Improving Automotive Radar Spectra Object Classification using Deep Learning and Multi-Class Uncertainty Calibration

von der Fakultät Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart
zur Erlangung der Würde eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigte
Abhandlung

vorgelegt von

Kanil Patel

aus Johannesburg, South Africa

Hauptberichter:	Prof. Dr.-Ing. Bin Yang
Mitberichter:	Prof. Dr. Steffen Staab

Tag der mündlichen Prüfung: 13.07.2022

Institut für Signalverarbeitung und Systemtheorie
der Universität Stuttgart

2022

Abstract

Being a prerequisite for successful automated driving, ameliorating the perception capabilities of vehicles is of paramount importance for reliable and robust scene understanding. Required for decision-making in autonomous vehicles, scene understanding becomes particularly challenging in adverse weather and lighting conditions; situations also often posing challenges for human drivers. Automotive radars can greatly assist sensors currently deployed on vehicles for robust measurements, especially in challenging conditions where other sensors often fail to operate reliably. However, classification using radar sensors is often limited to a few classes (e.g. cars, humans, and stop signs), controlled laboratory settings, and/or simulations. Already offering reliable distance, azimuth and velocity estimates of the objects in the scene, improving radar-based classification greatly expands the usage of radar sensors for tackling multiple driving-related tasks which are often performed by other less robust sensors. This thesis investigates how automated driving perception can be improved using multi-class radar classification by using deep learning algorithms for exploiting object class characteristics captured in the radar spectra. Despite the highly-accurate predictions of deep learning models, such classifiers exhibit severe over-confidence which can lead decision-making systems to false conclusions, with possibly catastrophic consequences - often a matter of life and death for automated driving. Consequently, high-quality, robust, and interpretable uncertainty estimates are indispensable characteristics of any unassailable automated driving system. With the goal of uncertainty estimates for real-time predictive systems, this thesis also aims at tackling the prominent over-confidence of deep learning classification models, which persists for all data modalities. Being an important measure for the quality of uncertainty estimates, this work focuses on the accurate estimation of the calibration of trained classifiers, as well as present novel techniques for improving their calibration. The presented solutions offer high-quality real-time confidence estimates for classification models of all data modalities (e.g. non-radar applications), as well as classifiers which are already trained and used in practise and new training strategies for learning new classifiers. Furthermore, the presented uncertainty calibration algorithms could also be extended to tasks other than classification, for example, regression and segmentation. On a challenging new realistic automated driving radar dataset, the solutions proposed in this thesis show that radar classifiers are able to generalize to novel driving environments, driving patterns, and object instances in realistic static driving scenes. To further replicate realistic encounters of autonomous vehicles, we study the behaviour of the classifiers to spectra corruptions and outlier detection of unknown objects, showing significant performance improvements in safely handling these prevalent encounters through accurate uncertainty estimates. With the proposed generalization and requisite accurate uncertainty estimation techniques, the radar classifiers in this study greatly improve radar-based perception for scene understanding and lay a solid foundation for current sensor fusion techniques to leverage radar measurements for object classification.

Zusammenfassung

Als Voraussetzung für erfolgreiches automatisiertes Fahren ist die Verbesserung der Wahrnehmungsfähigkeiten von Fahrzeugen von entscheidender Bedeutung für ein zuverlässiges und robustes Szenenverständnis. Das für die Entscheidungsfindung in autonomen Fahrzeugen erforderliche Szenenverständnis wird bei ungünstigen Wetter- und Lichtverhältnissen besonders anspruchsvoll; solche Situationen stellen auch für menschliche Fahrer oft eine große Herausforderung dar. Fahrzeugradare können die derzeit in Fahrzeugen eingesetzten Sensoren bei robusten Messungen erheblich unterstützen, insbesondere unter schwierigen Bedingungen, bei denen andere Sensoren oft nicht zuverlässig arbeiten. Allerdings ist die Klassifizierung mit Radarsensoren oft auf einige wenige Klassen (z. B. Autos, Menschen und Stoppschilder), kontrollierte Laboreinstellungen und/oder Simulationen beschränkt. Radar bietet bereits zuverlässige Abstands-, Azimut- und Geschwindigkeitsschätzungen der Objekte in einer Szene, daher erweitert eine verbesserte radar-basierte Klassifikation den Einsatzbereich von Radarsensoren noch deutlich, und kann somit zur Bewältigung zahlreicher fahrbezogener Aufgaben verwendet werden, die oft von anderen, weniger robusten Sensoren übernommen werden. In dieser Arbeit wird untersucht, wie die Perzeption automatisierter Fahrzeuge durch Mehrklassen-Radarklassifikation verbessert werden kann, bei der Deep-Learning-Algorithmen zum Lernen und Erkennen typischer Objektklassenmerkmale in Radarspektren zum Einsatz kommen. Trotz der hochpräzisen Vorhersagen von Deep-Learning-Modellen weisen solche Klassifikatoren eine stark überhöhte Konfidenz auf, die Entscheidungssysteme zu falschen Schlussfolgerungen verleiten kann, mit möglicherweise katastrophalen Folgen - beim automatisierten Fahren oft eine Frage von Leben und Tod. Folglich sind qualitativ hochwertige, robuste und interpretierbare Unsicherheitsschätzungen unabdingbare Eigenschaften jedes automatisierten Fahrsystems. Mit dem Ziel von Unsicherheitsschätzungen für Vorhersagen in Echtzeit zielt diese Arbeit auch darauf ab, die bekannte Über-Konfidenz von Deep-Learning-Klassifikationsmodellen zu addressieren, die für alle Datenmodalitäten besteht. Als wichtiges Maß für die Qualität von Unsicherheitsschätzungen konzentriert sich diese Arbeit auf die genaue Schätzung der Kalibrierung von trainierten Klassifikatoren, sowie auf die Entwicklung neuartiger Techniken zur Verbesserung ihrer Kalibrierung. Die vorgestellten Lösungen bieten hochwertige Konfidenzschätzungen in Echtzeit für Klassifikationsmodelle aller Datenmodalitäten (z.B. für Anwendungen außerhalb von Radar), sowie für bereits trainierte und in der Praxis eingesetzte Klassifikatoren und neue Trainingsstrategien zum Erlernen neuer Klassifikatoren. Darüber hinaus könnten die vorgestellten Algorithmen zur Unsicherheitskalibrierung auch auf andere Aufgaben jenseits von Klassifikation erweitert werden, z. B. auf Regression und Segmentierung. Anhand eines anspruchsvollen neuen realistischen Radardatensatzes für automatisiertes Fahren zeigen die in dieser Arbeit vorgeschlagenen Lösungen, dass Radarklassifikatoren in der Lage sind, auf neuartige Fahrumgebungen, Fahrmuster und Objektinstanzen in realistischen statischen Fahrszenen zu generalisieren. Um weitere realistische Situationen für autonome Fahrzeuge zu replizieren, untersuchen wir das Verhalten der Klassifikatoren auf verfälschten Spektren und zur Erkennung von davor unbekannten Objekten. Wir zeigen signifikante Leistungsverbesserungen im sicheren Umgang mit diesen häufigen Situationen durch genaue Unsicherheitsschätzungen.

Mit der vorgeschlagenen Verallgemeinerung und den erforderlichen genauen Unsicherheitsschätzungen verbessern die Radarklassifikatoren in dieser Arbeit die radarbasierte Wahrnehmung für das Szenenverständnis erheblich. Damit bilden sie eine solide Grundlage um Radarmessungen für gängige Sensor-Fusionstechniken zur Objektklassifizierung vorteilhaft einzusetzen.

Acknowledgements

I have been looking forward to writing the following acknowledgements as it marks the end of a three year journey which I am glad to, soon, officially conclude. The completion of this thesis could not have been possible without the exceptional and amazing support I have received at every point during this journey. Now, it is the time to finally say "thank you".

There are a couple of people without whom this thesis would not possible.

Firstly, I would like to thank my university supervisor, Prof. Bin Yang, for his support throughout the past years. I greatly appreciate the freedom I have received to develop my own research agenda, as well as the interest he has shown in my work. My Bosch supervisor and I, both thank him for his trust in us to find a suitable research topic without being too restrictive. He also played an important role in suggesting the combination of the radar and uncertainty topics presented in this thesis. Lastly, he has also always been very accommodating to my publication plans and PhD timeline.

I also want to express my deepest appreciation to my Bosch supervisor, Michael, for his continuous support and patience through the years. Michael has played a decisive role in successfully helping me complete my PhD and his help at various points in time (especially the days leading up to conference deadlines) cannot be overestimated. I have learned a great deal about research, possible applications of research, and scientific writing from Michael. His expertise were invaluable in helping me formulate the right research questions and his insightful feedback greatly helped me develop my research skills. With the help of Michael, we have managed to turn simple fundamental ideas into publications. Michael has a great way with words, and as a result, he has been especially helpful at the writing phases of all publications and this thesis. Even though he had very little time given his other responsibilities, Michael never made me or my PhD feel like a low-priority among his many duties. I am very grateful for his help.

I am also deeply indebted to Dan, who has played an instrumental role in my thesis with her guidance and novel ideas. She has not only taught me how to ask and answer the correct research questions, but also a great deal about turning an idea into a successful research project followed by a publication. Even though Dan was not my official PhD supervisor, she has always been available for discussions and treated me as one of her PhD students. For the amount of time she has given me, I cannot thank her enough.

I gratefully recognize the help of another colleague Bill. Being experienced with the uncertainty topic himself, Bill eased my transition to the uncertainty topic. It is very important to strike a healthy balance between life and the demanding PhD. With the constant deadlines, and tremendous amount of stress which comes with it, it becomes hard to remember this. Along side being a co-author for many of my publications and offering his expertise on the topic, Bill played another, arguably more important, role as my close friend. It almost felt as if he was tasked with reminding me about the healthy work-life balance, which I often neglected. Bill has stayed up and worked with me until the early hours of the day, as well as on some weekends, during the most stressful times of this PhD. He often got me to destress, as well as helped me break out of my unproductive pondering sessions. Lastly, he has always offered a listening ear to my rants and laments about the PhD, and calmed me down in most stressful situations, surprisingly even when I felt like I had reached an impasse.

I would also like to thank my colleagues Kilian (my co-supervisor on the radar topic and my personal-and-always-available radar expert) and Eliza (my favourite work-neighbour who always found a way to make me laugh and listened to my weekly/monthly rants about something). I am also grateful for my fellow PhD colleagues Alex, Thomas, and Kumar who have been there since the start and were always available for "quick" coffee breaks.

When I started this journey, I was under the misconception that I will make this journey on my own. I, of course, expected the help of many others, but my expectation was that their help will come in waves at different times and with varying lengths (sometimes lasting hours, days or even weeks). However, this was far from being true. I have come to realize that I embarked on and now ended the journey alongside someone else, who has been there throughout my PhD. Someone who did not officially sign up for this journey but who also had to deal with some of its consequences. I probably don’t have the words to express my appreciation for my partner, Iris. She has always been their during the best and, more importantly, the worst times during this roller coaster of a journey. One soon realizes that this PhD does not only control my mood, energy, motivation and sleep, but to some extent also controlled hers. She played an important role in continuously helping me boost my confidence, talking me out of giving up and most importantly assuring me that it will all be alright in the end, regardless of the outcome. Being such an active participant of this journey, maybe the end is far better for her than it is for me. I would also like to thank her parents for giving me a home away from home.

The list of other important people in my life who I would like to thank will quickly become very exhaustive, so instead I will just offer a big thank you to everyone who has played even the slightest role over the past years. Each and everyone of you have contributed in some or other way to bring me to the finish line.

Abbreviations

ACE	Adaptive Calibration Error
BN	Batch Normalization
BNN	Bayesian Neural Network
Chap.	Chapter
dECE	Equal Distance (binning) ECE
DL	Deep Learning
DNN	Deep Neural Networks
ECE	Expected Calibration Error
Env1	Environment 1
Env2	Environment 2
Env3	Environment 3
FOV	Field-of-View
Fig.	Figure
GAN	Generative Adversarial Networks
GP	Gaussian Process
HB	Histogram Binning
HDE	Histogram Density Estimation
I-Max	Mutual Information Maximiation binning
iECE	I-Max (binning) ECE
JSD	Jensen-Shannon divergence
kECE	K-means (binning) ECE
KDE	Kernel Density Estimation
KLD	Kullback-Leibler divergence
mECE	Equal Mass (binning) ECE
NN	Neural Networks
RCS	Radar cross-section

ROI	Region-of-Interest
$_{\mathrm{top1}}$ECE	Top-1 ECE
$_{\mathrm{CW}}$ECE	Class-wise ECE
SAR	Synthetic-aperture radar
Scal.	Scaling
Mtx. Scal.	Matrix Scaling
w. L2	with L2 Regularization
MMC	Mean Maximal Confidence
MMC$_{\mathrm{inc}}$	Mean Maximal Confidence of Incorrect Samples
NLL	Negative Log-Likelihood
OMADA	On-Manifold Adversarial Data Augmentation
OvR	One-vs-Rest
ReLU	Rectified Linear Unit
Sec.	Section
sCW	Shared Class-wise Strategy
Tab.	Table
TS	Temperature Scaling
VAE	Variational Autoencoder

Contents

Chapter 1.

Introduction

1.1. Motivation

Ever since artificial intelligence (AI) has become a subject of discussion among the media, politicians, public figures and researchers, *automated driving* has received an influx of interest to improve a core part of modern society: transportation. With many relying on personally driving and/or using public transport (e.g. taxis, buses and car-sharing) for performing day-to-day activities, the increase in active driving participants on the roads has amplified the negative effects on the driving individuals and the environment. These negative consequences include increased delays and stress caused by traffic jams, environmental burdens caused by air pollution, as well as high death and injury rates caused by reckless and careless driving accidents. An autonomous agent with increased and reliable perception capabilities, which is also able to understand the environment through the automotive sensor measurements, could offer a reliable solution to many of these concerns.

1.1.1. Automated driving and multi-modal perception

Automated driving has the potential to greatly remedy many traffic-related problems currently faced by both personal driving and using public transportation. The largest influence would be the improvement of the safety of traffic participants when replacing the erroneous human factor (e.g. poor awareness, distraction and drowsiness) by an intelligent system with the potential to better perceive the environment. Allowing a machine to make driving decisions additionally optimizes the driving styles of all vehicles, ultimately reducing the release of toxic substances into the environment. Efficient and regulated automated driving styles can also reduce traffic jams (e.g. all cars can depart synchronously at a traffic light). Additionally, better perception can accelerate the search for parking spots which also contributes a great amount to the urban traffic problem. Moreover, the efficiency of autonomous traffic is expected to have a large benefit in improving the reliability and convenience of public transport, and as a result encouraging more drivers to instead seek the usage of public transport solutions. Altogether, the efficacy of autonomy in driving has the potential to greatly save many human lives; the World Health Organization has attributed approximately 1.35 million deaths, annually, to road traffic accidents[1] with 94% of all road accidents involving human error [Singh, 2015]. However, recent fatalities

[1] https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries/

caused by autonomous vehicles (from Tesla[2] and Uber[3]) have raised doubts about this potential and concerns over the ethics of autonomous vehicles [Holstein et al., 2018].

In order to propitiously realize automated driving, a crucial component is the ability to cognize the environment. Perception of the environment requires the detection and classification of relevant objects by relying on a fleet of sensors for a semantic understanding. For automated driving, this fleet commonly includes sensor modalities such as cameras, lidar (light detection and ranging), ultrasonic and radar (radio detection and ranging) sensors, as well as multiple sensors of the same modality depending on the sensor type (see Fig. 1.1). These various sensor modalities allow the ability to detect and measure different characteristics of the objects for a diverse representation of the state of the environment. For the detected objects in the scene, cameras provide cues to object color and sizes; lidar, ultrasonic and radar sensors offer estimates of the object distance, with radar sensors additionally offering velocity estimates and providing cues to the object material and reflectivity. Unlike human-driven vehicles, which heavily rely on vision and to a much lesser extent on acoustic signals, autonomous vehicles base their decisions on the information rich measurements from these multi-modal sensors and, as a result, have the potential to offer robust and reliable decision-making.

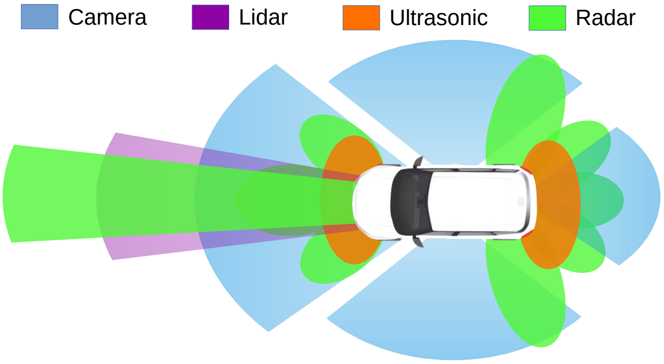


Figure 1.1.: An example of a fleet of various sensors equipped on vehicles and the area perceived by each sensor. Cameras, which are excellent for classification, are used for tasks such as traffic sign recognition, lane detection and lane departure warning, Lidar sensors are used for emergency brake detection and collision avoidance, though are among the most expensive sensors and, as a result, vehicles are equipped with very few or often with no lidar sensors. Ultrasonic sensors, which are cheap but only operate in the short-range and slow changing environments, are often limited to park assistance tasks. Owing to their affordability and reliability, a large network of radar sensors (long-/mid-/short-range) are often used for driving tasks such as blind spot detection, rear collision warning, and adaptive cruise control.

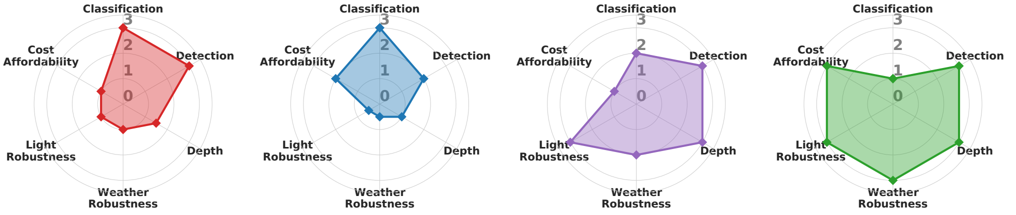
[2] Tesla driver killed after smashing into truck had just enabled autopilot. 2019 (https://www.theregister.com/2019/05/17/tesla_autopilot_crash/)

[3] Uber's self-driving car sensors ignored cyclist in fatal accident. Gizmodo, 2018 (https://gizmodo.com/report-ubers-self-driving-car-sensors-ignored-cyclist-1825832504)

Given the distinct nature in which these multi-modal sensors operate (i.e. cameras passively collect light waves, whereas lidar, ultrasonic and radar actively transmit/receive light, sound and radio waves, respectively), they are sensitive to disparate changes in the environment and offer varying degrees of robustness to these changes. Cameras are less sensitive to minor and possibly irrelevant changes such as viewing angles, though can be unreliable sources of information under brightness or weather changes [Shizhe Zang, 2019]. Lidar sensors are less sensitive to natural environmental changes, though are often expensive and bulky components which are hard to mount and conceal [Bagloee et al., 2016]. Perceiving the environment through dense point-cloud measurements, these sensors (similar to high-resolution cameras) demand large compute power to process and save the measurements and some lidar sensors are also more vulnerable to damages given their mechanical nature (e.g. moving parts) during operation. For short distances, ultrasonic sensors offer cheap perception solutions but are limited to environments with few and slow changes, and as a result are predominantly only utilized for object presence detection for parking assistance [Wang et al., 2014]. In contrast, radar sensors use electromagnetic waves for perception of the environment up to large distances and are much more robust to naturally occurring environmental changes (e.g. weather and sun glare) [Yoneda et al., 2019a]. For increased reliability and redundancy in the case of individual sensor failures, autonomous vehicles are often equipped with multiple sensors of the same modality. This often becomes impractical for expensive sensors such as lidar and some high-resolution cameras, and therefore cheap alternatives such as radar have the potential for offering increased reliability through redundancy.

For easier comparison, we visualize the characteristics of the different sensor modalities in Fig. 1.2, where the advantages and disadvantages become clear. It is clear that there is no single sensor providing all information and the desired reliability needed for the level of autonomy required by automated driving and therefore the fusion of information from multiple sensors is vital. The best manner to fuse the information is still an open problem, which relies on the accurate processing of the measurements from *each* sensor.

Despite the ultimate goal of automated driving aiming to employ a single decision-making system able to solve *all* driving-related tasks using *all* sensors, many tasks are currently still performed by individual systems and sensors. Ultrasonic sensors are often only used for park assistance; cameras are most reliable for recognition tasks such as traffic sign recognition or object type classification; adaptive cruise control is only possible using long-range radar sensors; and lidar sensors are useful for reliable collision avoidance and detecting emergency braking situations. Even though there has been an (informal) consensus on pairing specific driving-related tasks with sensors for reliable performance, recent years have seen a shift in this convention. For example, automotive cameras have been exploited for depth estimation [Godard et al., 2019], lidar sensors for traffic sign detection and classification [Weng et al., 2016], ultrasonic sensors for road surface detection [Bystrov et al., 2016] and pedestrian classification using radar [Prophet et al., 2018]. This change has been strongly motivated to improve the overall reliability for all tasks by ensuring reliable information streams even in the presence of measurement difficulties (e.g. using a camera at night or in bad weather conditions) or sensor malfunctions. For example, object type classification has been performed using lidar and radar, and stereo cameras have been used for depth estimation.



(a) Human Driver **(b)** Camera **(c)** Lidar **(d)** Radar

Figure 1.2.: Comparison between the characteristics of human drivers and various sensor modalities in the context of automated driving. Despite all the benefits of the radar sensor, classification performance has not reached the desirable performance. This thesis focuses on improving the object classification abilities using a radar sensor.

1.1.2. Radar classification and deep learning

An important step in acquiring a semantic understanding of the environment is the accurate classification of the objects in the scene. Even though all these sensor modalities have been exploited for object classification, cameras have been, by far, the most successful. As one of the most human interpretable sensors, this success can be mostly attributed to the algorithmic developments which are often inspired by the pattern recognition abilities of the human brain (e.g. exploiting object edges, or mimicking human robustness to slight changes through data augmentations). Furthermore, images have a higher information content (e.g. shape, color, and texture) than radar or even lidar reflections. We notice that despite the many benefits of radar sensors which offer distance and velocity estimates, are robust to weather and lighting changes, and are cheap enough to offer perception through a network of radar sensors, the literature for radar classification is still at an relatively early stage and, thus far, limited to simulated and/or small datasets (e.g. often 2 or 3 classes). In order to maximally leverage the capabilities of the radar sensor for automated driving perception, radar-based multi-class object classification is a vital problem which is not sufficiently solved. A major focus of this thesis is improving the reliability of driving perception through the accurate and reliable classification of objects using radar.

