2331.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Kingma, D. P., Salimans, T., and Welling, M. (2015). Variational dropout and the local reparameterization trick. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.

Kononenko, I. (1989). Bayesian neural networks. *Biological Cybernetics*, 61(5):361–370.

Kretz, T., Müller, K.-R., Schaeffter, T., and Elster, C. (2020). Mammography image quality assurance using deep learning. *IEEE Transactions on Biomedical Engineering*, 67(12):3317–3326.

Kristiadi, A., Hein, M., and Hennig, P. (2020). Being Bayesian, even just a bit, fixes overconfidence in relu networks. In *International Conference on Machine Learning*, pages 5436–5446. PMLR.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86.

Kuutti, S., Bowden, R., Jin, Y., Barber, P., and Fallah, S. (2020). A survey of deep learning applications to autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):712–733.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30.

Lampinen, J. and Vehtari, A. (2001). Bayesian approach for neural networks—review and case studies. *Neural networks*, 14(3):257–274.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. (2019). Unmasking Clever Hans predictors and assessing what machines really learn. *Nature communications*, 10:1096.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

LeCun, Y., Denker, J., and Solla, S. (1989a). Optimal brain damage. In Touretzky, D., editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

LeCun, Y. et al. (1989b). Generalization and network design strategies. *Connectionism in perspective*, 19:143–155.

LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R. (2012). Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer.

Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., and Batra, D. (2015). Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*.

Leibig, C., Allken, V., Ayhan, M. S., Berens, P., and Wahl, S. (2017). Leveraging uncertainty information from deep neural networks for disease detection. *Scientific Reports*, 7(1):1–14.

LeNail, A. (2019). NN-SVG: Publication-ready neural network architecture schematics. *Journal of Open Source Software*, 4(33):747.

Lerner, S. A. and Sasian, J. M. (2000). Use of implicitly defined optical surfaces for the design of imaging and illumination systems. *Optical Engineering*, 39(7):1796 – 1801.

Li, H., Schwab, J., Antholzer, S., and Haltmeier, M. (2020). NETT: solving inverse problems with deep neural networks. *Inverse Problems*, 36(6):065005.

Liang, D., Cheng, J., Ke, Z., and Ying, L. (2020). Deep magnetic resonance image reconstruction: Inverse problems meet neural networks. *IEEE Signal Processing Magazine*, 37(1):141–151.

Liang, S., Li, Y., and Srikant, R. (2018). Enhancing the reliability of out-of-distribution image detection in neural networks. *International Conference on Learning Representations (ICLR)*.

Liesener, J. (2007). *Zum Einsatz räumlicher Lichtmodulatoren in der interferometrischen Wellenfrontmesstechnik*. Institut für Technische Optik, Universität Stuttgart.

Lucas, A., Iliadis, M., Molina, R., and Katsaggelos, A. K. (2018). Using deep neural networks for inverse problems in imaging: Beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36.

MacKay, D. J. C. (1992). A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3):448–472.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2019). A simple baseline for Bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32:13153–13164.

Mahr GmbH (2021). MarOpto TWI 60. `https://www.mahr.de/en-us/Services/Production-metrology/Products/MarOpto---Measuring-Devices-for-Optics-Industry/MarOpto-TWI-60/`. Accessed: 2021-10-14.

Malacara, D. (2007). *Optical shop testing*, volume 59. John Wiley & Sons.

Martin, J. and Elster, C. (2021). Detecting unusual input to neural networks. *Applied Intelligence*, 51(4):2198–2209.

McCann, M. T., Jin, K. H., and Unser, M. (2017). Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Processing Magazine*, 34(6):85–95.

Michelmore, R., Wicker, M., Laurenti, L., Cardelli, L., Gal, Y., and Kwiatkowska, M. (2020). Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7344–7350. IEEE.

Montavon, G., Samek, W., and Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.

Moraru, A., Pesko, M., Porcius, M., Fortuna, C., and Mladenic, D. (2010). Using machine learning on sensor data. *Journal of computing and information technology*, 18(4):341–347.

Mousavi, A. and Baraniuk, R. G. (2017). Learning to invert: Signal recovery via deep convolutional networks. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2272–2276. IEEE.

Mühlig, S., Siepmann, J., Lotz, M., Jung, S., Schindler, J., and Baer, G. (2014). Tilted wave interferometer-improved measurement uncertainty. In *58 th Ilmenau Scientific Colloquium*.