The challenging task of object classification in a dynamic environment not only relies on reliably perceiving the environment, but also, to a larger extend, on categorizing the measurements into classes. Therefore, in order to successfully imitate human driving, another important and useful human trait to simulate is *learning*. The field of machine learning achieves this through the help of humans in the form of data annotations, and specifically the task of classification through large corpora of *labelled* data. Analytically solving this task from data is not possible for real-world scenarios with large input spaces (e.g. pixel space of images), and instead require high-complexity modeling. The rapidly changing algorithmic landscape of learning algorithms has made it possible to learn complex features present in the data with limited intervention from humans. A specific set of learning algorithms which have become ubiquitous to classification, called deep learning, have immensely contributed in the ability of classifiers to obtain high performance on difficult tasks. They easily surpass previous non-neural-network based algorithms and in many cases also human performance.

1.1.3. Uncertainty estimation

While the accuracy of the classifiers can always be improved by learning from more data, it is infeasible to cover all possible situations encountered in the future. While no guarantees can be given that the correctness of the predictions will always remain perfect, their uncertainty estimates can provide cues of unknown situations. Detecting such situations allows human intervention for handling such situations to avoid mis-guided decision-making. To ensure trustworthy predictions, it is crucial that these classifier predictions reflect when it encounters such unknown situations for which it is forced to extrapolate its knowledge and emphasize that its outcomes are uncertain.

In contrast, deep learning classifiers are characterised as black-box systems which provide no reasoning behind their decisions or even make clear the exact computations on the input. Despite these classifiers proving to be highly-accurate, this black-box nature has led to the identification of several unexpected and notorious properties which include the classifiers inability to identify situations when they *do not know*. Such classifier attributes can often lead to catastrophic consequences in safety-critical applications due to their inability to output predictions from which decision-making systems are able to determine their correctness. Given the high modeling complexity of these classifiers, *all* predictions are often *over-confident* regardless of the correctness of the prediction or the noise or uncertainty inherent in the input. To amplify this effect, driving situations are often encountered with many unexpected scenarios and points of failure, therefore such predictions do not allow meaningful decision-making. Anomaly/outlier detection [Pang et al., 2020] can be performed, in parallel to classification, to detect such unexpected scenarios and allow meaningful abstention, though this can at times be computationally expensive and infeasible for most applications. Therefore, anomaly/outlier detection is often based on the predicted confidences of the classification network [Hendrycks & Gimpel, 2017; Lee et al., 2018], which are often severely over-confident and, as a result, negatively affect the anomaly/outlier detection performance. Altogether, these factors hinder the understanding of deep learning classifiers and more importantly does not allow the detection of incorrect predictions as the models themselves do not often know this.

Multiple factors can be attributed to the over-confidence of deep learning classifiers which include high modeling capacities, limited training datasets, one-hot hard labels which encourage maximally confident predictions and loss functions which are optimal when the predicted distributions match the over-confident ground truth hard labels. During the training of the classifier, the predictions become increasingly accurate but at the same time become increasingly over-confident.

To overcome these challenges, the field of uncertainty quantification has taken great interest in teaching these classifiers to better reflect uncertainty estimates in their predictions [Guo et al., 2017; Gal & Ghahramani, 2016; Thulasidasan et al., 2019], as well as understand the reasons behind the over-confidence [Nguyen et al., 2015; Hein et al., 2019] phenomena. These developments are important to ensure the applicability of deep learning classifiers for safety-critical applications such as automated driving. One could claim that regardless of further algorithmic developments to improve the accuracy of classification, without reliable uncertainty estimates the highly-accurate classifiers have limited use. Therefore, addressing the over-confidence issues of these complex classifiers is an important open research problem.

1.2. Contributions and publications

In this thesis, we contribute towards improving the reliability of automated driving through two aspects. First, we aim to develop highly-accurate object classifiers for radar sensors to ensure that classification can *also* be based on a more robust sensor able to deal with situations where other sensors often fail (see Part I). For this study we create a novel real-world automotive radar spectra dataset consisting of multiple environments and measurements from a static scene (see Chap. 4) and use radar-specific knowledge to aid the learning process for larger performance gains (see Chap. 5). Secondly, given the importance of high-quality predictive uncertainty estimates for *all* classifiers used for a safety-critical task such as automated driving, we directly address the over-confidence issue of deep learning classifiers (see Part II, specifically Chap. 7 and 8). These general real-time solutions, which are agnostic to the data modality, re-calibrate the classifier predictions such that the output uncertainties are correlated with the difficulty in determining a prediction for the samples. For Part II, the studies focus on image classifiers but generalize to any classifier. Compared to the vision classifiers analyzed in Part II, the radar classifiers, which are very susceptible to noise and are sensitive in their measurements, tend to suffer more from the problem of prediction over-confidence (see Part. III). As a result, even though the radar classifiers from Chap. 5 offer highly-accurate predictions, solutions similar to those introduced in Part II are essential for increased reliability. Improving the uncertainty estimates of these radar classifiers using solutions introduced in Chap. 7, we improve the reliability of their predictions (see Chap. 10). Additionally, we yield further performance gains in terms of accuracy and uncertainty estimation quality by proposing a novel solution for radar classifiers which utilizes radar-specific knowledge to implicitly improve the reliability of the predictions (see Chap. 11). Together with Parts I, II and III, we show that solving open problems in the literature (see Chap. 2, Sec. 2.4.2), takes a big step towards improving the trust between the classifier predictions and decision-making systems, as well as offering competitive classification through robust and cheap radar sensors which are able to measure novel object characteristics.

The core content of this dissertation is based on the following research publications (for which the first author was the primary contributor):

1. **Patel, Kanil**, Kilian Rambach, Tristan Visentin, Daniel Rusev, Michael Pfeiffer, and Bin Yang. "Deep Learning-based Object Classification on Automotive Radar Spectra." In IEEE Radar Conference (RadarConf), 2019. (see Chap. 4 and 5) [Patel et al., 2019]

2. **Patel, Kanil**, William Beluch, Dan Zhang, Michael Pfeiffer, and Bin Yang. "On-manifold Adversarial Data Augmentation Improves Uncertainty Calibration." In International Conference on Pattern Recognition (ICPR), 2020. (see Chap. 8) [Patel et al., 2020]

3. **Patel, Kanil**, William Beluch, Bin Yang, Michael Pfeiffer, and Dan Zhang. "Multi-Class Uncertainty Calibration via Mutual Information Maximization-based Binning." In International Conference on Learning Representations (ICLR), 2021. (see Chap. 6 and 7) [Patel et al., 2021b]

4. **Patel, Kanil**, William Beluch, Kilian Rambach, Adriana-Eliza Cozma, Michael Pfeiffer, Bin Yang. "Investigation of Uncertainty of Deep Learning-based Object Classification on Radar Spectra." In IEEE Radar Conference (RadarConf), 2021. (see Chap. 10) [Patel et al., 2021a]

5. **Patel, Kanil**, William Beluch, Kilian Rambach, Michael Pfeiffer, Bin Yang. "P-smoothing: Improving Uncertainty Calibration of Automotive Radar Spectra Classifiers with Soft Labels" In IEEE Radar Conference (RadarConf), 2022. (see Chap. 11)

The first author of *all* publications has had the largest contribution in these publications from being a big part of the initial ideas and discussions to defining and running the experiments, as well as leading the writing process. The help of the co-authors include the processing of the radar dataset (Kilian Rambach, Adriana-Eliza Cozma, Daniel Rusev and Tristan Visentin); creation of figures (William Beluch); discussion of initial paper ideas (Michael Pfeiffer and Dan Zhang); and finally writing and proof reading the papers (Michael Pfeiffer, William Beluch, Dan Zhang, Kilian Rambach, and Bin Yang). The thesis would also not be possible without their valuable contribution in achieving these publications.

1.3. Organization of thesis

We provide a more detailed introduction into the main topics of this dissertation in Chap. 2, as well as strengthen the link between these topics. The thesis is further split into the following four parts: (1) Radar spectra multi-class classification using deep learning, (2) Uncertainty calibration in deep learning, (3) Radar uncertainty calibration, and (4) Conclusion, discussion and outlook.

In Part I, the problem formulation and literature review of radar-based deep learning methods is provided in Chap. 3. Details of the construction of the real-world radar datasets are given in Chap. 4 and in Chap. 5 we propose a novel solution for radar spectra classification using radar-specific prior knowledge for increased generalization performance.

Part II begins with a problem formulation and literature review of uncertainty calibration of deep learning classifiers in Chap. 6. In Chap. 7, we propose a real-time post-hoc calibration technique to improve the uncertainty estimation quality using a novel histogram binning method which is learned using a mutual information based optimization. A novel generative modeling-based data augmentation technique is presented in Chap. 8 which aims to improve uncertainty estimation quality by synthesizing ambiguous samples.

After the problem formulation and literature review of the theme *uncertainty of radar classifiers* in Chap. 9, Part III continues with Chap. 10, which is the first ever study of the uncertainty calibration of radar classifiers and application of uncertainty methods from Chap. 7. The part is ended off with Chap. 11 which offers a novel label smoothing technique for improving generalization and uncertainty estimation of radar classifiers.

Finally, this work is concluded in Part IV with a discussion and final remarks on future work.

Chapter 2.

Automotive radar perception, deep learning and uncertainty

In this chapter we provide a more in-depth introduction into the topics addressed in this thesis. The main themes discussing the current status on classification using a radar sensor, uncertainty estimates of deep learning-based classifiers and the connection between these two topics.

2.1. Radar-based object classification

Historically, radar sensors have been exploited for *detection of objects* (e.g. airplanes [Khan & Power, 1995], ships [Schuster et al., 2014], and weather formations through precipitation detection [Saltikoff et al., 01 Sep. 2019]). As a result, in the context of automated driving, its research and application was initially limited to advanced assistance driving systems (ADAS). Being an important part of the ADAS landscape, radars offered reliable perception for performing tasks such as blind spot detection [Liu et al., 2017], obstacle detection and collision avoidance [Grosch, 1995], and parking assistance [Shi & Wang, 2010]. Despite the acronym for radar meaning detection and localization of objects using radio waves, recent years have seen a surge in exploiting radar sensors for the task of classification [Yeo et al., 2016; Roodaki et al., 2011]. Similar to light waves, used by vision sensors, which react differently based on the material of the objects (i.e. reflecting different colors), the electromagnetic radio waves transmitted and received by the radar sensors are also sensitive to object-specific properties which include the object materials, roughness of the surface, reflectivity and the size of the objects. Even though these cues are helpful for learning discriminative object features, the sensitivity of the sensors to these factors can, at times, post a great challenge for classification. The object size, reflectivity and surface roughness (i.e. attributes of the object) also control the amount of information measured at the sensor and could possibly hinder accurate classification for objects which can, at times, be invisible to the radar sensor (e.g. small plastic object or very distant objects).

Another contributing factor for challenges in radar-based classification stems often from performing classification at a rather late phase during the signal processing pipeline. In a similar fashion to lidar classifiers, early works for radar-based classification used occupancy grids and point-cloud representations constructed from the detected reflexes[1] [Dubé et al., 2014; Lombacher et al., 2015, 2017; Schumann et al., 2018; Feng et al., 2019; Schumann et al., 2018]. This representation is

[1] Reflexes refers to the points (range and azimuth coordinates) resulting from the received signal after a detection algorithm is applied.

specifically susceptible to the lack of some objects to reflect back significant signals for reliable classification of the resulting point cloud. As a result, despite showing some initial success, radar-based classification was limited to small number of classes, controlled laboratory setting and/or simulated datasets [Oezcan et al., 2016; Schubert et al., 2013; Rohling et al., 2010; Heuel & Rohling, 2010, 2011, 2012; Molchanov et al., 2013].

In order to successfully scale classification to a real-word setting with multiple classes, we utilise a richer source of information in the radar signal processing pipeline: the radar spectra. The multi-dimensional FFT spectra has previously shown beneficial for tasks such as pedestrian detection [Bartsch et al., 2012], human fall detection [Jokanovic & Amin, 2018], human pose estimation [Zhao et al., 2018a,b] and human-robot classification [Abdulatif et al., 2018]. Compared to the point-cloud radar reflections or occupancy grids, the spectra contains a loss-less representation of the received signal which we aim to exploit for object type classification.

At this point, we pose the following research question (RQ) and answering this, as well as the upcoming ones, will be a reoccurring theme throughout this thesis:

RQ1	*How can automated driving be ameliorated for reliable and robust sensor perception?*

2.2. Deep learning

Even though these object-specific spectra characteristics are difficult, if not impossible, for humans to recognize, the complex patterns can be learned from the data. Unlike early works which required hand crafted features (e.g. approximation of the car shape by an L-shape of the resulting point-cloud [Keat et al., 2005]; discussed in more detail in Sec. 3.3), learning algorithms can discover these complex discriminative features in the data which are best suited for classification. A set of learning algorithms, called deep learning [LeCun et al., 2015], have immensely contributed to surpassing human level intelligence for classification by mimicking the human brain's ability to learn. These neural network based algorithms, which learn data representations as a nested hierarchy of concepts, has become a common tool for processing and analysing large corpora of data to perform challenging tasks such as 1000-class classification [Russakovsky et al., 2015], automatic machine translation to multiple languages [Zhang & Zong, 2015], and automatic game playing [Silver et al., 2016]; tasks which are hard or even impossible for most humans.

Exploiting the large modelling capacities of these algorithms, we aim at learning complex discriminative radar spectra features for accurate multi-class object classification. Given that (deep learning) algorithmic developments are mostly inspired by vision (i.e. research has a large focus on images/videos), we also aim at utilising radar-specific knowledge to aid the learning process to address novel challenges which arise from the direct application of deep learning to the radar spectra modality.

2.3. Importance of uncertainty quantification

Even though these deep learning algorithms enjoy success in a range of applications, the reliability of their predictions have become a constant criticism, especially in new environments previously unseen by the algorithm.

RQ2	*Is optimizing the generalization performance of classification models enough for safety-critical applications?*

Apart from deep learning algorithms not being robust *enough* for real-world applications, they also fail to realize their own limitations in new environments (i.e. they do not know when they do not know). This has been shown in the context of fooling images [Nguyen et al., 2015], single pixel attacks [Su et al., 2019] and hate-speech detectors [Grondahl et al., 2018]. As a result, the lack of abstention for such situations limits their use for safety-critical applications which can lead to catastrophic consequences such as loss of life. To amplify this problem, these algorithms are susceptible to exploiting unwanted dataset biases [Tommasi et al., 2017; Torralba & Efros, 2011; Kim et al., 2019] (e.g. environment setup or background), as well as unwanted human biases [Larrazabal et al., 2020; Zou & Schiebinger, 2018; Hutson et al., 2017] (e.g. race, gender or religion) captured in the data [2], and as a result are often encountered with unfamiliar situations in the real-world when these biases are not present anymore. Most of these negative characteristics of deep learning classifiers (which learn features not interpretable by humans but are still highly accurate) can be attributed to its black-box nature which poses a great difficulty in finding explanations behind their decisions. Solving these issues has been a large focus of the community by studying and improving the robustness, explainability and uncertainty of deep learning classifiers. We focus on improving the uncertainty estimation of the classifiers which aims at teaching the classifiers to reflect uncertainty in challenging situations when the classifier predictions become unreliable.

RQ3	*How can classifiers learn meticulous real-time uncertainty estimation?*

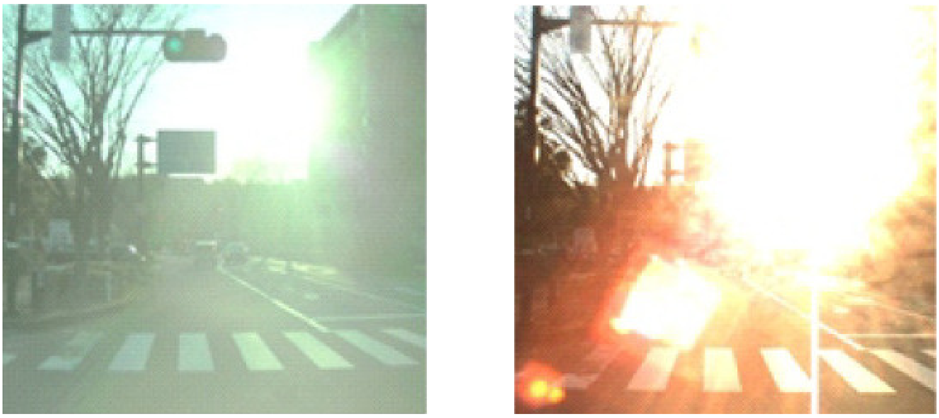
Deep learning classifiers also exhibit the characteristic of being poorly calibrated [Guo et al., 2017] and mostly over-confident [Hein et al., 2019] in their predicted confidences. The reasons behind this notorious property include large model complexity and over-fitting [Guo et al., 2017; Pereyra et al., 2017], hard labels [Szegedy et al., 2016; Patel et al., 2020] (see Sec. 11.3.1), and model architecture [Hein et al., 2019]. The over-confidence is often evident in the incorrect assignment of high confidences for mis-classified samples [Guo et al., 2017], as well as unfamiliar and unknown input distributions [Hendrycks & Gimpel, 2017]. The predictions from such classifiers, albeit being highly accurate, have limited use in safety critical applications as it becomes difficult to distinguish between highly-confident correct and over-confident incorrect classifications. Such classifiers cannot reliably be used, as decision-making systems can be misguided into making wrong conclusions. An example of the consequence of over-confident classifiers can be seen in Fig. 2.1. If the two images are both assigned the same high confidence, autonomous systems cannot determine which of the two predictions are trustworthy for further decision-making, even though it is clear that the image without any sunlight glare should be used because the other prediction was a mere guess. Over-confidently providing a *wrong* prediction about the status of a traffic light, could lead to an unnecessary braking maneuver (endangering

[2] https://www.scientificamerican.com/article/how-a-machine-learns-prejudice/

the vehicles following the autonomous agent) or acceleration (leading to fatal collisions at an intersection).

Such situations are more often encountered during radar-based classification, where the radar spectra, even without any exterior interventions, are ambiguous for small objects with low reflectivity. Therefore, the problem of incorrectly assigning high confidences for radar predictions can amplify the catastrophic consequences as they can occur much more frequently. Specifically, for radar-based classification we also pose the following question concerning these issues which are predominantly affect radar classifiers:

RQ4	*Can radar specific knowledge aid data-driven techniques?*



(a) **(b)**

Figure 2.1.: Source: [Yoneda et al., 2019b]. An image of the same automated driving scene using two different camera types. The left image uses a high dynamic range (HDR) camera and the right image does not [Yoneda et al., 2019b]. Camera image (b) produces a highly distorted image of the traffic light, due to the direct sunlight, and as a result preventing any *accurate* classification of the traffic light status. For such an image, any classifier would have to resort to random guessing for predicting the traffic light status. The status of the traffic light of the same scene can confidently be determined using another sensor (e.g. the sensor producing the left image) which does not suffer from the bad lighting conditions. It is clear that a decision-making system should solely base its classification on camera image (a), however, if the classification of *both* images are associated with high over-confident predictions, it becomes difficult to determine which high confident prediction is trustworthy.

To amplify the negative consequences of over-confident classifiers, it becomes unenviable to compare the predictions from different classifiers for the same and other sensor modalities. One goal of using multiple modalities is to increase the perception reliability, as some sensors are affected differently by the dynamic environment. For example, vision sensors should be less trustworthy during poor visibility such as lack of light or bad weather conditions, whereas radar sensors provide noisy measurements for heavily cluttered scenes and have lower resolutions. Without the classifiers correctly reflecting the predictive uncertainty, it becomes difficult for a decision-making system to determine which classifiers' predictions are trustworthy or how the information from the different sensors should be handled. This poses as a conspicuous limitation as the initial justification of using multiple sensors (of the same and different modalities) is to increase the perception reliability of the environment, by falling back on sensors which are more robust in some situations than others. Even in the presence of faulty or unreliable sensor measurements, over-confident classifiers do not allow meaningful abstention which should be practised for these situations.

Therefore, to effectuate automated driving, it is important to have an accurate classification from *each* sensor, as well as high quality uncertainty estimates to allow predictions from multiple sensor classifiers to be fused in a meaningful way. The focus of this thesis can be broken up into two parts, where one aims to improve the classification performance of automotive radar sensors and the other provides general (real-time) techniques to improve the predictive uncertainty quality of any classifier. The former goal aims to motivate reliable object classification using radar spectra, which has the potential to greatly improve the perception reliability of an automated driving system. The latter goal aims to address the problem of over-confidence which poses great difficulty in comparing predictions from multiple sensors, and ultimately limiting the decision-making systems ability to leverage information from multiple sources.

2.4. Questions addressed and literature gaps

2.4.1. Research questions

In order to summarize the motivation and contributions of this thesis, we list the following research questions (RQ), posed thus far, again and aim to answer these throughout the chapters of this thesis.

RQ1 *How can automated driving be ameliorated for reliable and robust sensor perception?*

RQ2 *Is optimizing the generalization performance of classification models enough for safety-critical applications?*

RQ3 *How can classifiers learn meticulous real-time uncertainty estimation?*

RQ4 *Can radar specific knowledge aid data-driven techniques?*

2.4.2. Literature gaps

In this section, we identify several literature gaps, which we take to be areas of scientific research which are unexplored or often times under-explored by existing studies within the field. These allow the contributions of the this thesis to be put into the context of the literature, as well as highlight limitations of current state-of-the-art. We identify and list below three such literature gaps which motivate the research projects of this thesis. A thorough literature review is given in Chapters 3, 6 and 9, and here we only give a brief overview of the literature in the context of each literature gap. We note that we do not entirely solve these literature gaps in this thesis, though we do explore ideas which can lead to further research developments for solving these gaps.

LG1 *Radar object classification*

Accurate classification of objects using the radar sensor is currently a relatively underexplored field of research, with most current research limited to few-class, controlled laboratory setting and/or simulated datasets. We hypothesize that these limitations are a result of classification not being performed using a more information rich representation. Instead, currently classification is predominantly based on the point-cloud representation [Schumann et al., 2018], which is a sparse information representation. The vast literature which exploits the more informative radar spectra has shown success in many tasks which leverage the micro-Doppler signature in dynamic scenes, though spectra-based multi-class classification of a realistic static scene has not yet been thoroughly explored. Requiring classification models to learn features other than the micro-Doppler signature of the objects (which is often very diverse for different objects in dynamic scenes) pose a great learning challenge and we have identified a gap in the literature with a lack of solutions to sufficiently overcome this problem. Currently, no learning algorithms are used to study the reflected signals measured in the radar spectra for static scenes. These algorithms require learning complex discriminative features which study the signals for determining the object type through studying the complex interaction of the signals to the object material, surface roughness, distance and viewing angles. In an attempt to address this literature gap, recently, similar work has been done in parallel to this thesis in [Sheeny et al., 2020]. Similar to the work presented here, those authors also performed radar spectra-based multi-class classification on a static scene, but were limited to a laboratory setting with dummy objects such as a mannequin and stuffed dog and a static radar sensor. A more thorough literature review of learning algorithms for radar is given in Sec. 3.3.

LG2 *Reliable evaluation of radar classifiers*

Generalization performance of radar classifiers is an important performance indicator for the classifier's ability to support automated driving systems in novel and unseen situations. Given the finite training and test sets, ensuring perfect generalization performance to *all* situations is not possible and instead it becomes important to ensure that the classifiers reflect uncertainty in such situations when it is extrapolating its knowledge. Therefore, evaluations and metrics which *only* measure the generalization performance of radar classifiers is not enough to ensure appropriate behaviours when encountering novel situations. Especially, when deep learning, which is becoming ubiquitous for radar classification, predictions are associated with being over-confident [Hein et al., 2019] and mis-calibrated [Guo et al., 2017]. Despite the importance of using performance indicators beyond generalization (i.e. accuracy), the radar classification literature does not employ any mechanism to thoroughly evaluate their reliability and robustness. To the best of our knowledge, evaluations are solely done based on the ability of the classifier to accurately predict a *finite* test set and minimal attention is given to other metrics or other situations to ensure safe, reliable and robust operation in practise. In addition to improving the generalization performance of radar classifiers (i.e. LG1), Part I and Part III also focuses on evaluating the uncertainty and robustness.

LG3 *Real-time uncertainty calibration*

Tackling the over-confidence [Hein et al., 2019] and mis-calibration [Guo et al., 2017] of deep learning classifiers requires solutions which improve the quality of the predictive uncertainty. Additionally, in order to ensure the applicability of these solutions for real-time applications, it is important that they offer low latency solutions. An effective strategy for improved uncertainty estimates, adopting Bayesian inference, uses Bayesian Neural Networks (BNNs) [Blundell et al., 2015] to estimate the models uncertainty in its predictions. Though they have shown to yield better confidence estimates, BNNs scale poorly to large parameter spaces and inference requires multiple forward passes at test time. The scalability issues of BNNs have been a focus of attention in the literature [Kristiadi et al., 2020; Lakshminarayanan et al., 2017; Gal & Ghahramani, 2016], though still require multiple forward passes. Therefore, Bayesian techniques are computationally and memory demanding solutions, as well as require model architecture changes and re-training of all classifiers for improved uncertainty estimation.

Post-hoc calibration techniques [Guo et al., 2017; Platt, 1999] offer simple solutions which do not require additional re-training and can be appended to existing, already trained, classifiers. Using the Bayesian perspective, Gaussian process (GP) [Wenger et al., 2020] calibration offers a highly expressive calibrator which offers both generalization and calibration improvements, though similar to other BNNs, require multiple passes and are also resource demanding at test time. In order to exploit the high modeling capacity of deep learning classifiers for uncertainty calibration, *during-training* calibration techniques [Thulasidasan et al., 2019; Mueller et al., 2019] implicitly improve the uncertainty calibration during the training process. Such techniques require re-training, though still offer real-time estimates at test time.

Despite the great interest in field of uncertainty quantification, effective real-time uncertainty calibration solutions are still lacking (i.e. solutions which are both computationally and memory efficient which add negligible additional computation time). In an attempt to fill this literature gap we offer both, a novel post-hoc and during-training calibration technique, with the former offering solutions for improving the uncertainty of classifiers which are already trained. With both offering real-time estimates and requiring negligible or no additional resource demands, the work addresses a large part of the literature gap. A more thorough literature review of uncertainty estimation of deep learning classifiers can be found in Sec. 6.6. Uncertainty estimation for radar classifiers is even less explored in the literature, especially leveraging radar specific knowledge for improving uncertainty estimates which has not been done before. We return to this topic in Part II.

Part I.

Radar spectra multi-class classification using deep learning

Chapter 3.

Problem formulation and literature review

3.1. Problem formulation

Reliable perception of autonomous vehicles relies on multiple sensors to obtain a reliable understanding of their environment and plan their actions accordingly. Simple localization of potential obstacles in the vehicle's path is insufficient; instead, a semantic understanding of the world in real-time is crucial to take into account possible reactions of identified traffic participants and to avoid unnecessary evasive/emergency brake maneuvers for harmless objects. At present, there is a strong focus on imaging sensors for scene understanding, because high-resolution color images contain substantial information that allow highly detailed and accurate object classification [Cordts et al., 2016]. However, vision is severely limited in difficult light or weather conditions, and automotive radar provides a particularly useful complementary source of information [Mukhtar et al., 2015].

The radar processing chains, typically, extract radar reflections and identify object classes by the shape of the resulting point-cloud of reflections belonging to the same object [Schubert et al., 2015]. This approach works well for large reflective objects such as cars, but distinguishing multiple object classes from small sparse point clouds has proven to be challenging. These small sparse point clouds are a result of the susceptibility of this representation to poor reflectivity of some objects. Poor reflectivity, which can also be indicative of the object class, can be caused by the object size, distance, viewing angle and surface roughness, as well as the material characteristics of the object. As a result, despite showing some initial success, radar-based classification was, at the time, limited to small number of classes and/or simulated datasets [Oezcan et al., 2016; Schubert et al., 2013; Rohling et al., 2010; Heuel & Rohling, 2010, 2011, 2012; Molchanov et al., 2013].

One reason for this difficulty can be attributed to the loss of a substantial amount of information characteristic for the object type at this late stage of the radar signal processing pipeline, the point-cloud representation. It is therefore advantageous to base radar classification on a more informative data representation, such as the multi-dimensional FFT radar spectrum which captures all signals reflected back to the sensor. Example spectra samples from seven distinct objects can be seen in Fig. 3.1 where, unlike camera images, the radar spectra are not easily interpretable by humans. As a result, creating accurate hand-crafted discriminative features for spectra has been challenging and therefore requires more robust solutions which can automatically learn these features directly from the data. An additional element of difficulty in spectra classification, which motivates an adaptive learning algorithm, stems from the sensitivity of the signals to minor viewing changes in observing the object.

In Fig. 3.2, we visualize a sequence of seven spectra samples measured during an approach of the sensor to an object (i.e. VW Golf car). Even though the sequence only lasts for a short 0.34 seconds, the sensor only gets 0.80m closer to the object, as well as the viewing angle only changes by a mere 0.46°, the measured signals can have large variations which further make the task of classification strenuous. This is not the case for cameras which often result in only minor changes in the image for time sequences of a couple of seconds. For radar sensors, there are often multiple paths a radar signal can travel before arriving back at the sensor (i.e. multipath reflections). For example, the reflected signal could directly go back to the sensor or first bounce of the ground before arriving at the sensor. Such multipath reflections can lead to signal interferences, and since the wavelength of the used radar is in the mm range, small changes in the distance can already lead to rapidly changing amplitudes. Moreover, small changes in the viewing angle of an object can lead to large radar cross-section (RCS) fluctuations, and result in different measured signal strengths.

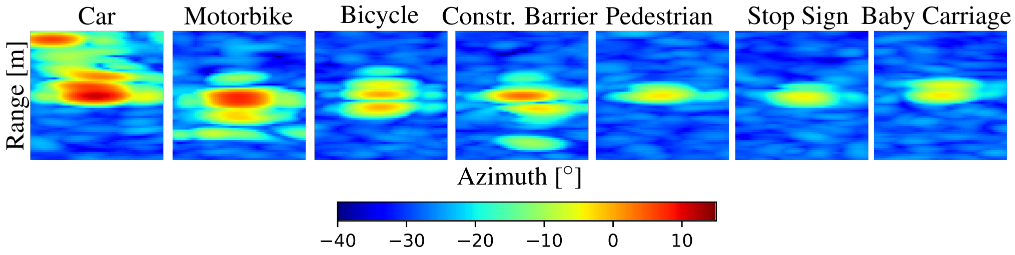


Figure 3.1.: The radar field-of-view (FOV) split up into cropped snippets (explained in more detail in Chap. 4) around the location of 7 objects each measured (in dB) from a 15m distance. Unlike in camera images (and for humans) where these objects (car, motorbike, barrier, bicycle, construction barrier, pedestrian, stop sign and baby carriage) are easily distinguishable, the spectra samples are harder to interpret, making the task of classification harder, even for humans. The challenge is even greater for small objects such as pedestrians, stop signs and baby carriages where the radar sensor measures very similar signals.

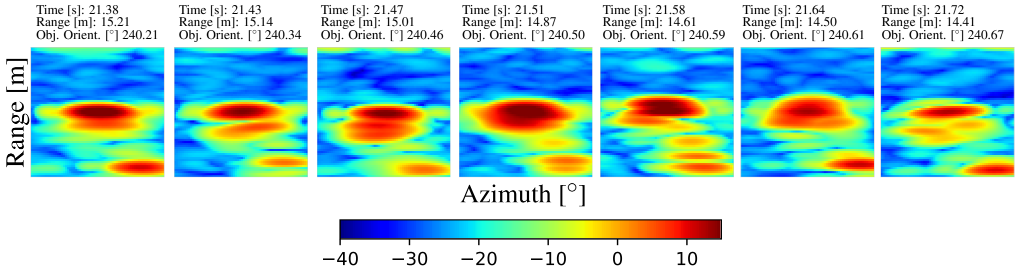


Figure 3.2.: Sequential spectra samples measured by approaching an object (car) for 0.34 seconds. Even though all seven samples of the car are measured less than half a second apart, the sensitivity of the radar sensor to the complex components of the object result in vastly distinct measurements. During this driving sequence of 0.34 seconds, the radar sensor has only moved towards the object by 0.80m and the viewing angle of the sensor relative to the object (i.e. object orientation) changes by at most 0.46°. This shows the great difficulty in creating hand-crafted discriminative features as the sensor measurements are sensitive to even minor changes in the viewing range and angle, in addition to the other object reflectivity characteristics.

Recent years have seen deep learning methods [LeCun et al., 2015] successfully solve such challenging tasks though learning algorithms which leverage complex features present in large copora of data. Computer vision has greatly benefited from these algorithms for object classification, object detection and semantic segmentation. In this part of the thesis, we aim to solve the task of multi-class radar spectra classification by utilizing algorithmic advancements in deep learning for automatically learning discriminative features which are beneficial for classification, as well as address other unexpected issues raised by the direct application of deep learning for radar spectra. The efficacy of such a study depends on available data measurements, therefore we construct a real-world automotive radar spectra dataset in Chap. 4. Using this dataset, we learn the first deep learning classifiers for radar classification in Chap. 5, where we also address some radar-specific challenges by proposing novel solutions which yield improved generalization performance.

3.2. Challenges of deep learning for radar spectra classification

In this section, we describe and discuss some challenges which need to be considered before the direct application of deep learning to radar. These include challenges which are also faced by other domains exploiting deep learning for classification, but which especially become worse for radar, as well as some radar specific challenges to be aware of.

Deep learning algorithms have large modelling capacities, capable of learning complex features from data for classification tasks which are difficult even for humans [Russakovsky et al., 2015]. These algorithms, which are great at memorization, have shown the ability to erroneously reproduce *even* noisy and random labels [Arpit et al., 2017], as well as shown to be susceptible to over-fitting large datasets. As a result, the vision community has proposed multiple ways to address these challenges to prevent such memorization or over-fitting in the form of larger datasets [Russakovsky et al., 2015], data augmentation [Thulasidasan et al., 2019; Patel et al., 2020], and regularization [Srivastava et al., 2014; Ioffe & Szegedy, 2015].

We find that radar classification is especially vulnerable to these challenges of deep learning. Even though many of the algorithmic developments can be used for radar classification, we still find a number of these factors to pose great challenges. We list these challenges and briefly discuss them in comparison to vision datasets:

1. **Dataset size**: Given the expensive nature of measuring and labelling radar spectra datasets, the over-fitting phenomenon of deep learning is amplified for radar classifiers. In addition to the lack of effective augmentation techniques to diversify the data, the high complexity models quickly overfit when learning in the low data regime. The biggest contrast to vision datasets, which are often magnitudes larger, is the difficulty in labelling the spectra due to their limited interpretability by even human experts. This is especially restrictive when considering a real-world setup with multiple objects in the field-of-view (FOV) of the sensor, where separating the reflections from each object becomes harder.

2. **Limited diversity in dataset to learn important features**: Large vision datasets are rich in diversity of the type of measurements which capture tens or hundreds of instances of the same object, multiple viewing angles, and many more environment setups. In the case of radar datasets, which are already much smaller in comparison, the diversity is also limited due to the difficulty in measuring and labeling the data in a real-world setting. The diversity of the training set is often limited to a single instance of each object, a handful of viewing

angles and a single environment setup. Additionally, data augmentation for radar spectra for diversification is still an open problem. The consequence of such limited diversity in the dataset is especially bad for radar classifiers, as the sensor is very sensitive to multiple properties of the object such as its material, shape, surface roughness, distance, and size. If small perturbations to these properties (e.g. measuring the same scene few microseconds apart) can result in large changes to the measured spectra, then it is natural to expect that the classifiers would easily encounter situations in the real-world which greatly differ from the training distribution. It is thus important that the training datasets are diverse enough for the classifiers to generalize to unseen settings.

3. **Large modelling capacities do not translate to large performance gains**: Deep learning classifiers for vision usually have complex, deep and large model architectures allowing them to obtain human-level generalization performance. Even though large model architectures are more susceptible to memorization and over-fitting, using more data, data augmentation [Thulasidasan et al., 2019; Patel et al., 2020], and regularization [Srivastava et al., 2014; Ioffe & Szegedy, 2015] have allowed these factors to have limited effect. Despite many of solutions from vision can be borrowed for radar, not all options are able to minimize the effect of these factors and improve generalization performance. We find that relatively small, LeNet [LeCun et al., 2015]-type, model architectures are best suited for radar spectra classifiers. Increasing the model capacity only results in greater levels of over-fitting and memorization, which do not translate to generalization performance. Even though large datasets could partially fix this over-fitting, the diversity of the data is also just as important. Though, both are limited in radar datasets.

4. **Dataset imbalance**: A common problem in vision is the class imbalance in datasets, where the data can contain more classes of some objects than others. This problem is amplified for radar datasets which, in addition to class imbalance, have an additional factor of imbalance. Due to the nature of how radar sensors perceive the objects, some objects are more reflective than others, and as a result these objects are detected, and the region-of-interests (ROIs) are extracted, much more often than other smaller less-reflective objects. This class imbalance is made worse by the fact that the measured signal degrades with increasing distance, and as a result, larger objects are more detectable than smaller objects at these larger distances. Additionally, given that the field-of-view (FOV) of the radar sensor increases with distance (due to the sensor operating in polar coordinates), a single frame captures more objects which are farther away in the FOV. As a result, the dataset is heavily biased towards ROIs from objects which are extracted from farther distances. A similar imbalance also exists in semantic segmentation for automated driving vision datasets, where the datasets are largely imbalanced towards the sky, road and building *pixels* compared to smaller, less frequent, object *pixels* from traffic signs and light poles.

5. **Sensitivity of radar spectra signal**: Measurements from radar sensors are more sensitive than in vision, where even slight changes in the orientation of the object and viewing angle of the object can lead to drastic changes in the spectra. This is partially owed to the fact that the radar reflections depend on the composition of and interaction with the components objects.

3.3. Literature review

The application of learning algorithms to the radar domain is still at a relatively early stage. Many tasks which are currently solved using radar perception operate on different representations of the measurements. The literature can be grouped according to the application of learning algorithms to these different representations.

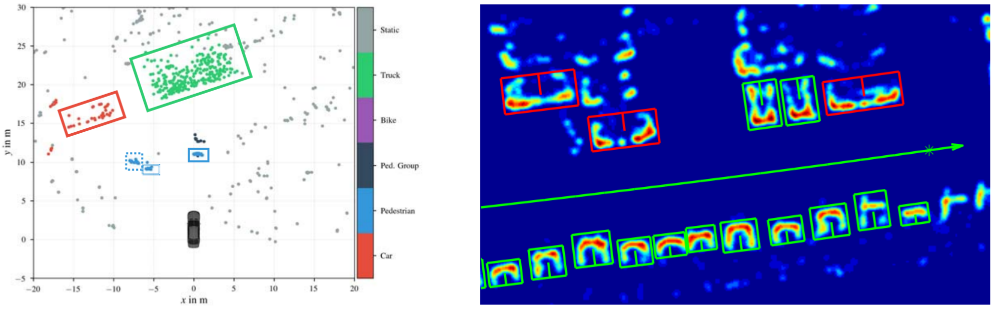
3.3.1. Point-cloud reflections and occupancy grids

Most approaches for automotive object classification work with radar reflections (see Fig. 3.3a), which first requires applying a statistical detection algorithm (e.g. constant false alarm rate (CFAR)) in order to curtail the information of the power spectrum to a set of detection points. These reflections can directly be used for object detection [Lee, 2020], but for classification an extra step is often first required. Reflections belonging to the same object are then typically grouped via clustering algorithms, before classifying based on the shape of the resulting point cloud. The performance of these classifiers often relies greatly on clustering algorithms in order to first reliably build clusters [Schubert et al., 2015] before accurate classification can be performed. Additionally, these classifiers also rely on the quality of the *hand-crafted* features [Prophet et al., 2019]. For example, the L-shape [Keat et al., 2005] or U-shape [Dubé et al., 2014] of the resulting clusters for classifying cars. Other examples of other feature extraction techniques can be found in Zhao et al. [2020]; Prophet et al. [2019]; Ulrich et al. [2021]. Given that radar measurements are often not human-interpretable, making it hard to construct hand-crafted features, they also struggle to generalize to other objects and scenes. Recent years have began to see a trend to replace this hand-craft feature selection step by learning algorithms which instead learn the discriminative features directly from the data measurements [Schumann et al., 2018; Feng et al., 2019]. The authors of Scheiner et al. [2019] proposed a deep learning-based recurrent neural network (RNN) ensemble technique for road user classification, where after point-cloud clustering each classifier in the ensemble is presented with its own specialized feature set. Even though deep learning was used, this feature extraction still employed a list of hand-crafted features, and the RNN was only used for learning temporal relationships of these features.

Recent years have also seen the removal of point-cloud clustering algorithms before the classification step. In order to avoid determining the association of each radar reflection to object clusters, an alternative strategy is to classify *each* reflection instead of clusters of reflections. Such methods yield a semantic segmentation of the point cloud, meaning that a potential class label is assigned to every detected reflection. PointNet [Qi et al., 2017a] was used in [Danzer et al., 2019] for classifying a single object, a car. For radar-based multi-class road user classification, a more recent modification of the model architecture, PointNet++[Qi et al., 2017b] was employed for classifying multiple classes in Schumann et al. [2018]; Kraus et al. [2020]; Feng et al. [2019]. These approaches, however, do not resolve individual instances, but merely provides an indication of how many reflections belong to a certain class in a given scene, and where those classes are located. Many point-cloud based classification algorithms rely on temporal aggregation of reflections for dense point-cloud representations before reliable classification can be done. Algorithms which work on single snapshot measurements (i.e. no temporal aggregation) are used in Wang [2017]; Prophet et al. [2018] for low-latency classification.

Similar to the point-cloud representation which uses individual detected reflections, *occupancy grids* accumulate these reflections, over time, in a grid to construct probability maps (see

Fig. 3.3b). This allows the use of convolutional neural networks (CNNs) for object classification [Lombacher et al., 2016] and detection of parked vehicles [Dubé et al., 2014]. The extraction and detection of parked vehicles was also shown in Lombacher et al. [2015] using CNNs for learning rotation-invariant features of the vehicles. A deep learning-based object detection algorithm was proposed in Engelhardt et al. [2019] by learning to generate occupancy grids directly from raw radar spectra data.



(a) Source: [Schumann et al., 2018]

(b) Source: [Dubé et al., 2014]

Figure 3.3.: An example of radar classification based on (a) point-cloud semantic segmentation from detected reflections and (b) occupancy grid representation of a parking lot scene with multiple cars (with a bounding boxes). Both representations require long integration times to aggregate reflection information over time.

Overall, point-cloud and occupancy grids methods work well for distinguishing object classes with distinctive shapes, but for harder tasks they are limited by the loss of information due to CFAR detection. Additionally, these representations often require long integration times to gather reflections from multiple frames and thus face a latency in fast changing environments such as automated driving. The reflex association task further complicates classification using these representations, as each reflection first needs to be grouped into tentative object clusters which faces challenges of its own.

As an alternative to these sparse reflection-based representations which are acquired at a late stage in the radar signal processing pipeline, radar sensor measurements can be represented as images which are a richer source of information. Extracting complex features from such information filled representations have shown great success in various tasks with the aid of deep learning algorithms such as CNNs. These include multi-dimensional FFT spectra images or synthetic aperture radar (SAR) images as alternative representations of the measurements, where the former spectra images are more often used in automotive applications.

3.3.2. Synthetic aperture radar (SAR)

Synthetic aperture radar (SAR) characterises radar sensors which exploit the motion of the sensor for an extended artificial aperture for finer spatial resolution, and is often used for aircrafts or spacecrafts. Even though SAR is a form of radar rather than a different representation of the data, it is used for generating two- and three-dimensional reconstructions of objects (e.g. terrain mapping). These SAR images (i.e. representations) of the scene allows one of the most direct application of convolutional neural networks (CNNs) [Lecun & Bengio, 1995], which has arguable shown the greatest success among all deep learning algorithms, to radar [Lang et al., 2020]. CNNs have been used for SAR image and target classification [Zhao et al., 2017a; Zaied et al., 2018] and ground target classification [Wang et al., 2018]. Target recognition using CNNs have also been combined with data augmentations (translations, speckle noising and pose synthesis) [Ding et al., 2016a]. To remedy the issue of speckle noise (i.e. a strong multiplicative noise) in SAR images, a residual learning strategy was proposed for image despeckling using CNNs [Chierchia et al., 2017], as well as a component-wise division-residual layer with a skip-connection to estimate the denoised image [Wang et al., 2017]. An autoencoder-based CNN architecture has also been proposed for denoising of SAR images [Mukherjee et al., 2018]. The task of change detection in SAR images has also been exploited using unsupervised feature learning and supervised fine-tuning using deep neural networks [Gong et al., 2016].

In the context of automated driving applications, SAR imaging has been used to measure automotive scenarios [Iqbal et al., 2015; Kan et al., 2020] but has been limited to parking lot detection [Wu & Zwick, 2009; Mure-Dubois et al., 2011] and detection of moving vehicles using semi-supervised learning [Mostajabi et al., 2020]. To the best of our knowledge, performing other driving tasks have not been exploited using SAR imaging. One possible reason is that generating these SAR images also require long integration times and have limited use in real-time applications such as automated driving. An alternative, larger, body of work focus on applying learning algorithms directly on the radar spectrum frames for various automotive applications.

3.3.3. Spectra

Radar spectra are a result of applying a multi-dimensional FFT on the raw radar signals and produce image-like maps of the environments measuring the range, azimuth, elevation and velocity (i.e. Doppler velocity). Most learning-based tasks are performed using range-Doppler maps which aim to exploit micro-Doppler signatures of moving objects, but range-azimuth and range-azimuth-Doppler maps have also been used for various tasks, including object detection [Wang et al., 2021], especially for static scenes where the micro-Doppler signatures are small or non-existant.