Murphy, P., Forbes, G., Fleig, J., Dumas, P., and Tricard, M. (2003). Stitching interferometry: a flexible solution for surface metrology. *Optics and Photonics News*, 14(5):38–43.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Icml*.

Neal, B., Mittal, S., Baratin, A., Tantia, V., Scicluna, M., Lacoste-Julien, S., and Mitliagkas, I. (2018). A modern take on the bias-variance tradeoff in neural networks. *arXiv preprint arXiv:1810.08591*.

Neal, R. M. (1992). Bayesian training of backpropagation networks by the hybrid Monte Carlo method. Technical report, Citeseer.

Neal, R. M. (1995). *Bayesian learning for neural networks*. PhD thesis, Citeseer.

Nguyen, V.-L., Destercke, S., and Hüllermeier, E. (2019). Epistemic uncertainty sampling. In *International Conference on Discovery Science*, pages 72–86. Springer.

Nielsen, M. A. (2015). *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA.

Nikolenko, S. I. et al. (2021). *Synthetic data for deep learning*. Springer.

Nishizaki, Y., Valdivia, M., Horisaki, R., Kitaguchi, K., Saito, M., Tanida, J., and Vera, E. (2019). Deep learning wavefront sensing. *Optics express*, 27(1):240–251.

NTSB (2019). Highway investigation. *National Transport Safety Board*, HWY18MH010.

O'Hagan, A. and Forster, J. J. (2004). *Kendall's advanced theory of statistics, volume 2B: Bayesian inference*, volume 2. Arnold.

Ortigosa, I., Lopez, R., and Garcia, J. (2007). A neural networks approach to residuary resistance of sailing yachts prediction. In *Proceedings of the international conference on marine engineering MARINE*, volume 2007, page 250.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Oxford University Press (2021a). `https://www.lexico.com/definition/artificial_intelligence`. Accessed: 2021-11-01.

Oxford University Press (2021b). `https://www.lexico.com/definition/machine_learning`. Accessed: 2021-11-01.

Oxford University Press (2021c). `https://www.lexico.com/definition/deep_learning`. Accessed: 2021-11-02.

Park, S., Yun, C., Lee, J., and Shin, J. (2020). Minimum width for universal approximation. In *International Conference on Learning Representations*.

Park, S. R. and Lee, J. (2016). A fully convolutional neural network for speech enhancement. *arXiv preprint arXiv:1609.07132*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pearce, T., Leibfried, F., and Brintrup, A. (2020). Uncertainty in neural networks: Approximately Bayesian ensembling. In *International conference on artificial intelligence and statistics*, pages 234–244. PMLR.

Pruss, C., Baer, G. B., Schindler, J., and Osten, W. (2017). Measuring aspheres quickly: tilted wave interferometry. *Optical Engineering*, 56(11):1 – 7.

Pruss, C., Reichelt, S., Tiziani, H. J., and Osten, W. (2004). Computer-generated holograms in interferometric testing. *Optical Engineering*, 43(11):2534 – 2540.

Raissi, M. (2018). Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, 19(1):932–955.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X., and Wang, X. (2021). A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54(9):1–40.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.

Richter, M. (2020). *Discretization of Inverse Problems*, pages 31–83. Springer International Publishing, Cham.

Ritter, H., Botev, A., and Barber, D. (2018). A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning.

Rivenson, Y., Zhang, Y., Günaydın, H., Teng, D., and Ozcan, A. (2018). Phase recovery and holographic image reconstruction using deep learning in neural networks. *Light: Science & Applications*, 7(2):17141–17141.

Robert, C. (2007). *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media.

Rolland, J. P., Davies, M. A., Suleski, T. J., Evans, C., Bauer, A., Lambropoulos, J. C., and Falaggis, K. (2021). Freeform optics for imaging. *Optica*, 8(2):161–176.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Sagi, O. and Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249.

Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., and Müller, K.-R. (2021). Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278.

Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., and Müller, K.-R. (2019). *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature.

Santos, L., Santos, F. N., Oliveira, P. M., and Shinde, P. (2019). Deep learning applications in agriculture: A short review. In *Iberian Robotics conference*, pages 139–151. Springer.

Sasakawa, T., Hu, J., and Hirasawa, K. (2008). A brainlike learning system with supervised, unsupervised, and reinforcement learning. *Electrical Engineering in Japan*, 162(1):32–39.

Scalia, G., Grambow, C. A., Pernici, B., Li, Y.-P., and Green, W. H. (2020). Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. *Journal of chemical information and modeling*, 60(6):2697–2717.

Schachtschneider, R., Fortmeier, I., Stavridis, M., Asfour, J., Berger, G., Bergmann, R., Beutler, A., Blümel, T., Klawitter, H., Kubo, K., et al. (2018). Interlaboratory comparison measurements of aspheres. *Measurement Science and Technology*, 29(5):055010.

Schachtschneider, R., Stavridis, M., Fortmeier, I., Schulz, M., and Elster, C. (2019). Simoptdevice: a library for virtual optical experiments. *Journal of Sensors and Sensor Systems*, 8(1):105–110.

Schenker, M., Stavridis, M., Schulz, M., and Tutsch, R. (2020). Effects of misalignments on the modulation transfer function measurement of camera lenses analyzed in optomechanical simulations. *Optical Engineering*, 59(3):1 – 17.

Schindler, J. (2020). *Methoden zur selbstkalibrierenden Vermessung von Asphären und Freiformflächen in der Tilted-Wave-Interferometrie*, volume 105. Institut für Technische Optik, Universität Stuttgart.

Schindler, J., Pruss, C., and Osten, W. (2017). Increasing the accuracy of tilted-wave-interferometry by elimination of systematic errors. In *Optical Measurement Systems for Industrial Inspection X*, volume 10329, page 1032904. International Society for Optics and Photonics.

Schmähling, F., Martin, J., and Elster, C. (2021). A framework for benchmarking uncertainty in deep regression. *arXiv preprint arXiv:2109.09048*.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

Scholz, G., Fortmeier, I., Marschall, M., Stavridis, M., Schulz, M., and Elster, C. (2022). Experimental design for virtual experiments in tilted-wave interferometry. *Metrology*, 2(1):84–97.

Schulz, G. (1987). Imaging performance of aspherics in comparison with sphericalsurfaces. *Appl. Opt.*, 26(23):5118–5124.

Schulz, M., Blobel, G., Fortmeier, I., Stavridis, M., and Elster, C. (2015). Traceability in interferometric form metrology. In Lehmann, P., Osten, W., and Jr., A. A. G., editors, *Optical Measurement Systems for Industrial Inspection IX*, volume 9525, pages 325 – 333. International Society for Optics and Photonics, SPIE.

Schulz, M., Ehret, G., Stavridis, M., and Elster, C. (2010). Concept, design and capability analysis of the new deflectometric flatness reference at ptb. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 616(2):134–139. X-Ray Mirror.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Shannon, R. R. (1997). *The art and science of optical design*. Cambridge University Press.

Sharma, S., Sharma, S., and Athaiya, A. (2017). Activation functions in neural networks. *towards data science*, 6(12):310–316.

Shinde, P. P. and Shah, S. (2018). A review of machine learning and deep learning applications. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pages 1–6.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48.

Sola, J. and Sevilla, J. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*, 44(3):1464–1468.

Sonoda, S. and Murata, N. (2017). Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43(2):233–268.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Staar, B., Lütjen, M., and Freitag, M. (2019). Anomaly detection with convolutional neural networks for industrial surface inspection. *Procedia CIRP*, 79:484–489.

Stavridis, M., Fortmeier, I., Schulz, M., and Elster, C. (2018). SimOptDevice Werkzeug für virtuelle Experimente. `https://www.ptb.de/cms/fileadmin/internet/PSt/pst1/Vortraege_VirtMess2018/SimOptDevice_stavridis.pdf`. VirtMess2018 talk. Accessed: 2022-01-14.

Sun, R., Li, D., Liang, S., Ding, T., and Srikant, R. (2020). The global landscape of neural networks: An overview. *IEEE Signal Processing Magazine*, 37(5):95–108.

Sun, Y., Huang, X., Kroening, D., Sharp, J., Hill, M., and Ashmore, R. (2019). Structural test coverage criteria for deep neural networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 18(5s):1–23.

Sun, Y., Wu, M., Ruan, W., Huang, X., Kwiatkowska, M., and Kroening, D. (2018). Concolic testing for deep neural networks. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, pages 109–119.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Taylor, B. (2006). *Methods and Procedures for the Verification and Validation of Artificial Neural Networks*. Springer US.

Thalhammer, R. and Wachutka, G. (2003a). Virtual optical experiments. part i. modeling the measurement process. *J. Opt. Soc. Am. A*, 20(4):698–706.