For dynamic scenes, learning algorithms have used the micro-Doppler signatures of object motions, which offer cues to the object class or the activity, for performing various tasks [Tahmoush, 2015; Chen, 2008]. Exploiting the kinetic and dynamic properties of the objects, deep learning has been used for the classification of unmanned aerial vehicle (UAV) [Huizing et al., 2019] and drones [Brooks et al., 2018], emotion recognition [Zhao et al., 2017c], as well as human-robot classification [Abdulatif et al., 2018] and human detection [Kim & Moon, 2016]. Learning algorithms have also been used for human motion classification and activity recognition [Li et al., 2019a; Le et al., 2018; Lang et al., 2017; Zhang & Cao, 2019; Gurbuz & Amin, 2019] using the micro-Doppler signature, including human fall detection [Jokanovic & Amin, 2018] and hand gesture recognition [Zhang et al., 2017]. A more comprehensive literature review on radar-based human activity recognition using deep learning can be found in Li et al. [2019b]. Radar spectra has also been exploited for human pose estimation [Zhao et al., 2018a,b] where the full radar spectrum was used to identify poses of multiple humans in the scene, as well as the use of CNNs and RNNs for learning sleep stages from the radar spectrum [Zhao et al., 2017b].

In the context of automated driving, the three-dimensional range-azimuth-Doppler spectrum (i.e. radar cube) has been used in Major et al. [2019] for deep learning-based vehicle detection, where the Doppler information was especially exploited for detecting the moving vehicles. The range-azimuth-Doppler spectrum has also been used for multi-class classification in Pérez et al. [2018]

and Palffy et al. [2020] using CNNs for classifying road users such as pedestrians, cyclists and cars, where the latter additionally used the individual radar point reflections. Similarly, Angelov et al. [2018] suggested to use a combination of a recurrent neural network-based long short-term memory (LSTM) and CNN for classification. The CNN was used for extracting discriminative features from the micro-Doppler spectra and use the LSTM to learn their representation as a time series. A CNN-LSTM architecture was also used for performing radar-based fall detection in Maitre et al. [2020], and additionally breathing detection was done in Bhattacharya & Vaughan [2020] using range Doppler spectra. The convolutional LSTM [Donahue et al., 2016] architecture was modified in Khalid et al. [2019] by introducing a novel convolutional-based compression layer to the model architecture.

The simultaneous detection and classification of objects in the radar spectrum as done in Lin et al. [2020], where the objects in the full radar spectrum were first detected using the YOLO algorithm [Shafiee et al., 2017] followed by classification using a CNN which also adopted transfer learning during training. Transfer learning and random crop data augmentation was also used in [Sheeny et al., 2019] for object classification in an indoor static scene. For further boosting the generalization performance data augmentation techniques for radar spectra was proposed in Sheeny et al. [2020]. These augmentations proposed were spectra attenuation, change of range resolution, background shift and speckle noise; partially based on augmentations introduced for SAR images in Ding et al. [2016b].

In Bartsch et al. [2012] the processing of raw radar spectra for pedestrian detection was suggested, but no machine learning was applied. Pedestrian detection was also performed in Prophet et al. [2018] where other objects such as vehicles, cyclists, and dogs are also considered.

Similar to these spectra-based methods, in this work we aim at performing multi-class object classification using region-of-interest (ROIs) of the radar range-azimuth spectrum. We construct a challenging real-world automotive radar dataset which measures multiple objects from various viewpoints in a static scene. It should be noted that unlike dynamic scenes where the micro-Doppler signature of the objects give strong cues to the class, the static scene is much more challenging as it needs to rely on other features to perform accurate classification. Such static object classifiers are particularly useful for automated driving tasks such as automatic emergency braking where the obstacles are often static and do not have a micro-Doppler signature.

In an attempt to address the literature gap LG1, this part aims at performing object classification using the radar spectra for static scenes. Different to other current state-of-the-art methods in the literature which use point-clouds [Schumann et al., 2018; Feng et al., 2019] or micro-Doppler signatures in dynamic scenes [Pérez et al., 2018; Palffy et al., 2020] for classification, we propose to apply convolutional neural networks (CNNs) to the range-azimuth spectra of *static* objects which will directly learn class discriminative features using data, unlike methods which use hand-crafted feature extraction techniques [Zhao et al., 2020; Prophet et al., 2019]. This literature gap also exists as a result of a lack of automotive radar spectra datasets, and as a first step in Chap. 4 we create a realistic static scene dataset for developing radar spectra-based classification algorithms. In Chap. 5, we discuss one of the first studies into radar spectra object classification which also utilizes domain knowledge to improving classification performance.

Chapter 4.

Radar system for automated driving

Given the lack of publicly available radar spectra datasets for multi-class classification, the big focus of the study in this thesis requires the creation of a new automotive radar dataset. In this chapter, we present the details of the measurement process, radar sensor, ground truth labelling, and pre-processing of the data to create the first realistic automotive static scene dataset for multi-class radar spectra classification.

4.1. Measurement process and setup

The goal of our study is to create a realistic scenario for classification with automotive radar sensors. In order to achieve this, we mount a radar sensor to the front bumper of a test vehicle which drives through multiple environments with static objects, approaching them from several different directions.

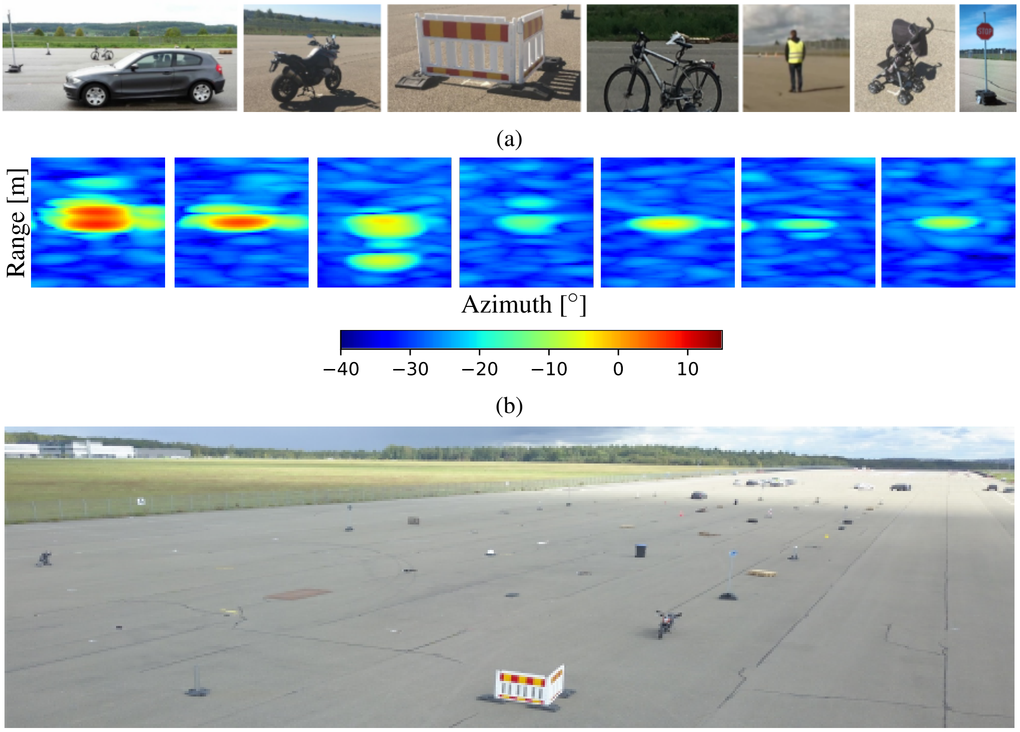
4.1.1. Object classes

As objects we selected the following seven objects (commonly found in urban scenarios): car, construction barrier, motorbike, baby carriage, bicycle, pedestrian, and stop sign. The learning algorithms have access to only a single instance of each object type (e.g. the car samples from a single environment only contain measurements from a BMW car) measured from different distances and angles. Camera images, and corresponding radar spectra, of these objects can be seen in Fig. 4.1(a-b). Although these objects are visually easy to distinguish, they pose a greater challenge for classification algorithms when working in the radio frequency spectrum. These static objects are placed in a 9m hex-grid layout on a test track (Fig. 4.1c) with the test vehicle driven through this layout.

4.1.2. Training and testing environment setup

To ensure an accurate estimation of the generalization performance of the learned classifiers, we record measurements from three different environment setups (i.e. scenes). We only use one of the environments (Env1) during the learning phase, and further split this dataset into the training set (Env1-Train) and validation set (Env1-Valid). We keep the other two unseen environments, Env2 and Env3, as tests sets to measure the generalization of the classifiers to a realistic unseen setting.

To increase the contrast, each environment setup has a different configuration of the object locations and different driving patterns through this scene. Env1 consists only of straight



(a)

(b)

(c)

Figure 4.1.: (a) Camera images of all 7 objects (car, motorbike, construction barrier, bicycle, pedestrian, baby carriage and stop sign) in the classification dataset. (b) The corresponding range-azimuth spectra (in dB) of these 7 objects (in the same order), each measured from a distance of 25m from the radar sensor (see Sec. 4.4 for more details). (c) Example environmental setup (Env2) of various objects in a hex-grid layout.

(horizontal and diagonal) driving patterns through the scene, whereas Env2 additionally consists of driving patterns which follow unique and special driving patterns involving a series of curves through the scene. Env3 only consists of driving patterns which drive straight towards the objects which are placed in front of the sensor. All three environments capture the objects from different angles and orientations and have non-identical distributions, thus posing a greater difficulty in generalizing to these novel unseen target orientations.

The diversity of the driving patterns are best seen in Fig. 4.2, which depicts the joint density of the ground truth range and azimuth of each object relative to the sensor in the three environments. The figures depict the object range and azimuth to the sensor (which are in the polar representation) in the Cartesian grid, hence the straight driving patterns of Env1 are seen as (banana-like) curves at closer range (i.e. range less than approx. 15m). For Env2 the actual curvy driving patterns are visible by the faint lines seen throughout the range-azimuth density plot. Env3 shows very high density at Azimuth 0° which is obtained when approaching objects which are located in front of the sensor. Despite the driving patterns in Env3 only approaching a single object at a time, the field-of-view of the sensor was large enough to capture neighbouring objects (seen by the faint lines in Fig. 4.2c).

Even though the range distributions (plotted on the right of each subfigure) in Fig. 4.2 are roughly similar, the azimuth distributions (plotted above each subfigure) greatly differ, showing that the three environments capture objects from new angles not seen in other environments. Another factor of difficulty in generalizing to Env2 comes from the fact that some object instances (e.g. car, motorbike, bicycle and pedestrian) were different from Env1 and Env3. For example, a Volkswagen (VW) Golf was used in Env1 and Env3, whereas a BMW 1 Series was used in Env2. Env2 also uses another radar sensor instance, though still the same sensor type. Additionally, the hex grid of the scene in Env2 consists of at least 60 *other* objects which are also commonly found in urban scenarios (e.g. garbage containers, truck tyres, traffic cones, wooden pallets) to mimic a realistic setting, as they can (especially at larger distances) interfere with the measurements of the 7 objects considered in this study (we note that the other 60 objects are *not* included in the resulting dataset prepared for classification). A camera image of the Env2 layout can be seen in Fig. 4.1(c). Considering these factors, Env2 poses the most challenges for classifiers trained using data from Env1, and throughout this thesis we extensively evaluate the performance of all classifiers and methods on this challenging test set.

4.1.3. Static vs. dynamic objects in the scene

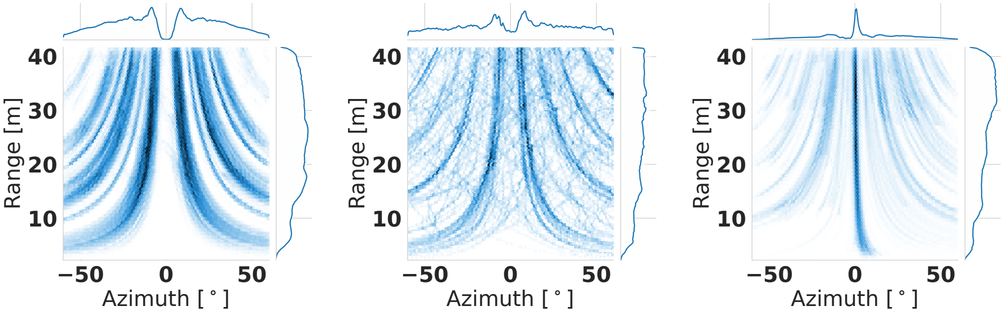
We note that all three environments consist only of static objects and only the test vehicle moves between these objects throughout the measurement process. Given the static nature of the scene, the measurements do not allow identifying the objects solely through the Doppler spectrum. For radar, dynamic objects with different Doppler spectra are easier to classify, due to their micro Doppler signature, but harder to record and annotate. For example, a moving bicycle with different moving parts may have an idiosyncratic signature in the Doppler domain, which would facilitate classification from permanent static objects such as stop signs. Therefore, a static scene classification dataset requires the learning algorithms to find other features which facilitate classifying the spectra of the objects.

4.2. Ground truth labeling

For every object in each environment, the differential GPS (DGPS) position is used to obtain the coordinate of each object.. During the measurement process when the test vehicle is driven through the environments, we also measure the DGPS and velocity of the test vehicle. This allows computing the relative coordinates of the different objects with respect to the radar sensor, i.e. the range r, the relative radial velocity v, and the direction-of-arrival (DOA) (azimuth angle) ϑ which serves as the ground truth in the following data processing.

4.3. Radar sensor system

The radar system is a multiple input multiple output (MIMO) automotive radar. The carrier frequency is $77\,\mathrm{GHz}$ with a bandwidth of $1\,\mathrm{GHz}$. It uses a chirp sequence modulation, i.e. a sequence of frequency modulated continuous wave (FMCW) chirps. The measurement time of one coherent processing interval is $15\,\mathrm{ms}$. The fully polarimetric sensor and its dual polarized waveguide antenna with 8 transmitting (Tx) and 8 receiving (Rx) elements is described in more detail in [Visentin et al., 2017, 2018]. We only use 4 transmitting (Tx) and 4 receiving (Rx) horizontally polarized antennas. The resulting virtual array of the MIMO radar is a linear array of



(a) Env1 **(b)** Env2 **(c)** Env3

Figure 4.2.: Ground truth object range and azimuth distribution for the three different environments plotted in the Cartesian grid. We use (a) Env1 for the training of all radar classifiers in this thesis and use (b) Env2 and (c) Env3 to test the generalization performance to unseen data. It can be seen that the ground truth distribution of the three environments greatly differ, making the 2 test sets challenging. Env1 consists only of straight horizontal and diagonal driving patterns, Env2 in addition consists of a series of driving curves through the scene and Env3 directly approaches each object in the environment.

16 antennas with an aperture of $8.5\ \lambda$, with $\lambda = 3.9$mm being the carrier wavelength. The cycle time of the radar system is approximately $57\,\mathrm{ms}$. The field-of-view (FOV) of the radar sensor covers a distance of 42m and we use an azimuth coverage of $-60°$ to $60°$.

4.4. Pre-processing and region-of-interest (ROI) extraction

The data preprocessing pipeline consists of the following steps, cf. Fig. 4.3:

1. A range-velocity spectrum is computed via a 2D-FFT using the raw time signals, resulting in 512 range and 64 velocity bins.

2. For every range-velocity bin, the magnitude of the azimuth spectrum is calculated, resulting in a 3D range-velocity-azimuth spectrum with 256 azimuth bins. The 3D range-velocity-azimuth spectrum has dimensions $(512, 64, 256)$. Examples of range-azimuth spectra are depicted in Fig 4.1.

3. Non-coherent integration over the virtual antennas (i.e. spatial dimension) of the range-velocity spectrum is performed and an ordered statistics constant false alarm detector (OS-CFAR) [Rohling, 2011] is applied to the spectrum to detect potential objects in the FOV.

4. For each *detected* object from the OS-CFAR, a region-of-interest (ROI) of the 3D-spectrum is extracted with the range, velocity and azimuth extent of $5\,\mathrm{m}$, $0.7\,\mathrm{m/s}$, and $0.5\,\mathrm{rad}$ [1], respectively, where the highest detected peak of the object is in the center of the ROI.

5. The ground truth information (i.e. the location of the objects in the environment relative to the sensor) is combined with the preprocessed data in order to automatically label each ROI.

[1] We note that we use electrical angle $sin(\vartheta)$. For $\vartheta = 0$ the azimuth extent is $0.5\,\mathrm{rad}$ and for $\vartheta \neq 0$ it increases. For the sake of convenience we refer to it as azimuth.

In this study we are mainly interested in the range-azimuth spectrum, therefore we take the maximum velocity at each bin in the 3D ROI which results in a 2D ROI containing 64 range and 66 azimuth bins (i.e. for each bin in the 2D ROI the maximum velocity was taken). Example ROIs from the range-azimuth spectrum, of each object measured at a distance of 25m, can be seen in Fig. 4.1(b). The set of these 2D ROIs and the corresponding labels are the inputs for the learning algorithms. Using this computation pipeline for each environment results in 414,281, 49,726 and 355,544 data samples for the three environments Env1, Env2 and Env3, respectively, where only part of Env1 is used during the learning phase.

4.5. Synthetic radar spectra corruptions

Even though Env2 serves as a test for the classifier's ability to generalize to realistic changes in the spectra (due to other object instances, driving patterns and environment setup), it does not provide a framework for incrementally changing the spectra in a controlled setting. This has been studied in vision after the construction of the ImageNet-C corruption datasets [Hendrycks & Dietterich, 2019]. Classification robustness to radar spectra corruptions has not been studied before, though plays a vital role in accessing the reliability of the learned classifiers. In order to study the behavior of the learning algorithms to slight but incremental changes in the spectra, we incrementally corrupt the ROI spectra to measure the robustness of the radar classifiers to perturbations in the input. Seven corruption types, each at three severity levels are used. In Fig. 4.4, we depict some examples of the corrupted spectra of a construction barrier. Some corruptions are more realistic and could feasibly be encountered in the real-world (e.g. amplification, dampening) and some are more unrealistic (e.g. subsample). Despite these not all accurately simulating real-world corruptions, they are still well suited for the task of studying the behavior to unseen changes in the input of the classifier. We also note that the parameter controlling the amount of corruption, c, for each corruption type, is arbitrarily chosen in order to obtain roughly similar accuracies from a deep learning baseline classifier across all corruption types. Other choices c are also valid, but we are more interested in the relative performance of multiple classifiers, so the exact choice of c does not play a vital role in the evaluation.

The corruption spectra dataset consists of the following corruptions and severity levels (performed on the linear scale spectra before transforming back to the dB scale):

1. **Amplify**: the entire spectra is multiplied by some scalar $c > 1.0$. We pick the following values for c: 1.01, 1.05, 1.08.
2. **Dampen**: the entire spectra is multiplied by some scalar $c < 1.0$. We pick the following values for c: 0.99, 0.95, 0.90.
3. **Speckle Noise**: multiplicative Gaussian noise is applied to the (linear scale) spectra with unit mean and standard deviation c. We pick the following values for c: 0.002, 0.005, 0.01 (truncated for negative values). We note that speckle noise has also been used in [Sheeny et al., 2020] as an augmentation technique used during training; we only use this during the evaluation phase.
4. **Shifts**: the entire ROI is randomly shifted by c pixels in one of the four directions. We pick the following values for c: 3, 6, 8.
5. **Subsample**: the ROI is subsampled by sampling every c^{th} row and column and then resized back to the original dimension. We pick the following values for c: 2, 8, 12.

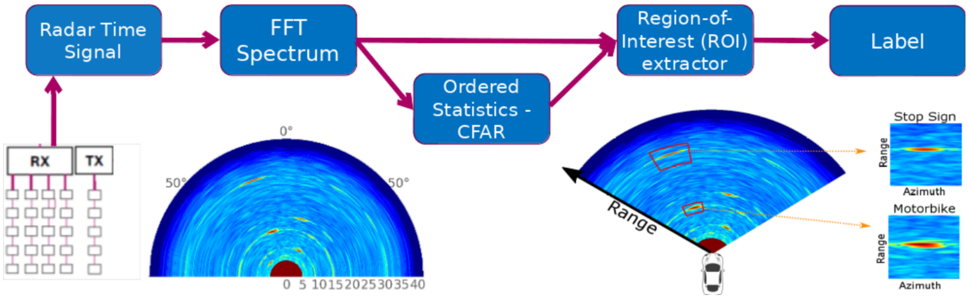


Figure 4.3.: Illustration of the computational pipeline of the preprocessing and region-of-interest (ROI) extraction for the creation of the radar spectra classification dataset. An example of two detected objects (stop sign and motorbike) in the field-of-view is depicted on the right.

6. **Zoom**: the peripheral regions (defined by the outer-most c pixels in each direction) of the ROI are ignored and the remaining spectra is resized back to the original dimension. We pick the following values for c: 3, 5, 8.
7. **Smoothing**: Gaussian smoothing, with standard deviation c, is applied to the spectra. We pick the following values for c: 0.75, 2.50, 4.50.

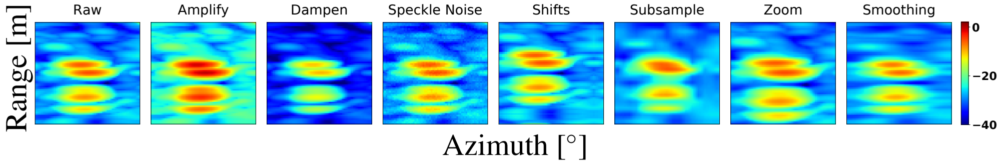


Figure 4.4.: Construction Barrier ROI spectra (in dB) after corrupting the Raw signal (left) with various corruptions.

4.6. Out-of-distribution class dataset

Autonomous systems constantly encounter unknown objects which they are not trained to classify, therefore it is important to study how the classifiers react in such situations. In our case, the learning algorithms are only trained to classify seven objects, which is far less than what will be encountered in the real-world. To study this behavior we collect measurements from objects which are not seen during the training. These objects range from various metallic objects to sand bags and concrete blocks. In total we use the following 5 objects to construct a special out-of-distribution (OOD) dataset from Env1: metallic manhole cover[2], 500ml coke can, sand bag, concrete block, and metallic pole.

[2] i.e. gullydeckel or schachtdeckel

Chapter 5.

Using deep learning for radar spectra classification

Contribution: Automotive radar has shown great potential as a sensor for driver assistance systems due to its robustness to weather and light conditions, but reliable classification of object types in real time has proved to be very challenging. This chapter proposes a novel concept for radar-based classification, which utilizes the power of modern deep learning methods to learn favorable data representations for multi-class classification of static objects using radar spectra. We propose to apply deep convolutional neural networks (CNNs) directly to regions-of-interests (ROIs) in the radar spectrum and thereby achieve an accurate classification of different objects in a scene. Experiments on a real-world dataset demonstrate the ability to distinguish relevant objects from challening new viewpoints. In this chapter, we will also identify deep learning challenges that are specific to radar classification and introduce a novel mechanism that lead to significant improvements in object classification performance compared to simpler classifiers. Specifically, we introduce the concept of utilizing the range-azimuth information available in the signals to improve the feature representation learned by the classifier. Our results demonstrate that deep learning methods can greatly augment the classification capabilities of automotive radar sensors, and thereby making them more attractive as both a stand-alone sensor for scene understanding and as a vital component for multi-modal sensor sets.

5.1. Introduction

Deep learning is capable of learning favorable representations for raw input data in deeper layers of neural networks, which capture the crucial features necessary for object classification, but also exhibit invariances to viewpoints, noise, and other transformations. In order to exploit algorithmic advancements from deep learning algorithms applied to vision, we perform radar object classification based on the radar spectra generated by the multi-dimensional Fast Fourier Transform (FFT). This representation not only preserves all information available in the raw signal but also yields a data representations favourable for powerful deep learning methods, such as convolutional neural networks (CNNs).

In this chapter, we address a number of the challenges identified in Sec. 3.2 by borrowing ideas from the well-established literature for vision. Through careful design of the model architecture and training procedure for radar classifiers, we are able to ensure highly accurate classification of the objects using radar spectra. In order to address the radar-specific challenges, which rise from the direct and naive application of deep learning, we develop novel techniques which lead to significant improvements in the classification performance. In particular, we observe that combining the spectra with information about the range and direction of arrival (DOA) of the object is beneficial for the classifier to learn more discriminative features. Even though deep learning classifiers are able to learn such features from the data, the limited size and diversity of the training set pose a great impediment. We show that despite this limitation, classifiers can greatly benefit by the intervention of radar-specific knowledge to the training process. Furthermore, we show that integration of predictions over time can lead to substantial improvements.

In order to demonstrate our results, we perform extensive experiments and evaluations on the challenging automotive radar spectra test sets introduced in Chap. 4, which contain measurements from multiple object classes which are relevant for scene understanding. The results demonstrate that prediction of object classes in real time with neural networks works reliably for all classes, and filtering classification results over time can greatly improve the performance.

5.2. Deep learning architecture: CNN

For the task of object classification, we train a convolutional neural network (CNN) [LeCun et al., 2015], which takes as input the radar ROI spectra. The model architecture used in this work was empirically found after a hyper-parameter search for multiple architecture design choices. The CNN consists of 3 convolutional layers, containing 16, 32 and 64 3 × 3 filters, in addition to 2 × 2 max-pooling layers. They are followed by 2 fully-connected layers (with 512 and 32 neurons), each using Batch Normalization [Ioffe & Szegedy, 2015] and Dropout [Srivastava et al., 2014] (with drop out probability of 0.40). Training of the network weights uses the Adam optimizer [Kingma & Ba, 2014] with a batch size of 128, where each batch is ensured to be class-balanced through class weighted sampling of the data. The CNN architecture and the feature maps at different layers can be seen in Fig. 5.1.

Despite multi-class classification using radar being challenging, rather small capacity models are used for this task. Unlike in vision, where the classifiers greatly benefit from large deep model architectures, radar classifiers are much more susceptible to over-fitting (mainly due to the lack of sufficient data), and as a result relatively small model architectures need to be used.

We train the networks only using Env1-Train and select the model with the best accuracy on the independent validation set Env1-Valid. We evaluate the performance of the networks on two unseen test datasets: Env2 and Env3. We train 10 independent networks with different initializations, but identical architecture, and report their mean and standard deviations.

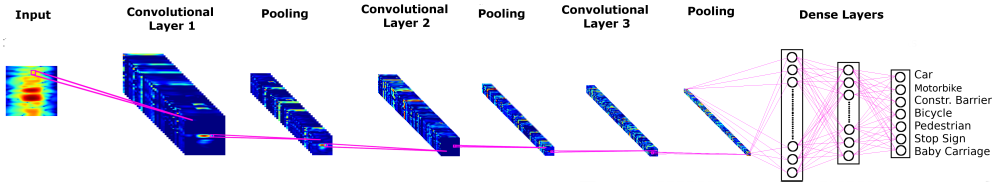


Figure 5.1.: CNN Architecture for ROI input. This CNN architecture with 3 convolutional and 2 fully-connected layers is kept identical for all CNN experiments.

5.3. Incorporating range-azimuth information

The ROIs represent only a portion of the entire field-of-view (FOV), i.e. full range-velocity-azimuth spectrum. The direct application of CNNs to ROIs completely ignores the absolute range and azimuth information of the objects, which is lost when considering the ROI alone. In this study, we find that as the sensor operates in the polar coordinate system, the physical area covered by the ROI expands with range and affects the signals measured in the spectra, and consequently its accurate classification. Therefore, the ROI may capture reflections from multiple targets and their side-lobes, presenting themselves as pernicious noise. Hence, most periphery parts of the ROI present distractions to the classification algorithm.

Without any prior information about the location of the ROI in the FOV, a machine learning algorithm needs to learn to deal with these distortions from the data, as well as learn to ignore reflections from other objects. Achieving this would require a much larger and more diverse dataset than we have available, and more importantly would require measurements of ROIs from all regions of the FOV, as well as measuring all combinations of the objects distorting one another. Alternatively, data augmentation by selected transformations applied to ROIs could aid for such a task, though data augmentation for radar spectra is still an open problem.

An additional factor of difficulty faced by the CNN is the imbalance of the range and azimuth distribution of the ROI data. For example, given that the datasets are biased towards ROI from objects which are farther away (discussed in Sec. 3.2), the CNNs learn from very little data from nearby objects. As this imbalance will always be present, even with more measurements, this range and azimuth information can be leveraged by CNNs to implicitly fix this imbalance.

In the following, we propose two novel radar-specific approaches to incorporate prior information about the location of the ROI in the FOV as an additional input to the networks, and to suppress interfering reflections from nearby objects.

Additional information about the geometry of the ROI can be provided to the CNN in the form of an additional input channel. This so-called distance-to-center (DTC) map contains the *physical* distance (in meters) of each bin of the ROI to the center bin of the ROI, thereby implicitly encoding both the range and azimuth information. The CNN can learn to leverage this information (by applying the filters across the two input channels) to extract object-specific features, such as its size and reflectivity, as well as efficiently learn how the signal is distorted according to its relative location.

In order to explicitly attenuate all reflections and noise (including most but not all side-lobes from other object reflections in the FOV) from bins which do not originate from the direct vicinity of the object, the DTC map can be fused with the radar spectrum by decaying the intensity of each bin as a function of its distance to the center bin. The linear scale spectrum in the ROI is exponentially decayed by multiplying with $e^{-a\cdot(d-d_{\text{min}})}$, where d is the distance to the center bin obtained from the DTC map, and a is a hyper-parameter determining the rate of decay (empirically optimized as $a = 0.5$). In order to avoid attenuating reflections caused by the object, a pre-selected minimum decaying distance, d_{min}, is set where bins with $d < d_{\text{min}}$ are considered important and kept unaffected. This hyper-parameter is easily exchangeable and can be set in order to capture all possible reflections originating from the largest object class to be predicted. For all experiments, we set $d_{\text{min}} = 2.5\,\text{m}$ which captures most reflections from objects in this study. An ablation of other choices of d_{min} can be found in Sec. 5.8. The exponential decay generates a combined single input channel, which implicitly contains the location information (which can be learned from the decay rate of the signal), and pronounces object reflections by attenuating the periphery signals. An example of a radar ROI spectra, the DTC map and the decayed spectra can be seen in Fig. 5.2.

Overall, this allows us to compare four variants that incorporate the spectra and radar-specific knowledge in the input to the CNN:

- Baseline: ROI spectrum
- DTC: ROI spectrum + DTC map (2 input channels)
- SpecDecay: Decaying the ROI spectrum outside d_{min}
- DTC + SpecDecay: Combination of DTC and SpecDecay

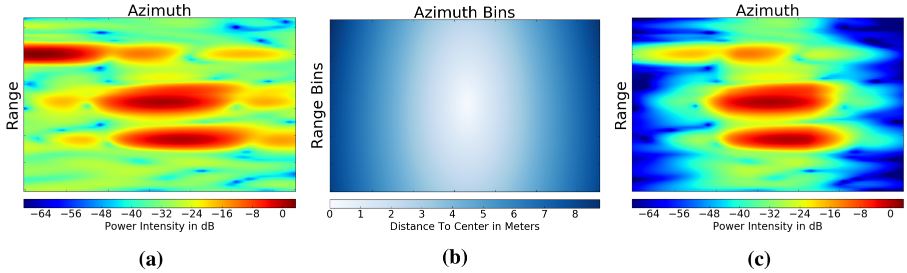


Figure 5.2.: An example ROI of a construction barrier measured at range $30.36\,\mathrm{m}$ and azimuth $0.56°$ with reflections from another object (top left region). (a) ROI spectrum, (b) Distance-to-Center (DTC) map, and (c) SpecDecay (spectra decayed) ROI spectrum. It can be seen in (c) that the peripheral reflections are attenuated and important reflections are pronounced.

5.4. Experimental setup

In this section, we describe the experimental setup which evaluates the generalization performance of the classifiers. We compare the CNN classifiers to alternative learning algorithms, as well as extensively compare the performance of the different variants of the CNN classifiers proposed in this chapter.

5.4.1. Tasks

We evaluate and compare the performances of the classifiers at multiple tasks:

1. **Generalization performance** (Sec. 5.5): Evaluation of the classification accuracy, negative log-likelihood (NLL) [1] (i.e. the cross entropy loss used during training) and Brier score (mean-squared-error of the predictions to the one-hot labels). These evaluations measure the ability of the classifiers to generalize to unseen data in the real-world, and plays an important role to ensure that the predictions are accurate and reliable in the future.
2. **Spectra corruptions** (Sec. 5.6): To study the behavior of the classifiers to incremental corruptions of the input, the input spectra are corrupted and evaluated.

[1] NLL is defined as the multi-class cross-entropy between the ground truth label vector $\mathbf{y}$ and predicted probability vector $\mathbf{f}(\mathbf{x})$ for an input $\mathbf{x}$ as $-\sum_{k=1}^{K} -\mathbf{y}_k \log \mathbf{f}(\mathbf{x})_k$ for a K-class classification problem.

3. **Merging predictions over time** (Sec. 5.7): A study of the combination of multiple single-frame predictions from consecutive frames to incorporates knowledge from past observations. Showing the benefits in incorporating temporal information, allows the real-world classifiers to exploit previously observed and predictions for the current measurements.

4. **Ablation of d_{min} for spectra decay** (Sec. 5.8): A study of the efficacy of the hyper-parameter d_{min} to show that the results are consistent for other settings of the hyper-parameter.

5. **Analysis of the per-class confusions** (Sec. 5.9): A study of the confusion between objects faced by the classifiers.

5.4.2. Classification with KNN and SVM

There are no public comparable datasets or algorithms for radar spectrum classification, hence we have to create our own baseline to put the accuracy of the CNN classifiers into context. We compare against other machine learning classification algorithms, in particular, K-Nearest Neighbor (KNN) for $k = 3$ and $k = 5$ with the standard Euclidean distance as the metric and Support Vector Machines (SVM) with a radial basis function kernel, which can both operate directly on the 2D ROI images. The classification accuracy of 3-NN, 5-NN and SVM is 38.21%, 40.10%, and 42.34%, respectively, on Env2, with the Baseline CNN classifier achieving 52.40%. We find that these algorithms were not competitive and that these solutions do no offer real-time solutions to classification, therefore were not considered in the rest of the studies.

5.5. Generalization performance

The first experiments evaluate the effect of the different input representations I_1 (ROI), I_2 (ROI + DTC map), I_3 (decayed ROI) and I_4 (DTC + SpecDecay). For this experiment, the network architecture and hyper-parameters are kept identical and only the network inputs are changed.

In Tab. 5.1, we compare all classification algorithms where we report the mean and standard deviations, across 10 independent runs (i.e. exact same trainings but with different initializations), of the classification accuracy, negative log-likelihood (NLL) and Brier score. The input representation with distance-dependent exponential decay in peripheral parts of the ROI (SpecDecay) consistently leads to the best performance across all metrics and test sets. We note that the CNNs have higher accuracies for Env3 than Env2, because Env2 contains other object instances and many more novel viewing angles than Env1, making the classification task more challenging. The CNNs benefit from the implicitly encoded geometrical information in the DTC map and the decayed spectrum (SpecDecay). This shows that CNNs greatly benefit from incorporating radar-specific knowledge, such as the radar and azimuth information, into the training process in order to generalization to real-world settings, especially when this information is not automatically learned directly from the data.

We find that the combination of DTC + SpecDecay (I_4) performs slightly worse (e.g. 1.1% at accuracy) than SpecDecay (I_3) (which is counter-intuitive) but still better than DTC (I_2). One possible reason behind this is that the classifiers could be relying more on the information from the DTC map than the SpecDecay ROI, and not learning to ignore the peripheral regions during the training. Without the DTC map, SpecDecay needs to learn this map information which is implicit in the decayed spectra, and as a result learns more discriminative features in the process. In summary, the use of radar-specific input representations has a clearly beneficial effect.

5.6. Spectra corruptions

In order to study the generalization of the classifiers to unseen distribution shifts, we corrupt the radar spectra with corruptions defined in Sec. 4.5. In Fig. 5.3, we visualize boxplots of the (a) Accuracy and (b) NLL performance of the CNN classifiers evaluated on corrupted spectra. We aggregate the evaluations of all 7 corruption types per severity and study the behavior as the spectra are increasingly corrupted. Given that all 7 corruptions are very different in nature, the classifiers exhibit different degrees of robustness against them, with this effect amplified for larger severities. As a result of visualizing the performances of all 7 corruptions, all boxes show large variances as the severities become larger, especially for NLL. Even though boxplots with larger variances are harder to compare, we use the mean (white circle marker of each box) and the median (horizontal line in each box) as indicators of the overall performance.

As expected, the overall performance of the Accuracy and NLL, both become worse as the corruption severity increases. In terms of the accuracy, the classifiers DTC, SpecDecay and their combination show significant improvements compared to the Baseline, with the best performance observed with DTC + SpecDecay. We observe something interesting for the NLL performance where only DTC + SpecDecay yields the best overall performance: there is no obvious winner between the Baseline, DTC and SpecDecay. We note that the corrupted spectra could also be used during training to further boost generalization performance but we are only interested in using these corruptions as a measure to judge the quality of the classification, and rather focus on the algorithmic developments.

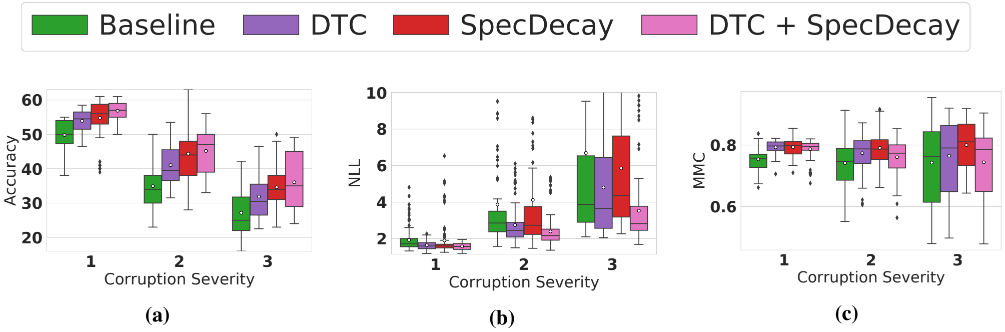
Unlike the accuracy metric which only considers whether a sample is correctly classified or not, the NLL considers the divergence between predicted probabilities and the one-hot hard labels. Improvements in accuracy but worse performance on NLL indicates an issue of over-confidence of the classifier, which is penalized much more by the NLL metric than accuracy. We observe that despite the accuracy of *all* methods severely degrading with larger corruption severities (i.e. larger than approx. 30% for severity 2), the confidences of the predictions remain high. This can be observed in Fig. 5.3(c) which depicts the mean-maximal-confidence (MMC), which is the average confidence assigned to each prediction by the classifier. It can be seen that despite higher rates of incorrect predictions when encountering unseen spectra corruptions, the predicted confidences remain roughly the same (seen by the mean and median of each box staying mostly constant). An ideal classifier should reduce the confidence of its incorrect predictions to reflect uncertainty in such situations, though we observe that regardless of the correctness of the predictions, the current classifiers over-confidently assign high confidences to incorrect predictions. This observation explains the poorer NLL performance of Baseline, DTC and SpecDecay, despite the latter 2 methods showing significant accuracy gains. We tackle the over-confidence problem of deep learning classifiers in Part II of this thesis. We also discuss ways in which this over-confidence can be detected and quantified (in Chap. 6), as well as discuss the negative effects it has on radar classifiers (in Chap. 10) which limits their use in safety-critical applications such as automated driving.

In summary, we find that significant accuracy gains can be achieved by the CNN variants which include the range and azimuth information in the input, with the best robustness obtained by

Table 5.1.: Comparison of the CNN-based classifier variants at Accuracy, NLL and Brier score on Env2 and Env3. Across all metrics, we find that CNN greatly benefits from the range and azimuth information in the form of the DTC map, and implicitly in SpecDecay. We bold the best performing method for all metrics and test datasets. The mean and standard deviations of 10 independent runs are reported.

	Env2			Env3		
Method	Acc ↑	NLL ↓	Brier ↓	Acc ↑	NLL ↓	Brier ↓
Baseline	52.4 ± 0.52	1.621 ± 0.11	0.675 ± 0.02	57.1 ± 0.32	1.572 ± 0.09	0.612 ± 0.01
DTC	57.7 ± 0.67	1.567 ± 0.12	0.621 ± 0.02	59.9 ± 0.32	1.605 ± 0.09	0.592 ± 0.01
SpecDecay	**59.2** ± 0.42	**1.415** ± 0.06	**0.589** ± 0.01	**60.5** ± 0.53	**1.499** ± 0.08	**0.580** ± 0.01
DTC + SpecDecay	58.1 ± 0.57	1.536 ± 0.12	0.614 ± 0.02	59.6 ± 0.52	1.576 ± 0.17	0.591 ± 0.02

the combination DTC + SpecDecay. Despite this robustness, we begin to observe the negative effects of the large modelling capacity of deep learning classifiers, which have a tendency to be over-confident in its predictions.



(a)

(b)

(c)

Figure 5.3.: Boxplot visualizations of the (a) Accuracy and (b) NLL performance for increasing intensities of dataset shift (i.e. spectra corruptions) on Env2. The boxplots aggregate the evaluation of all 7 corruptions for each severity (which explain the large variances), with the mean (white marker of each box) and median (horizontal line in each box) used to judge the relative performances. We observe that DTC, SpecDecay and its combination all help improve the accuracy performance of the Baseline classifier for all severities, showing their robustness to unseen corruptions. Based on NLL performance, we begin to see that the classifiers suffer greatly from the issue of over-confidence, with only the DTC + SpecDecay showing a performance improvement at large severities. This over-confidence is more clear when analysing the (c) mean-maximal-confidence (MMC), which is the average of all predicted confidences. We note that despite lower accuracies at larger severities (i.e. more samples are mis-classified), the average confidences remain roughly the same. This means that even though more samples have been mis-classified, their predicted confidences have remained very high, therefore these show up as large NLL losses in (b).

5.7. Merging predictions from consecutive frames

Due to the sensitivity of radar reflections to the aspect angle to the objects, radar spectra and their classification may abruptly change from one frame to the next. Merging classification results over time is, therefore, an obvious way to improve classification performance. A simple and computationally efficient approach is to combine multiple single-frame predictions by averaging the softmax class probabilities of the last T seconds. This operation improves performance by integrating previous predictions, and using the prior knowledge that classification results for *static* objects should not change abruptly across time.

In Fig. 5.4, we evaluate the accuracy and NLL performance of merging multi-frames over time. Fig. 5.4(a) shows that, as expected, increasing the integration time improves the classification accuracy for a static environment. It can further be seen that the performance order between the four CNN variants Baseline, DTC, SpecDecay and DTC+SpecDecay is maintained, and again the distance-dependent exponential decay (SpecDecay) shows the best performance for all integration lengths. As temporal filtering is a simple method to incorporate temporal information into the prediction stage, it relies highly on the single-frame classification performance. We also experimented with presenting multiple frames simultaneously as inputs to the CNN, but no advantage over majority-voting was visible, and the resulting CNN is of significantly higher complexity. Alternative approaches to integrate temporal information is a focus of future work.

Interestingly, we observe that the NLL performance in Fig. 5.4(b) begins to degrade after $T = 1$s. This is owed to the same over-confidence issues observed in the spectra corruptions evaluations in Sec. 5.6. We find that merging single-frame predictions from multiple frames results in higher confidences for both the correct and incorrect classifications. For small integration times (< 1s), where the accuracy gains are still large, the classifiers benefit from this increased confidence as the confidence of the correctly classified samples increase. Though, longer integration times, for which the accuracy gains saturate, amplify this effect and wrongly increase the confidence of the incorrect samples. We tackle the over-confidence problem of deep learning classifiers in Part II of this thesis.

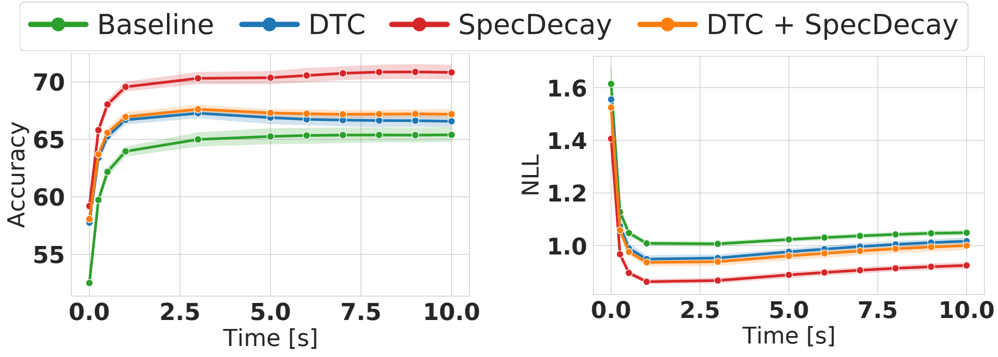
In summary, temporal filtering of CNN predictions significantly improves the classification accuracy performance over single-frame approaches, though become wrongly over-confident on the incorrect predictions.

5.8. Ablation of d_{min} for spectra decay

In Fig. 5.5, we evaluate the performance of SpecDecay training by performing an ablation on the minimum decaying distance, d_{min}. We observe that decaying larger parts of the peripheries (i.e. smaller d_{min}) to increasingly focus on the center of the ROI (where the main peak lies) shows better performance. This result shows that incrementally decaying larger parts of the peripheries translates into better performance gains, and shows that the idea behind decaying the spectra to force the classifiers to focus on the important parts of the spectra and ignore reflections from other objects, greatly benefits the learning algorithm. We also observe small performance gains with using $d_{\text{min}} < 2.5$m, though we choose not use them, in order to avoid changing reflections from large objects. The reason behind the performance gains, even though $d_{\text{min}} < 2.5$m can decay some reflections of larger objects, can be explained by the fact that most of the objects in the dataset are relatively small and are unaffected by this change.