Thalhammer, R. and Wachutka, G. (2003b). Virtual optical experiments. part ii. design of experiments. *J. Opt. Soc. Am. A*, 20(4):707–713.

The MathWorks, I. (2018). Deep Learning Toolbox. `https://www.mathworks.com/help/deeplearning/`. Accessed: 2022-02-17.

Thompson, K. P. and Rolland, J. P. (2012). Freeform optical surfaces: A revolution in imaging optical design. *Opt. Photon. News*, 23(6):30–35.

Tian, Y., Pei, K., Jana, S., and Ray, B. (2018). Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering*, ICSE '18, page 303–314, New York, NY, USA. Association for Computing Machinery.

Tiwari, M. K. and Chatterjee, C. (2010). Uncertainty assessment and ensemble flood forecasting using bootstrap based artificial neural networks (banns). *Journal of Hydrology*, 382(1-4):20–33.

Vahdat, A. and Kautz, J. (2020). Nvae: A deep hierarchical variational autoencoder. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19667–19679. Curran Associates, Inc.

Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

Vdovin, G. V. (1995). Model of an adaptive optical system controlled by a neural network. *Optical Engineering*, 34(11):3249–3253.

Wang, F., Zhang, M., Wang, X., Ma, X., and Liu, J. (2020). Deep learning for edge computing applications: A state-of-the-art survey. *IEEE Access*, 8:58322–58336.

Wang, H., Bah, M. J., and Hammad, M. (2019). Progress in outlier detection techniques: A survey. *Ieee Access*, 7:107964–108000.

Wang, J. Y. and Silva, D. E. (1980). Wave-front interpretation with zernike polynomials. *Appl. Opt.*, 19(9):1510–1518.

Weinshall, D., Cohen, G., and Amir, D. (2018). Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International Conference on Machine Learning*, pages 5238–5246. PMLR.

Wenzel, F., Snoek, J., Tran, D., and Jenatton, R. (2020). Hyperparameter ensembles for robustness and uncertainty quantification. *arXiv preprint arXiv:2006.13570*.

Wiegmann, A., Stavridis, M., Walzel, M., Siewert, F., Zeschke, T., Schulz, M., and Elster, C. (2011). Accuracy evaluation for sub-aperture interferometry measurements of a synchrotron mirror using virtual experiments. *Precision Engineering*, 35(2):183–190.

Wilson, A. G. and Izmailov, P. (2020). Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*.

Xu, K., Li, J., Zhang, M., Du, S. S., Kawarabayashi, K.-i., and Jegelka, S. (2020). How neural networks extrapolate: From feedforward to graph neural networks. *arXiv preprint arXiv:2009.11848*.

Yaman, F., Yakhno, V. G., and Potthast, R. (2013). A survey on inverse problems for applied sciences. *Mathematical problems in engineering*, 2013.

Yan, K., Yu, Y., and Jiaxing, L. (2018). Neural networks for interferograms recognition. In *icOPEN 2018*, volume 10827, page 108273Q. International Society for Optics and Photonics.

Yao, J., Pan, W., Ghosh, S., and Doshi-Velez, F. (2019). Quality of uncertainty quantification for Bayesian neural network inference. *arXiv preprint arXiv:1906.09686*.

Yeom, S.-K., Seegerer, P., Lapuschkin, S., Binder, A., Wiedemann, S., Müller, K.-R., and Samek, W. (2021). Pruning by explaining: A novel criterion for deep neural network pruning. *Pattern Recognition*, 115:107899.

Ying, X. (2019). An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, volume 1168, page 022022. IOP Publishing.

Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018). Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75.

Zeng, J., Lesnikowski, A., and Alvarez, J. M. (2018). The relevance of Bayesian layer positioning to model uncertainty in deep Bayesian active learning. *arXiv preprint arXiv:1811.12535*.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017). Beyond a Gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155.

Zhang, L., Li, C., Zhou, S., Li, J., and Yu, B. (2020). Enhanced calibration for freeform surface misalignments in non-null interferometers by convolutional neural network. *Optics Express*, 28(4):4988–4999.

Zhang, L., Zhou, S., Li, J., and Yu, B. (2019). Deep neural network based calibration for freeform surface misalignments in general interferometer. *Optics express*, 27(23):33709–33723.

Zhou, X., Liu, H., Pourpanah, F., Zeng, T., and Wang, X. (2021). A survey on epistemic (model) uncertainty in supervised learning: Recent advances and applications. *Neurocomputing*.