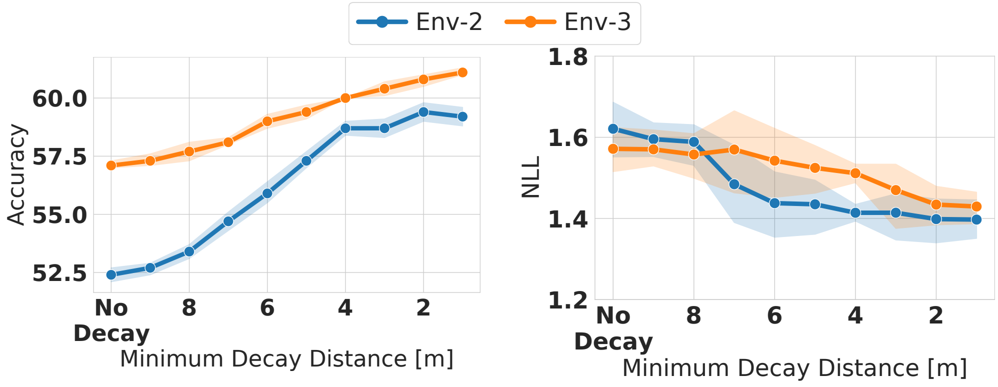
5.9. Analysis of the per-class confusions

Not all objects are equally difficult to classify, hence it is interesting to observe the per-class accuracies and confusion among the classes. The confusion matrices in Fig. 5.6 show how often



(a) (b)

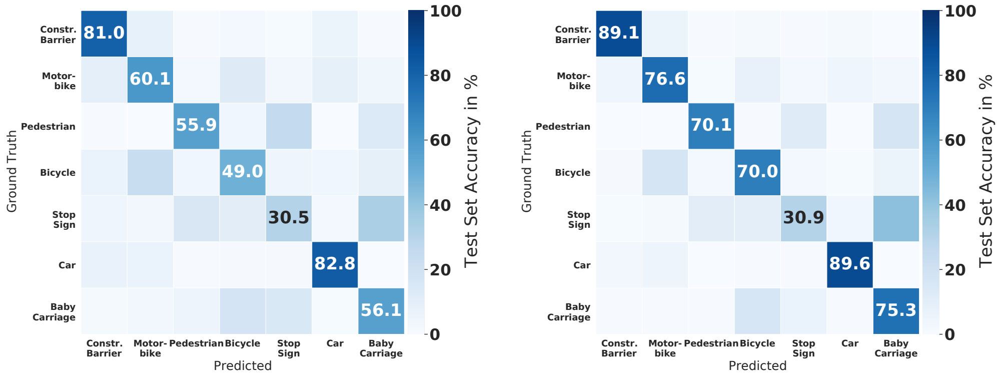
Figure 5.4.: (a) Accuracy and (b) NLL performance after merging single-frame predictions over time for all CNN classifiers. A clear improvement on accuracy for longer time windows over single frame ($T = 0.0$) approaches is visible. The NLL performance also significantly improves up until $T = 1.0$s, but all methods become slightly worse for longer time windows. The opposite trends of the accuracy and NLL indicate that longer time windows make the classifiers over-confident when simply average the softmax predictions. Using $T = 1.0$s yields the largest performance gains within a reasonable time window to restrict the latency for larger time windows.



(a) (b)

Figure 5.5.: Evaluation of the (a) Accuracy and (b) NLL for the minimum decay distance, d_{min}, of SpecDecay for Env2 and Env3. Larger minimum distances mean more parts of the spectra are unchanged, with the left most "No Decay" corresponding to no spectra decaying at all (i.e. Baseline). We notice that as the d_{min} becomes smaller, the performances mostly improve. In order to ensure that all reflections from all objects considered in the study are captured in the spectra, a reasonable minimum decay distance would range between $1\text{m} - 3\text{m}$ and still ensures significant performance gains. We selected 2.5m as it covers the largest object in the dataset.

each of the 7 objects (true class in rows) is classified into every other class (predicted class in columns). Fig. 5.6(a) shows the confusion matrix of the best single-frame CNN (SpecDecay), whereas Fig. 5.6(b) shows the confusion matrix after merging predictions with a window size of $T = 1$s. A window size of 1 second was chosen here because it is a reasonable time frame for sequentially observing a single static object in the dataset. Both matrices show the desired concentration in the diagonal (indicating correct classifications). For the single frame case, large objects such as car or construction barrier are best classified, whereas there is some confusion between stop sign, pedestrian and baby carriage, baby carriage and bicycles, and bicycles and motorbikes. With temporal merging, many of the confusions are removed, although there are still several cases where confusions still remain. Ultimately, it will be more important to identify the functional relevance of the object, e.g. whether an emergency brake is necessary, rather than the exact identity. This will be the focus of further studies with a larger set of objects.



(a) Env2 - Single-frame

(b) Env2 - Multi-frame (1s)

Figure 5.6.: Test set confusion matrices for (a) single-frame SpecDecay CNN and (b) merging these predictions over time with a window size of $T = 1$s.

5.10. Conclusion

The above results indicate that deep learning applied to automotive radar spectra is a promising approach for object classification and scene understanding. Since all objects were measured from many different distances and aspect angles, and under real-world conditions, the high accuracies of CNNs show that deep networks are able to extract features from spectra that allow them to generalize well. This opens up possible research questions about interpreting the features which the network extracts. Furthermore, the architectures of the CNNs are small enough to be efficiently implemented in hardware.

This means that our proposed system has a high potential for real-time radar-based object classification. Currently, the ROI extraction and classification only occurs for the detected objects for which the ground truth (i.e. label) exists. The system does not yet have the ability to classify reflections from an unknown object in the full spectrum which also produced a detection (e.g. road curb or false detections). The detection process currently uses a conventional detection algorithm (OS-CFAR) in order to extract ROIs to ensure that the target was detected; this step can potentially be replaced by a neural-network based detection approach modeled after successful region-proposal schemes used in the vision domain [Ren et al., 2015; Liu et al., 2016].

There are no directly comparable data sets or classification methods for this task, hence our results show relative comparisons between different machine learning methods and input representations. We find a clear advantage for deep learning methods over simpler methods such as KNN and SVM. In the future, a comparison with state-of-the-art reflection-based methods is necessary to evaluate whether the advantage of the method lies in the additional information available in the spectra, or in the powerful CNN classifier. We also plan to expand measurements to even more object classes and multiple instances of each class to evaluate the true generalization capabilities. Currently, our database contains only static objects, but the approach should easily transfer to dynamic scenes, where Doppler information could provide an additional cue to distinguish objects.

Two insights from our study are particularly interesting for future studies: First, the explicit integration of radar know-how into input pre-processing yields significant improvements. This suggests that additional insights from radar signal processing, e.g. for data augmentation or input normalization, could improve the performance even more. Second, the merging of classification predictions over time provides a significant boost to the CNN classifiers. Our results show that even an integration over a single second can already improve the accuracy by 12%. Our current approach uses a simple averaging over multiple single-frame predictions, but it seems likely that a direct accumulation of evidence in a recurrent neural network architecture such as LSTM [Hochreiter & Schmidhuber, 1997] yields similar or potentially better results.

This work has presented the first evaluation of deep learning methods applied directly to radar spectra for scene understanding under real-world conditions (addressing literatre gap LG1). The approach presents a promising alternative to classical radar signal processing methods and outperforms other machine learning approaches on a novel dataset with realistic objects. The best results can be obtained by combining state-of-the-art deep learning with specific radar know-how and prior understanding of the task. This suggests that a hybrid between data-driven and model-based approaches may have the greatest chance for success, in particular with limited available real-world training data, for addressing literature gap LG1. These initial results show that CNNs can be used to learn discriminative features which can allow generalization to novel object instances, viewpoints, as well as to measurements from another sensor instance, from static scenes where micro-Doppler features cannot be exploited. This is an important step towards reliable *radar object classification* (LG1) which offers automated driving systems more robust perception in situations were other sensor modalities are not reliable.

Finally, we note that despite the high accuracy performance of all CNN classifiers, especially those which incorporate radar-specific information in the input, we find the classifiers to be over-confident. Undetectable through only measuring the accuracy performance, some observations using the NLL performance showed some first indicators of the bigger problem of over-confidence [Hein et al., 2019] and mis-calibration [Guo et al., 2017] in deep learning. Despite the accuracy not offering any cues to a classifiers over-confidence or mis-calibration, radar practioners often evaluate and judge the performance of classifiers *solely* in terms of the accuracy performance. Such evaluation frameworks result in the literature gap LG2, which points out the lack of reliable evaluation measures which measure the performance of radar classifiers beyond accuracy. In the next part of this thesis, we study general uncertainty methods to remedy these issues which deep learning classifiers are notorious for in all data modalities, as well as present ways in which the uncertainty estimation of classifiers can be quantified. The following part also consists of two novel general uncertainty estimation techniques which can improve the reliability of classifier predictions, and in Part III, we discuss the effects of such over-confidence and mis-calibration for radar classifiers and also present a novel solution for addressing these issues in radar classifiers.

Part II.

Uncertainty calibration in deep learning

Chapter 6.

Uncertainty calibration: problem formulation and literature review

Autonomous systems rely on highly-accurate classifiers for decision-making, as well as their predictive uncertainty estimates for judging the reliability of the predictions. In this chapter, we introduce uncertainty quantification and uncertainty calibration, as well as provide the background information needed for the two succeeding chapters (Chap. 7 and 8), where we present two novel approaches for improving the uncertainty calibration for DNN classifiers.

6.1. Uncertainty quantification

Deep neural networks (DNNs) have achieved spectacular success in classification tasks when trained on very large, but still finite training sets. DNN training mostly follows the principle of Empirical Risk Minimization (ERM) [Vapnik, 1992], which states that by minimizing the training error the classifier will generalize to previously unseen data, under the condition that novel data points and labels are drawn from the same distribution as the training data. Although this assumption works remarkably well on difficult benchmark datasets such as ImageNet [Russakovsky et al., 2015], the assumption of identically distributed training and test sets is likely to be violated in DNN-based systems deployed in real-world situations. Knowing when a DNN can or cannot be trusted because of dataset shift is of utmost importance whenever DNNs should be used in safety-critical applications [Ovadia et al., 2019; Meinke & Hein, 2020], such as automated driving, robotics, surveillance, or medical diagnosis. At the same time, there can be true ambiguity in the data, e.g. when human annotators cannot agree or make mistakes [Peterson et al., 2019; Desai & Durrett, 2020; Jiang et al., 2020], when inputs are occluded or corrupted [Hendrycks & Dietterich, 2019], or whenever environmental conditions prevent a conclusive classification, e.g. due to challenging light or weather conditions [Hendrycks & Dietterich, 2019]. Such situations require DNNs that do not just predict the most likely class, but also quantify the uncertainty or confidence of their prediction, thereby allowing decision-making systems to take the risk caused by perceptual uncertainty into account.

Recent years have seen a number of techniques for estimating the uncertainty of DNN predictions with many formulated in the context of Bayesian statistics. Instead of point-estimates of the network weights, Bayesian neural networks (BNNs) [Blundell et al., 2015] find distributions over these network weights which offer an understanding of what the models know and do not know. The exact Bayesian inference of DNNs is intractable as the number of parameters is very large and the functional form of a neural network does not easily allow exact integration, therefore Bayesian methods often rely on BNN approximations [Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017]. These approximations often offer high-quality uncertainty estimates but are computationally demanding solutions which are not practical for decision making systems requiring real-time estimates. One practical alternative for uncertainty estimation is uncertainty calibration which offers computationally simple and real-time solutions, and have shown to be effective for large high-capacity DNNs. From a sample-based frequentist inference perspective, calibration is a part of uncertainty quantification which strives to improve the accuracy of the predicted confidences.

6.2. Uncertainty calibration

Despite great ability in learning discriminative features, DNN classifiers often make overconfident [Hein et al., 2019] and mis-calibrated [Guo et al., 2017] predictions. Both these notorious characteristics of DNNs can be addressed by uncertainty calibration techniques which output calibrated predictions. A multi-class classifier is well-calibrated if among the samples receiving the prediction distribution $\mathbf{q}$, the ground truth class distribution is also $\mathbf{q}$. The mismatch between the prediction and ground truth distribution quantifies the degree of mis-calibration and is typically measured using the expected calibration error (ECE) [Naeini et al., 2015; Guo et al., 2017; Kull et al., 2019]. The task of minimizing this mis-match (i.e. improving the calibration) can be solved using two types of calibration techniques: post-hoc calibration and during-training calibration. The former achieves the goal by refining the confidences of trained DNNs, whereas the latter involves modification of the training process.

6.2.1. Post-hoc calibration

Since the pioneering work of Guo et al. [2017], post-hoc calibration has been widely acknowledged as an efficient multi-class calibration solution for modern DNNs. This set of solutions involve refining the predicted confidences of a *trained* classifier, by learning a calibration function from a small held-out validation set (also called the calibration set). Such calibration functions, which are computationally cheap and demand very little additional memory, can easily be added as an additional layer after the computations of the DNN predictions. These solutions do not require re-training and are, therefore, the best option for trained classifiers which are already used in practice. One limitation of post-hoc calibration is its inability to leverage the large modelling capacity of DNNs to significantly change the confidences of the classifiers. Given the relatively simple nature of the calibration functions and the small calibration sets used for learning, the improvements heavily depend on the output distribution of the classifiers and can be limited to classifiers which are already well-calibrated. Despite this limitation, to date some post-hoc calibration methods effectively and consistently offer the best calibration performance.

6.2.2. During-training calibration

An alternative approach is to modify the training process of DNNs to directly or implicitly optimize the uncertainty calibration. These include regularization techniques [Pereyra et al., 2017; Zhang et al., 2018], data augmentation [Thulasidasan et al., 2019; Yun et al., 2019; Hendrycks et al., 2020], modification of the training labels [Mueller et al., 2019; Hinton et al., 2015] or loss [Kumar et al., 2018; Mukhoti et al., 2020; Maronas & Paredes, 2020]. In addition to improving the calibration performance, these methods may also enjoy performance gains on alternative generalization (e.g. accuracy) or reliability (e.g. adversarial robustness) measures. For example, most post-hoc calibration techniques do not at all or only marginally improve the accuracy performance, whereas during-training calibration techniques have shown to also improve the generalization performance [Pereyra et al., 2017; Thulasidasan et al., 2019]. In general, we view post-hoc and during-training calibration as two orthogonal ways to improve the calibration, as they can be easily combined.

In this part, we offer novel general solutions for both categories of calibration techniques in Chap. 7 (Post-hoc: I-Max) and Chap. 8 (During-training: OMADA) for improving the calibration performance of all types of classification models, regardless of their data modality. In the latter chapter, we also show their effectiveness after combining post-hoc and during-training calibration. In Part III, we apply the post-hoc calibrator, I-Max, to radar classifiers from Part I and, inspired by OMADA, also present a novel during-training calibration technique for radar classifiers.

6.3. Measuring uncertainty calibration using the expected calibration error (ECE)

We address supervised multi-class classification tasks, where each input $\mathbf{x} \in \mathcal{X}$ belongs to one of K classes, and the ground truth labels are one-hot encoded, i.e., $\mathbf{y} = [y_1, y_2, \ldots, y_K] \in \{0,1\}^K$[1]. Let $f : \mathcal{X} \mapsto [0,1]^K$ be a DNN trained using the cross-entropy loss. It maps each $\mathbf{x}$ onto a probability vector $\mathbf{q} = [q_1, \ldots, q_K] \in [0,1]^K$, which is used to rank the K possible classes of the current instance, e.g., $\arg\max_k q_k$ being the top-1 ranked class. The ranking performance of $\{q_k\}$ determines the classifier's top-k prediction accuracy. Besides ranking, $\{q_k\}$ are frequently interpreted as an approximation to the ground truth prediction distribution. As the trained classifier tends to overfit to the cross-entropy loss rather than the accuracy (i.e., $0/1$ loss), $\mathbf{q}$ as the prediction distribution is typically poorly calibrated. A post-hoc calibrator $h \in \{h_1, h_2, \ldots, h_K\}$ to revise $\mathbf{q}$ can deliver an improved performance. To evaluate the calibration performance of $h \circ f$, class-wise expected calibration error ($_{\mathrm{CW}}$ECE) averaged over the K classes is a common metric, measuring the expected deviation of the predicted per-class confidence after calibration, i.e., $h_k(\mathbf{q})$, from the ground truth probability $p(y_k = 1|h(\mathbf{q}))$[2]:

$$ {}_{\mathrm{CW}}\mathrm{ECE}(h \circ f) = \frac{1}{K} \sum_{k=1}^{K} E_{\mathbf{q}=f(\mathbf{x})} \left\{ \Big| p(y_k = 1|h(\mathbf{q})) - h_k(\mathbf{q}) \Big| \right\}. \tag{6.1} $$

When h is a binning scheme, $h_k(\mathbf{q})$ is discrete and thus repetitive. We can then empirically set $p(y_k = 1|h(\mathbf{q}))$ as the frequency of label-1 samples (i.e. samples with a ground truth label of $y_k = 1$) among those receiving the same $h_k(\mathbf{q})$. On the contrary, classifier predictions are continuous. It is unlikely that two samples attain the same $h_k(\mathbf{q})$, thus requiring additional quantization, i.e., applying histogram density estimation (HDE) for modeling the distribution of $h_k(\mathbf{q})$, or alternatively using kernel density estimation (KDE). It is noted that ideally we should compare the whole distribution $h(\mathbf{q})$ with the ground truth $p(\mathbf{y}|h(\mathbf{q}))$. However, neither HDE nor KDE scales well with the number of classes. Therefore, the multi-class ECE evaluation often boils down to the one-dimensional class-wise ECE as in (6.1) or the top-1 expected calibration error ($_{\mathrm{top1}}$ECE), i.e., $E\left[\left| p(y_{k=\arg\max_k h_k(\mathbf{q})} = 1|h(\mathbf{q})) - \max_k h_k(\mathbf{q}) \right| \right]$.

[1] Even though in this work we focus on single-label classification, this definition also includes multi-label classification and the methods proposed later can be easily extended to the multi-label setting.

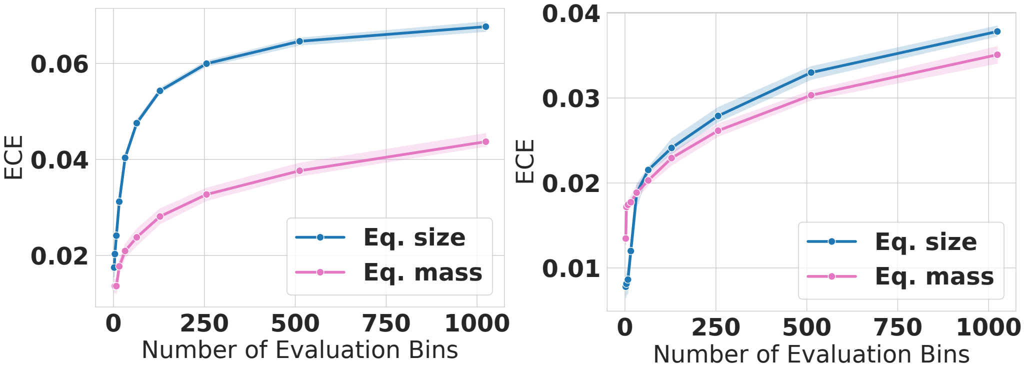
[2] The ground truth probability of correctness for the calibrated probabilities $h(\mathbf{q})$.

6.4. Challenges in evaluating calibration errors

The common practice of evaluating the calibration error of a classifier resorts to histogram density estimation (HDE) for modeling the distribution of the predictions. However, Vaicenavicius et al. [2019] proved that with a fixed number of evaluation bins the ECE is underestimated even with an infinite number of samples. This underestimation phenomena has also empirically been shown in Widmann et al. [2019]; Kumar et al. [2019]; Wenger et al. [2020]. This results in unreliable calibration solutions, as the true ECEs can be larger than their evaluations, putting many applications at risk.

To estimate the ECE from a set of evaluation samples, the continuous prediction probabilities need to be quantized/binned (i.e. HDE modelling). This discrete approximation of a continuous function introduces a bias-variance trade-off into the empirical estimate of the ECE. The evaluation of the ECE depends on the partitioning of the probabilities into disjoint bins using a binning scheme (i.e. h), where the mis-match between accuracy and predicted confidence of each bin is used as an approximation of the mis-calibration of the classifier. The binning scheme is defined by the number of evaluation bins and the bin edges, where the exact choice of both remains to be an open problem. Increasing the number of evaluation bins reduces the bias, as the evaluation quantization error is smaller, however, the estimation of the ground truth correctness begins to suffer from high variance. On the other hand, the correct choice between the two common variants of bin edges, Equal (Eq.) size (uniformly partitioning the probability interval $[0,1]$), and Eq. mass (uniformly distributing samples over bins) binning, also remains unclear.

Fig. 6.1 shows that the empirical (a)$_{\text{CW}}$ECEand (b)$_{\text{top1}}$ECEestimates of the network predictions are sensitive to the number of evaluation bins and the choice for bin edges. It remains unclear how to optimally choose the number of evaluation bins and bin edges which minimizes the estimation error.



(a) $_{\text{CW}}$ECE

(b) $_{\text{top1}}$ECE

Figure 6.1.: The (a) $_{\text{CW}}$ECEand (b) $_{\text{top1}}$ECEevaluations of a CIFAR-10 Wide ResNet (WRN) classifier, evaluated across multiple evaluation bins using equal size (Eq. size) and equal mass (Eq. mass) bin edges. We observe that the ECEs significantly increase with more evaluation bins and, additionally, also vary for the choice of the bin edges.

6.5. Alternative measures for calibration error

Since the initial work which introduced the ECE metric [Naeini et al., 2015; Guo et al., 2017], there have been a number of works which have proposed alternative ways for evaluating the calibration error. Some proposals are *minor* variants of the ECE such as adaptive calibration error (ACE) [Nixon et al., 2019] (which uses Eq. mass binning instead of Eq. size commonly used by the ECE) and thresholding-based ECE [Nixon et al., 2019] (which applies a threshold to filter out low-confident predictions). An alternative binning-based metric ($\text{ECE}_{\text{sweep}}$) was proposed in Roelofs et al. [2020], which offers a heuristic for the number of bins used for HDE modelling in order to reduce the bias in the estimation of the calibration error. Recent work [Zhang et al., 2020; Widmann et al., 2019] suggested to use kernel density estimation (KDE) for calibration error estimation as an alternative to binning-based metrics (i.e. HDE). However, the choice of the kernel and bandwidth also remains unclear, and the smoothness of the ground truth distribution is hard to verify in practice. Additionally, [Gupta et al., 2020] also offered a binning-free calibration measure based on the classical Kolmogorov-Smirnov (KS) statistical test which compares cumulative probability distributions. This measure is very similar to the maximum calibration error (MCE) [Naeini et al., 2015; Guo et al., 2017] which, unlike the ECE which takes the weighted average across all bins, computes the maximum error across all bins. Both the MCE and KS measures compute a worst-case deviation between confidence and accuracy.

Other uncertainty metrics, such as the negative log-likelihood (NLL) and Brier score, are also commonly used in the literature, as proxies, for the calibration errors. However, these metrics are not direct measures of the calibration performance but rather alternative measures for the quality of the predicted confidences and how well they are able to reproduce the hard ground truth labels. The NLL is a cross entropy measure which computes the divergences between the predictive and ground truth distributions, whereas the Brier score is the mean-squared-error of the prediction and label vectors. It should be noted that both measures do not penalize over-confidence in the predictions as much as the ECE.

As a cross entropy measure between two distributions, the NLL would be an ideal metric for calibration evaluation. However, empirical NLL and Brier favor high accuracy and high confident classifiers, as each sample only having one hard label essentially implies the maximum confidence on a single class. For this reason, during training, the empirical NLL loss will keep pushing the prediction probability to one even after reaching 100% training set accuracy. As a result, trained classifier often show poor calibration performance at test time [Guo et al., 2017]. In contrast to NLL/Brier, empirical ECEs use hard labels differently. The ground truth correctness associated to the prediction confidence p is estimated by averaging over the hard labels of the samples receiving the prediction probability p or close to p. Due to averaging, the empirical ground truth correctness is usually not a hard label. Lastly, we use a small example to show the difference between the NLL/Brier and ECE: for N predictions, all assigned a confidence of 1.0 and containing M mistakes, the calibrated confidence is $M/N < 1$. Unlike ECE, the NLL/Brier loss is only non-zero only for the M wrong predictions, despite all N predictions being miscalibrated. This example shows that NLL/Brier penalize miscalibration far less than ECE.

6.6. Literature review

Confidence calibration is an active research topic in deep learning. In this section we provide a literature review of uncertainty estimation for deep neural networks (DNNs), from resource-demanding Bayesian methods to augmentation strategies, as well as simple post-hoc calibration techniques.

6.6.1. Uncertainty quantification

An effective technique for uncertainty estimation in neural networks involves adopting Bayesian inference. Instead of treating the model weights as point-estimates, the weights of Bayesian neural networks (BNNs) are assumed to be random variables and modelled as distributions [Mullachery et al., 2018]. Blundell et al. [2015] proposed an algorithm to quantify the uncertainty by using unbiased gradient estimates of the cost function (i.e. compression cost) learning distributions over the weights. This algorithm was also extended to Bayesian recurrent neural networks [Fortunato et al., 2017] using a simple adaptation of the truncated back-propagation through time, as well as Bayesian 3D convolutional networks [de la Riva & Mettes, 2019]. A thorough review, and introduction, on BNNs can be found in Goan & Fookes [2020].

Bayesian techniques have shown to yield better confidence estimates than point-estimate DNNs [Liu et al., 2019; Wu et al., 2019], though the scalability of BNNs to larger parameter spaces, such as those of DNNs, have limited their use. Therefore, given the difficulty in inferring the model posterior in a BNN due to prohibitive computational costs, the exact posterior inference is often approximated.

One such approximation was proposed in Kristiadi et al. [2020], where the authors theoretically showed that limiting only the last fully connected linear layer in a model to being *Bayesian* is sufficient to maintain the good uncertainty estimates of BNNs. Another lightweight approximation was presented in Gast & Roth [2018], where the authors proposed to model the activations as distributions instead of the weights via assumed density filtering.

A commonly-used, simple and effective alternative approximation using dropout [Srivastava et al., 2014] was proposed in Gal & Ghahramani [2016], MC-Dropout. Unlike dropout [Srivastava et al., 2014], which was only proposed as a regularization technique during *training*, the stochastic forward passes of the neural network caused by the dropout layers are kept switched on during test time for Monte Carlo (MC) integration. MC-Dropout has been used to approximate the uncertainty estimation in a number of applications such as semantic segmentation [Kendall & Gal, 2017], medical imaging [Filos et al., 2019; Roy et al., 2019], emotion prediction [Harper & Southern, 2020], aircraft trajectory prediction [Pang & Liu, 2020] and generative modeling [Miok et al., 2019]. The authors of Meronen et al. [2020] introduced a novel non-linear activation function and used it in conjunction with MC-Dropout, showing better performance than using MC-Dropout with, the commonly used, ReLU networks [Hein et al., 2019] for uncertainty estimation [Foong et al., 2019; Ovadia et al., 2019]. In order to adopt MC-Dropout for real-time applications, a low-latency single-shot approximation was proposed in Brach et al. [2020] through an analytical approximation of each layer in a fully connected network. The authors of Huang et al. [2018] proposed to use consecutive frames from the past to approximate multiple forward passes into a single-shot real-time adaptation of MC-Dropout.

Similar to MC-Dropout, an alternative Bayesian approximation for uncertainty estimation employs deep ensembles [Lakshminarayanan et al., 2017]. Enhancing predictive performance through exploiting a number of trained models with Bayesian model averaging, ensemble approaches have shown to greatly improve the quality of the uncertainty estimation, in addition to the generalization performance. Larger ensembles, as well as encouraging larger overall diversity among the trained models in the ensemble, have shown to yield higher quality uncertainty estimates [Jain et al., 2020]. Ensemble techniques have shown superior performance over MC-Dropout in terms high-quality uncertainty estimates [Gustafsson et al., 2020], with MC-Dropout offering a less memory intensive but slower solution compared to ensembles. Nonetheless, ensembles involve additional memory and computational costs which are not feasible for many real-time applications.

With the goal of distilling an ensemble into a single efficient model, ensemble distribution distillation [Malinin et al., 2020] was proposed to reduce the costs but maintain the performance of deep ensembles. Other similar approaches also using a Dirichlet distribution to model the uncertainty of the classifier outputs have also been employed [Malinin & Gales, 2018; Sensoy et al., 2018; Malinin & Gales, 2019]. By enforcing low confidences for unfamiliar data distributions, a robust optimization technique, similar to adversarial training [Metzen et al., 2017], was proposed in Hein et al. [2019]; and a similar approach with probable guarantees in Meinke & Hein [2020].

Bayesian DNNs and their approximations, e.g. Blundell et al. [2015]; Kingma et al. [2015]; Louizos & Welling [2017]; Gal & Ghahramani [2016]; Lakshminarayanan et al. [2017] are resource-demanding methods to consider predictive model uncertainty. Such methods require exhaustive sampling from the prior distributions of Bayesian networks [Blundell et al., 2015], require multiple forward passes of the same input [Gal & Ghahramani, 2016] or require multiple classifiers simultaneously classifying the same input. However, applications with limited complexity overhead and latency require sampling-free and single-model based uncertainty methods. Examples of such real-time uncertainty estimation techniques which yield high quality uncertainty estimates include modifying the training loss [Kumar et al., 2018], scalable Gaussian processes [Milios et al., 2018], robust optimization technique [Hein et al., 2019], sampling-free uncertainty estimation [Postels et al., 2019], data augmentation [Thulasidasan et al., 2019; Yun et al., 2019; Hendrycks et al., 2020], and ensemble distribution distillation [Malinin & Gales, 2019; Malinin et al., 2020].

6.6.2. Post-hoc calibration

In comparison, a simple approach that requires no retraining of the models is post-hoc calibration [Guo et al., 2017]. Prediction probabilities (logits) scaling and binning are the two main solutions for post-hoc calibration. Scaling methods use parametric or non-parametric models to adjust the raw logits. Guo et al. [2017] investigated linear models, ranging from the single-parameter based temperature scaling (TS) to more complicated vector/matrix scaling. To avoid overfitting, Kull et al. [2019] suggested to regularize matrix scaling with a L_2 loss on the model weights. Recently, Wenger et al. [2020] adopted a latent Gaussian process for multi-class calibration. Ji et al. [2019] extended TS to a bin-wise setting, by learning separate temperatures for various confidence subsets. To improve the expressive capacity of TS, an ensemble of temperatures were adopted by Zhang et al. [2020]. Owing to continuous outputs of scaling methods, one critical issue discovered in the recent work is: Their empirical ECE estimate is not only non-verifiable [Kumar et al., 2019], but also asymptotically smaller than the ground truth [Vaicenavicius et al., 2019]. This means that using a *small* number of bins (e.g. 10 or 15, which is commonly used in the literature and proposed in Guo et al. [2017]) for large dataset sizes can lead to evaluations which are better than the actual model mis-calibration. Recent work [Zhang et al., 2020; Widmann et al., 2019] exploited KDEs for an improved ECE evaluation, however, the parameter setting requires further investigation. Nixon et al. [2019] and Ashukha et al. [2020] discussed potential issues of the ECE metric, and the former suggested to 1) use equal mass binning for ECE evaluation; 2) measure both top-1 and class-wise ECE to evaluate multi-class calibrators, 3) only include predictions with a confidence above some epsilon in the class-wise ECE score.

As an alternative to scaling, histogram binning (HB) quantizes the raw confidences with either Eq. size or Eq. mass bins [Zadrozny & Elkan, 2001]. It offers asymptotically convergent ECE estimation [Vaicenavicius et al., 2019], but is less sample efficient than scaling methods and also suffers from accuracy loss [Guo et al., 2017]. Kumar et al. [2019] proposed to perform scaling before binning for an improved sample efficiency. Isotonic regression [Zadrozny & Elkan, 2002] and Bayesian binning into quantiles (BBQ) [Naeini et al., 2015] are often viewed as binning methods. However, their ECE estimates face the same issue as scaling methods: though isotonic regression fits a piecewise linear function, its predictions are continuous as they are interpolated for unseen data. BBQ considers multiple binning schemes with different numbers of bins, and combines them using a continuous Bayesian score, resulting in continuous predictions.

In Chap. 7, we propose a novel binning-based calibrator which will extend the post-hoc calibration literature with a real-time uncertainty estimation technique. The work identifies the short-comings of histogram binning calibration [Zadrozny & Elkan, 2001; Guo et al., 2017], and remedies the problem with a mutual information objective for bin edge optimization.

6.6.3. During-training calibration

Data augmentation is a pragmatic technique to improve deep learning generalization performance. The recently proposed *Mixup* [Zhang et al., 2018] creates new training samples by linear interpolation in the image space between a random pair of training samples. In addition it also creates *soft* labels by linearly interpolating between the original one-hot label vectors. In Thulasidasan et al. [2019], it was shown that Mixup not only improves generalization, but also yields well-calibrated softmax scores, and less confident predictions for out-of-distribution data. A recent variant of Mixup is Manifold Mixup [Verma et al., 2019], which performs Mixup in the feature space of a DNN instead of the image space. A downside of these methods is that they may generate unrealistic samples that lie off the true data manifold. Furthermore, the labels are generated by interpolating between two or more hard label vectors, and may thus not reflect true ambiguity, e.g. if the image obtained by interpolation is more similar to a third class.

Soft labels were also used to improve generalization via ϵ-*smoothing* [Szegedy et al., 2016], where a probability mass of size ϵ is distributed over all but the correct class, thus penalizing over-confident predictions. Another simple and effective method to avoid over-confidence on outliers is to include out-of-distribution samples with uniform labels in the training set [Hein et al., 2019; Lee et al., 2018]; the samples can even be as simple as uniform noise images.

For studying adversarial robustness, the authors of Stutz et al. [2019] introduced the concept of on- and off-manifold adversarial examples. Augmenting the training set with on-manifold adversarial examples is particularly useful to improve the generalization performance. However, common perturbations in the image space, including the above-mentioned Mixup, Manifold Mixup, and additive random noise, are not constrained to the data manifold. In Chap. 8, the proposed method, OMADA, extends elements of recent successful approaches for uncertainty estimation, data augmentation, and adversarial training. In this work we are interested in the use of data augmentation for uncertainty calibration.

In the following two chapters, we aim at filling literature gap LG3 identified in Sec. 2.4.2. The goal of filling this gap is to tackle the over-confidence [Hein et al., 2019] and mis-calibration [Guo et al., 2017] issues identified for DNNs. These chapters focus on uncertainty calibration [Guo et al., 2017] as a means to improve the quality of the uncertainty estimates, through both post-hoc calibration (in Chap. 7) and during-training calibration (in Chap. 8). In Chap. 7, we additionally aim to address the challenges, identified in the literature [Kumar et al., 2019; Ashukha et al., 2020], in evaluating the calibration error of deep learning classifiers. With the combination of the two novel solutions presented in Chapters 7 and 8 we aim to offer competitive *real-time uncertainty calibration* (LG3) techniques for any deep learning classifier.

Chapter 7.

I-Max: Multi-class uncertainty calibration via mutual information-based binning

Contribution: Post-hoc multi-class calibration is a common approach for providing high-quality confidence estimates of deep neural network predictions. Recent work has shown that widely used scaling methods underestimate their calibration error, while alternative Histogram Binning (HB) methods often fail to preserve classification accuracy. We identify, discuss and resolve the identified issues of HB in order to provide calibrated confidence estimates using only a small holdout calibration dataset for bin optimization while preserving multi-class ranking accuracy. From an information-theoretic perspective, we derive the *I-Max* concept for binning, which maximizes the mutual information between labels and quantized logits. This concept mitigates potential loss in ranking performance due to lossy quantization, and by disentangling the optimization of bin edges and representatives allows simultaneous improvement of ranking and calibration performance. To improve the sample efficiency and estimates from a small calibration set, we propose a novel *shared class-wise* (sCW) calibration strategy, sharing one calibrator among similar classes (e.g., with similar class priors) so that the training sets of their class-wise calibration problems can be merged to train the single calibrator. The combination of sCW and I-Max binning outperforms the state of the art calibration methods on various evaluation metrics across different benchmark datasets and models, using a small calibration set (e.g., 1k samples for ImageNet).

7.1. Post-hoc uncertainty calibration

Post-hoc calibration techniques offer a simple and computationally cheap solution for improving the calibration of modern DNNs. The literature of post-hoc calibration mostly include continuous-output calibrators, such as temperature scaling (TS) [Guo et al., 2017], to refine the predicted confidences (often referred to as scaling methods). In Sec. 6.4, we showed that evaluating the expected calibration error (ECE) of continuous-output distributions (e.g. that of scaling methods) can be difficult because it requires histogram density estimation (HDE) or kernel density estimation (KDE) for modeling the predictive distributions.

An alternative technique for post-hoc calibration is Histogram Binning (HB) [Zadrozny & Elkan, 2001; Guo et al., 2017; Kumar et al., 2019], which is a quantized-output calibrator using bin edges and bin representations to perform quantization. Here, HB refers to a calibration method and is different to the histogram density estimation (HDE) used for evaluating ECEs of scaling methods. HB produces discrete predictions, whose probability mass functions can be empirically estimated without using HDE/KDE. Therefore, compare to the ECE estimate of the Baseline classifier (i.e. with no post-hoc calibration) and TS, the ECE of the HB methods remain constant and unaffected by the number of evaluation bins (as an additional HDE step does not affect the already quantized predictions) in Fig. 7.1a) and it can converge to the true value with increasing evaluation samples [Vaicenavicius et al., 2019], see Fig. 7.1b).

The most common variants of HB are Equal (Eq.) size (uniformly partitioning the probability interval $[0,1]$), and Eq. mass (uniformly distributing samples over bins) binning. Based on empirical results using 10 evaluation bins for ECE estimation, Guo et al. [2017] concluded that HB underperforms compared to scaling (can also be seen in Fig. 7.1a for small number of evaluation bins). These methods for multi-class calibration are known to degrade accuracy, since quantization through binning may remove a considerable amount of label information contained by the classifier's outputs.

In this work, we show that the key for HB to retain the accuracy of trained classifiers is choosing bin edges which minimize the amount of label information loss. For this, both Eq. size and mass binning are suboptimal. We present *I-Max*, a novel iterative method for optimizing bin edges with proved convergence. As the location of its bin edges inherently ensures sufficient calibration samples per bin, the bin representatives of I-Max can then be effectively optimized for calibration. Two design objectives, calibration and accuracy, are thus nicely disentangled under I-Max. For multi-class calibration, I-Max adopts the one-vs-rest (OvR) strategy to individually calibrate the prediction probability of each class. To cope with a limited number of calibration samples, we propose to share one binning scheme for calibrating the prediction probabilities of similar classes, e.g., with similar class priors or belonging to the same class category. At small data regime, we can even choose to fit one binning scheme on the merged training sets of all per-class calibrations. Such a shared class-wise (sCW) calibration strategy greatly improves the sample efficiency of I-Max binning.

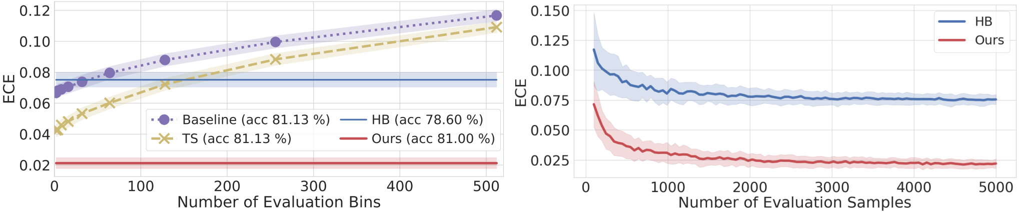
I-Max is evaluated according to multiple performance metrics, including accuracy, ECE, Brier and NLL, and compared against benchmark calibration methods across multiple datasets and trained classifiers. For ImageNet, I-Max obtains up to 66.11% reduction in ECE compared to the baseline and up to 38.14% reduction compared to the state-of-the-art GP-scaling method [Wenger et al., 2020].

7.2. Histogram binning (HB) calibration

Here, we introduce the I-Max binning scheme, which addresses the issues of HB in terms of preserving label-information in multi-class calibration. Sec. 7.2.1 presents a sample-efficient technique for one-vs-rest calibration. In Sec. 7.2.2, we formulate the training objective of binning as MI maximization and derive a simple algorithm for I-Max binning.

7.2.1. One-vs-Rest (OvR) strategy for multi-class calibration

HB was initially developed for two-class calibration. When dealing with multi-class calibration, it separately calibrates the prediction probability q_k of each class in a *one-vs-rest* (OvR) fashion: For any class-k, HB takes y_k as the binary label for a two-class calibration task in which the class-1 means $y_k = 1$ and class-0 collects all other $K-1$ classes. It then revises the prediction probability q_k of $y_k = 1$ by mapping its logit $\lambda_k \triangleq \log q_k - \log(1-q_k)$ onto a given number



(a) $_{\text{top1}}$ECE(5k evaluation samples) **(b)** $_{\text{top1}}$ECEconverging curve (based on 10^2 bootstraps)

Figure 7.1.: (a) Temperature scaling (TS), equally sized-histogram binning (HB), and our proposal, i.e., sCW I-Max binning are compared for post-hoc calibrating a CIFAR100 (WRN) classifier. (b) Binning offers a reliable ECE measure as the number of evaluation samples increases.

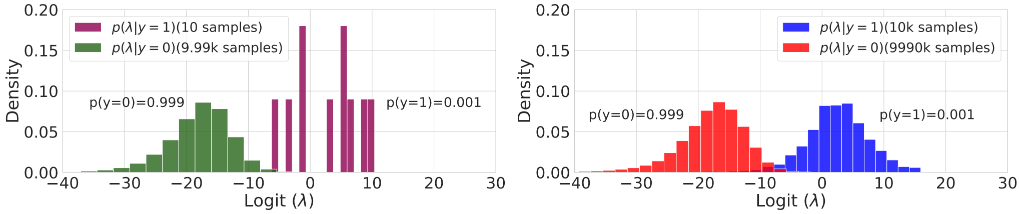
of bins, and yielding the calibrated prediction probability. Here, we choose to bin the logit λ_k (logit vector $\lambda = [\lambda_1, \ldots, \lambda_K] \in [0,1]^K$) instead of q_k, as the former is unbounded, i.e., $\lambda_k \in \mathbb{R}$, which eases the bin edge optimization process. Nevertheless, as q_k and λ_k have a monotonic bijective relation, binning q_k and λ_k are equivalent. It is further noted that after K class-wise calibrations we do not perform the extra normalization step as in Guo et al. [2017]. After OvR marginalizes the multi-class predictive distribution, each class shall then be treated independently (see Sec. 7.7.3).

The calibration performance of HB depends on the setting of its bin edges and representatives. From a calibration set $C = \{(\mathbf{y},\lambda)\}$, we can construct K training sets, i.e., $S_k = \{(y_k, \lambda_k)\}\,\forall k$, under the one-vs-rest strategy, and then optimize the class-wise (CW) HB over each training set. As two common solutions in the literature, Eq. size and Eq. mass binning focus on bin representative optimization. Their bin edge locations, on the other hand, are either fixed (independent of the calibration set) or only ensure a balanced training sample distribution over the bins. After binning the logits in the calibration set $S_k = \{(y_k, \lambda_k)\}$, the bin representatives are set as the empirical frequencies of samples with $y_k = 1$ in each bin. To improve the sample efficiency of bin representative optimization, Kumar et al. [2019] proposed to perform scaling-based calibration before HB. Namely, after properly scaling the logits $\{\lambda_k\}$, the bin representative per bin is then set as the averaged sigmoid-response[1] of the scaled logits in S_k belonging to each bin.

However, pre-scaling does not resolve the sample inefficiency issue arising from a small class prior p_k.The two-class ratio in S_k is $p_k : 1 - p_k$. When p_k is small we will need a large calibration set $C = \{(\mathbf{y},\lambda)\}$ to collect enough class-1 samples in S_k for setting the bin representatives. For ImageNet with 1k classes, S_k constructed from a class-balanced C of size 10k contains only $10\text{k}/K = 10$ class-1 samples, see Fig. 7.2-a). Too few class-1 samples will compromise the optimization of the calibration schemes. To address this, we propose to merge $\{S_k\}$ across similar classes and then use the merged set S for HB training, yielding one binning scheme shareable to multiple per-class calibration tasks, i.e., *shared* class-wise (sCW) binning instead of CW binning respectively trained on S_k. The size of the resulting merged set S will be $10\text{k} \times K = 10\text{m}$ and will contain 10k class-1 samples from the K different classes (K times larger than S_k), see Fig. 7.2-b) if all K classes S_k are merged together. In Sec. 7.7, we respectively experiment using a single binning scheme for all classes in the balanced multi-class setting, and sharing one binning among the classes with similar class priors in the imbalanced multi-class setting. Note, both S_k and S serve as empirical approximations to the inaccessible ground truth distribution

[1] $\sigma(x) = \frac{1}{1+e^{-x}}$

$p(y_k, \lambda_k)$ for bin optimization. The former suffers from high variances, arising from insufficient samples (Fig. 7.2-a), while the latter is biased due to having samples drawn from the other classes (Fig. 7.2-b). As the calibration set size is usually small, the variance is expected to outweigh the approximation error over the bias (see an empirical analysis in Sec. A1).



(a) training set $S_{k=394}$ for CW binning.

(b) training set S for *shared*-CW binning.

Figure 7.2.: Histogram of ImageNet (InceptionResNetv2) logits for (a) CW and (b) sCW training. By means of the set merging strategy to handle the two-class imbalance $1:999$, S has K=1000 times more class-1 samples than S_k with the same 10k calibration samples from C.

7.2.2. Bin optimization via mutual information (MI) maximization

Binning can be viewed as a quantizer Q that maps the real-valued logit $\lambda \in \mathbb{R}$ to the bin interval $m \in \{1, \ldots, M\}$ if $\lambda \in \mathcal{I}_m = [g_{m-1}, g_m)$, where M is the total number of bin intervals, and the bin edges g_m are sorted ($g_{m-1} < g_m$, and $g_0 = -\infty, g_M = \infty$). Any logit binned to $\mathcal{I}_m$ will be reproduced to the same bin representative r_m. In the context of calibration, the bin representative r_m assigned to the logit λ_k is used as the calibrated prediction probability of the class-k. As multiple classes can be assigned with the same bin representative, we will encounter ties when making top-k predictions based on calibrated probabilities. Therefore, binning as lossy quantization generally does not preserve the raw logit-based ranking performance, being subject to potential accuracy loss. In order to mitigate this accuracy loss, the quantization error can be reduced by increasing the number of bins or designing the quantization such that the amount of accuracy loss is significantly reduced.

Unfortunately, increasing M to reduce the quantization error is not a good solution here. For a given calibration set, the number of samples per bin generally reduces as M increases, and a reliable frequency estimation for setting the bin representatives $\{r_m\}$ demands sufficient samples per bin. Therefore, we consider optimizing the quantization step to tackle the accuracy drop using mutual information. Intuitively, a high mutual information between the ground truth labels and quantized outputs mean that the quantizer (i.e. a process including some information loss) has chosen to prioritize preserving information about the label (i.e. the quantized outputs are highly dependent on the labels which is required to preserve accuracy).

Considering that the top-k accuracy reflects how well the ground truth label can be recovered from the logits, we propose bin optimization via maximizing the MI between the quantized logits $Q(\lambda)$ and the label y

$$\{g_m^*\} = \arg \max_{Q\colon \{g_m\}} I(y; m = Q(\lambda)) \overset{(a)}{=} \arg \max_{Q\colon \{g_m\}} H(m) - H(m|y) \tag{7.1}$$

where the index m is viewed as a discrete random variable with $P(m|y) = \int_{g_{m-1}}^{g_m} p(\lambda|y) \mathrm{d}\lambda$ and $P(m) = \int_{g_{m-1}}^{g_m} p(\lambda) \mathrm{d}\lambda$, and the equality (a) is based the relation of MI to the entropy $H(m)$

and conditional entropy $H(m|y)$ of m. Such a formulation offers a quantizer Q^* optimal at preserving the label information for a given budget on the number of bins. Unlike designing distortion-based quantizers, the reproducer values of raw logits, i.e., the bin representatives $\{r_m\}$, are not a part of the optimization space, as it is sufficient to know the mapped bin index m of each logit. Once the bin edges $\{g_m^*\}$ are obtained, the bin representative r_m to achieve zero calibration error shall equal $P(y=1|m)$, which can be empirically estimated from the samples within the bin interval $\mathcal{I}_m$.

It is interesting to analyze the objective function after the equality (a) in (7.1). The first term $H(m)$ is maximized if $P(m)$ is uniform, which is attained by Eq. mass binning. A uniform sample distribution over the bins is a sample-efficient strategy to optimize the bin representatives for the sake of calibration. However, it does not consider any label information, and thus can suffer from severe accuracy loss. Through MI maximization, we can view I-Max as revising Eq. mass by incorporating the label information into the optimization objective, i.e., having the second term $H(m|y)$. As a result, I-Max not only enjoys a well balanced sample distribution for calibration, but also maximally preserved label information for accuracy.

In the example of Fig. 7.3, the bin edges of I-Max binning are densely located in an area where the uncertainty of y given the logit is high. This uncertainty results from small gaps between the top class predictions. With small bin widths, such nearby prediction logits are more likely located to different bins, and thus distinguishable after binning. On the other hand, Eq. mass binning has a single bin stretching across this high-uncertainty area due to an imbalanced ratio between the $p(\lambda|y=1)$ and $p(\lambda|y=0)$ samples. Eq. size binning follows a pattern closer to I-Max binning. However, its very narrow bin widths around zero may introduce large empirical frequency estimation errors when setting the bin representatives.

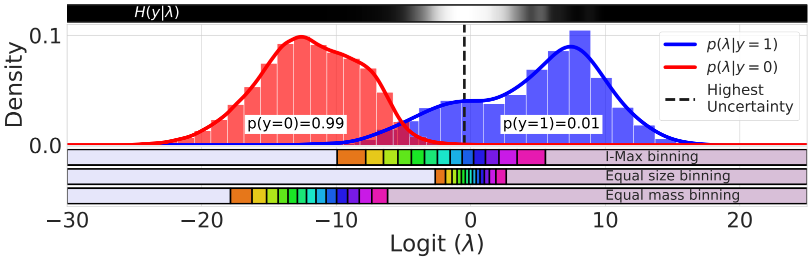


Figure 7.3.: Histogram and KDE of CIFAR100 (WRN) logits in S constructed from 1k calibration samples. The bin edges of Eq. mass binning with 15 bins are located at the high mass region, mainly covering class-0 due to the imbalanced two-class ratio 1 : 99. Both Eq. size and I-Max binning cover the high uncertainty region, but here only I-Max yields reasonable bin widths ensuring enough mass per bin. Note, Eq. size binning uniformly partitions the interval [0,1] in the probability domain. The observed dense and symmetric bin location around zero is the outcome of probability-to-logit translation.

For solving the problem (7.1), we formulate an equivalent problem.

Theorem 1. *The MI maximization problem given in (7.1) is equivalent to*

$$\max_{Q:\,\{g_m\}} I(y;m=Q(\lambda)) = -\min_{\{g_m,\phi_m\}} \mathcal{L}(\{g_m,\phi_m\}) \tag{7.2}$$

where the loss $\mathcal{L}(\{g_m,\phi_m\})$ *is defined as*

$$\mathcal{L}(\{g_m,\phi_m\}) \triangleq \sum_{m=0}^{M-1} \int_{g_m}^{g_{m+1}} p(\lambda) \sum_{y'\in\{0,1\}} P(y=y'|\lambda)\log\frac{P(y=y')}{\sigma\left((2y'-1)\phi_m\right)}\mathrm{d}\lambda \tag{7.3}$$

and $\{\phi_m\}^2$ *as a set of real-valued auxiliary variables.*

Proof. See Sec. A2 for the proof. □

Next, we compute the derivatives of the loss $\mathcal{L}$ with respect to $\{g_m, \phi_m\}$. When the conditional distribution $P(y|\lambda)$ takes the sigmoid model, i.e., $P(y|\lambda) \approx \sigma[(2y-1)\lambda]$, the stationary points of $\mathcal{L}$, zeroing the gradients over $\{g_m,\phi_m\}$, have a closed-form expression

$$g_m = \log\left\{\frac{\log\left[\frac{1+e^{\phi_m}}{1+e^{\phi_{m-1}}}\right]}{\log\left[\frac{1+e^{-\phi_{m-1}}}{1+e^{-\phi_m}}\right]}\right\},\ \phi_m = \log\left\{\frac{\int_{g_m}^{g_{m+1}}\sigma(\lambda)p(\lambda)\mathrm{d}\lambda}{\int_{g_m}^{g_{m+1}}\sigma(-\lambda)p(\lambda)\mathrm{d}\lambda}\right\} \approx \log\left\{\frac{\sum_{\lambda_n\in\mathcal{S}_m}\sigma(\lambda_n)}{\sum_{\lambda_n\in\mathcal{S}_m}\sigma(-\lambda_n)}\right\}, \tag{7.4}$$

where the approximation for ϕ_m arises from using the logits in the calibration set S as an empirical approximation to $p(\lambda)$ and $S_m \triangleq \lambda_n \cap [g_m, g_{m+1})$. So, we can solve the problem by iteratively and alternately updating $\{g_m\}$ and $\{\phi_m\}$ based on (7.4) (see Algorithm 1 for pseudocode and Sec. A3 for the derivation of the closed form updates).

7.3. Convergence of iterative updates

In the following, we show that the updates on $\{g_m\}$ and $\{\phi_m\}$ according to (7.4) continuously decrease the loss $\mathcal{L}$, i.e.,

$$\mathcal{L}(\{g_m^l,\phi_m^l\}) \geq \mathcal{L}(\{g_m^{l+1},\phi_m^l\}) \geq \mathcal{L}(\{g_m^{l+1},\phi_m^{l+1}\}). \tag{7.5}$$

The second inequality is based on the explained property of $\mathcal{L}$. Namely, it is convex over $\{\phi_m\}$ and the minimum for any given $\{g_m\}$ is attained by $P_\sigma(y;\phi_m) = P(y|m)$. As ϕ_m is the log-probability ratio of $P_\sigma(y;\phi_m)$, we shall have

$$\phi_m^{l+1} \leftarrow \log\frac{P(y=1|m)}{P(y=0|m)} \tag{7.6}$$

where $P(y=1|m)$ in this case is induced by $\{g_m^{l+1}\}$ and $P(y|\lambda) = \sigma[(2y-1)\lambda]$. Plugging $\{g_m^{l+1}\}$ and $P(y|\lambda) = \sigma[(2y-1)\lambda]$ into Eq. 13.7 (of the proof in the appendix), the resulting $P(y=y'|m)$ at the iteration $l+1$ yields the update equation of ϕ_m as given in (7.4).

To prove the first inequality, we start from showing that $\{g_m^{l+1}\}$ is a local minimum of $\mathcal{L}(\{g_m,\phi_m^l\})$. The update equation on $\{g_m\}$ is an outcome of solving the stationary point equation of $\mathcal{L}(\{g_m,\phi_m^l\})$ over $\{g_m\}$ under the condition $p(\lambda = g_m) > 0$ for any m

$$\frac{\partial\mathcal{L}(\{g_m,\phi_m^l\})}{\partial g_m} = p(\lambda = g_m)\sum_{y'\in\{0,1\}} P(y=y'|\lambda=g_m)\log\frac{P_\sigma(y=y';\phi_m^l)}{P_\sigma(y=y';\phi_{m-1}^l)} \overset{!}{=} 0 \quad \forall m \tag{7.7}$$

[2] bin representations

Being a stationary point is the necessary condition of local extremum when the function's first-order derivative exists at that point, i.e., first-derivative test. To further show that the local extremum is actually a local minimum, we resort to the second-derivative test, i.e., if the Hessian matrix of $\mathcal{L}(\{g_m, \phi_m^l\})$ is positive definite at the stationary point $\{g_m^{l+1}\}$. Due to $\phi_m > \phi_{m-1}$ with the monotonically increasing function sigmoid in its update equation, we have

$$\left.\frac{\partial^2 \mathcal{L}(\{g_m, \phi_m^l\})}{\partial g_m \partial g_{m'}}\right|_{g_m = g_m^{l+1}\,\forall m} = 0 \quad \text{and} \quad \left.\frac{\partial^2 \mathcal{L}(\{g_m, \phi_m^l\})}{\partial^2 g_m}\right|_{g_m = g_m^{l+1}\,\forall m} > 0, \tag{7.8}$$

implying that all eigenvalues of the Hessian matrix are positive (equivalently, is positive definite). Therefore, $\{g_m^{l+1}\}$ as the stationary point of $\mathcal{L}(\{g_m, \phi_m^l\})$ is a local minimum.

It is important to note that from the stationary point equation (7.7), $\{g_m^{l+1}\}$ as a local minimum is unique among $\{g_m\}$ with $p(\lambda = g_m) > 0$ for any m. In other words, the first inequality holds under the condition $p(\lambda = g_m^l) > 0$ for any m. Binning is a lossy data processing. In order to maximally preserve the label information, it is natural to exploit all bins in the optimization, not wasting any single bin in the area without mass, i.e., $p(\lambda = g_m) = 0$. Having said that, it is reasonable to constrain $\{g_m\}$ with $p(\lambda = g_m) > 0\,\forall m$ over iterations, thereby concluding that the iterative method will converge to a local minimum based on the two inequalities (7.5).

A remark on the iterative method derivation

The closed-form update on $\{g_m\}$ in (7.4) is based on the sigmoid-model approximation, which has been validated through our empirical experiments. It is expected to work with properly trained classifiers that are not overly overfitting to the cross-entropy loss, e.g., using data augmentation and other regularization techniques at training. Nevertheless, even in corner cases that classifiers are poorly trained, the iterative method can still be operated without the sigmoid-model approximation. Namely, as shown in Fig. 7.3, we can resort to KDE for an empirical estimation of the ground truth distribution $p(\lambda|y)$. Using the KDEs, we can compute the gradient of $\mathcal{L}$ over $\{g_m\}$ and perform iterative gradient based update on $\{g_m\}$, replacing the closed-form based update. Essentially, the sigmoid-model approximation is only necessary to find the stationary points of the gradient equations, speeding up the convergence of the method. If attempting to keep the closed-form update on $\{g_m\}$, an alternative solution could be to use the KDEs for adjusting the sigmoid-model, e.g., $p(y|\lambda) \approx \sigma\left[(2y-1)(a\lambda + ab)\right]$, where a and b are chosen to match the KDE based approximation to $p(y|\lambda)$. After setting a and b, they will be used as a scaling and bias term in the original closed-form update equations

$$\begin{aligned} g_m &= \frac{1}{a}\log\left\{\frac{\log\left[\frac{1+e^{\phi_m}}{1+e^{\phi_{m-1}}}\right]}{\log\left[\frac{1+e^{-\phi_{m-1}}}{1+e^{-\phi_m}}\right]}\right\} - b \\ \phi_m &= \log\left\{\frac{\int_{g_m}^{g_{m+1}} \sigma(a\lambda+ab)p(\lambda)\mathrm{d}\lambda}{\int_{g_m}^{g_{m+1}} \sigma(-a\lambda-ab)p(\lambda)\mathrm{d}\lambda}\right\} \approx \log\left\{\frac{\sum_{\lambda_n \in \mathcal{S}_m} \sigma(a\lambda_n+ab)}{\sum_{\lambda_n \in \mathcal{S}_m} \sigma(-a\lambda_n-ab)}\right\} \end{aligned} \qquad \forall m. \tag{7.9}$$

7.4. I-Max connection to information bottleneck (IB)

IB [Tishby et al., 1999] is a generic information-theoretic framework for stochastic quantization design. Viewing binning as quantization, IB aims to find a balance between two conflicting goals: 1) maximizing the information rate, i.e., the mutual information between the label and the quantized logits $I(y; Q(\lambda))$; and 2) minimizing the compression rate, i.e., mutual information between the logits and the quantized logits $I(\lambda; Q(\lambda))$. It unifies them by minimizing

$$\min_{p(m|\lambda)} \frac{1}{\beta} I(\lambda; m = Q(\lambda)) - I(y; m = Q(\lambda)), \tag{7.10}$$

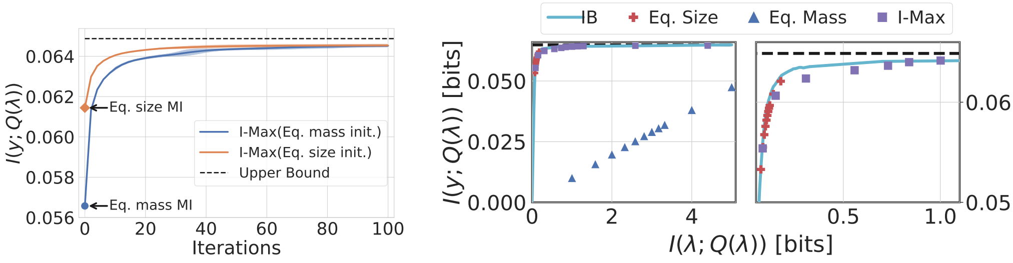
where m is the bin index assigned to λ and β is the weighting factor (with larger value focusing more on the information rate and smaller value on the compression rate). The compression rate is the bottleneck for maximizing the information rate. Note that IB optimizes the distribution $P(m|\lambda)$, which describes the probability of λ being assigned to the bin with the index m. Since it is not a deterministic assignment, IB offers a stochastic rather than deterministic quantizer. Our information maximization formulation is a special case of IB, i.e., β being infinitely large, as we care predominantly about how well the label can be predicted from a compressed representation (quantized logits), in other words, making the compression rate as small as possible is not a request from the problem. For us, the only bottleneck is the number of bins usable for quantization. Furthermore, with $\beta \to \infty$, stochastic quantization degenerating to a deterministic one. If using stochastic binning for calibration, it outputs a weighted sum of all bin representatives, thereby being continuous and not ECE verifiable. Given that, we do not use it for calibration.

As the IB defines the best trade-off between the information rate and compression rate, we use it as the upper limit for assessing the optimality of I-Max in Fig. 7.4-b). By varying β, IB depicts the maximal achievable information rate for the given compression rate. For binning schemes (Eq. size, Eq. mass and I-Max), we vary the number of bins, and evaluate their achieved information and compression rates. As we can clearly observe from Fig. 7.4-b), I-Max can approach the upper limit defined by IB. Note that, the compression rate, though being measured in bits, is different to the number of bins used for the quantizer. As quantization is lossy, the compression rate defines the common information between the logits and quantized logits. The number of bins used for quantization imposes an upper limit on the information that can be preserved after quantization.

As the iterative method (Eq. 7.4) operates under an approximation of the inaccessible ground truth distribution $p(y,\lambda)$, we synthesize an example, see Fig. 7.4, to assess its effectiveness. As quantization can only reduce the MI, we evaluate $I(y; \lambda)$, serving as the upper bound in Fig. 7.4-a) for $I(y; Q(\lambda))$. Among the three realizations of Q, I-Max achieves higher MI than Eq. size and Eq. mass, and more importantly, it approaches the upper bound over the iterations. Next, we assess the performance within the framework of information bottleneck (IB) [Tishby et al., 1999], see Fig. 7.4-b). In the context of our problem, IB tackles $\min 1/\beta \times I(\lambda; Q(\lambda)) - I(y; Q(\lambda))$ with the weight factor $\beta > 0$ to balance between 1) maximizing the information rate $I(y; Q(\lambda))$, and 2) minimizing the compression rate $I(\lambda; Q(\lambda))$. By varying β, IB gives the maximal achievable information rate for the given compression rate. Fig. 7.4-b) shows that I-Max approaches the theoretical limits and provides an information-theoretic perspective on the sub-optimal performance of the alternative binning schemes. Sec. 7.4 has a more detailed discussion on the connection of IB and our problem formulation.

7.5. I-Max initialization

We propose to initialize the iterative method by modifying the k-means++ algorithm [Arthur & Vassilvitskii, 2007] that was developed to initialize the cluster centers for k-means clustering



(a) Convergence behavior

(b) Label-information vs. compression rate

Figure 7.4.: MI evaluation: The KDEs of $p(\lambda|y)$ for $y \in \{0,1\}$ shown in Fig. 7.3 are used as the ground truth distribution S_{kde} and evaluate the MI of Eq. mass, Eq. size, and I-Max binning trained over S_{kde}. (a) The developed iterative solution for I-Max bin optimization over S_{kde} successfully increases the MI over iterations, approaching the theoretical upper bound $I(y;\lambda)$. For comparison, I-Max is initialized with both Eq. size and Eq. mass bin edges, both of which are suboptimal at label information preservation. (b) We compare the three binning schemes with 2 to 16 quantization levels against the IB limit [Tishby et al., 1999] on the label-information $I(y;Q(\lambda))$ vs. the compression rate $I(\lambda;Q(\lambda))$. The information-rate pairs achieved by I-Max binning are very close to the limit. The information loss of Eq. mass binning is considerably larger, whereas Eq. size binning gets stuck in the low rate regime, failing to reach the upper bound even with more bins.

algorithms. These clusters allow the I-Max algorithm to initliaze the bin edges according to the clusters, therefore we require converting the binning problem into a clustering problem for initialization. It is based on the following identification

$$\mathcal{L}(\{g_m,\phi_m\}) + I(y;\lambda) = \sum_{i=0}^{M-1} \int_{g_m}^{g_{m+1}} p(\lambda)\mathrm{KDL}\left[P(y=y'|\lambda)\|P_\sigma(y=y';\phi_m)\right]\mathrm{d}\lambda \tag{7.11}$$

$$\geq \int_{-\infty}^{\infty} p(\lambda)\min_m \mathrm{KLD}\left[P(y=y'|\lambda)\|P_\sigma(y=y';\phi_m)\right]\mathrm{d}\lambda$$

$$\approx \frac{1}{|S|}\sum_{\lambda_n\in S}\min_m \mathrm{KLD}\left[P(y=y'|\lambda_n)\|P_\sigma(y=y';\phi_m)\right]. \tag{7.12}$$

As $I(y;\lambda)^3$ is a constant with respect to $(\{g_m,\phi_m\})$, minimizing $\mathcal{L}$ is equivalent to minimizing the term on the RHS of (7.11). The last approximation is reached by turning the binning problem into a clustering problem, i.e., grouping the logit samples in the training set S according to the Kullback-Leibler divergence (KLD)[4] measure, where $\{\phi_m\}$ are effectively the centers of each cluster. k-means++ algorithm [Arthur & Vassilvitskii, 2007] initializes the cluster centers based on the Euclidean distance. In our case, we alternatively use the Jensen-Shannon divergence (JSD)[5] as the distance measure to initialize $\{\phi_m\}$. Comparing with KLD, JSD is symmetric and bounded.

[3] Note that this is the mutual information between the label y and and unquantized / uncalibrated logits λ. This is different to $I(y;Q(\lambda))$

[4] For discrete probability distributions P and Q, $\mathrm{KLD}(P\|Q) = \sum_{\mathbf{x}} P(\mathbf{x})\log\left(\frac{P(\mathbf{x})}{Q(\mathbf{x})}\right)$

[5] For discrete probability distributions P and Q, $\mathrm{JSD}(P\|Q) = 0.5 \times \mathrm{KLD}(P\|M) + 0.5 \times \mathrm{KLD}(Q\|M)$ where $M = 0.5 \times (P+Q)$.

7.6. Pseudo-code, complexity and memory analysis

To ease the reproducibility of I-Max, we provide the pseudocode in Algorithm. 1. Based on it, we further analyze the complexity and memory cost of I-Max at training and test time.

We simplify this complexity analysis as our algorithm runs completely offline and is purely numpy-based. We note that despite the underlying (numpy) operations performed at each step of the algorithm differ, we treat multiplication, division, logarithm and exponential functions each counting as the same unit cost and ignore the costs of the logic operations and add/subtract operators. The initialization has complexity of $\mathcal{O}(NM)$, for the one-dimensional logits. We exploit the sklearn implementation of Kmeans++ initialization initially used for Kmeans clustering, but replace the MSE with JSD in the distance measure. Following Algorithm 1, we arrive at the following complexity of $\mathcal{O}(N * M + I * (10 * M + 2 * M))$, where I is the number of iterations, N is the number of samples and M is the number of bins. Our python codes runs Algorithm. 1 within seconds for classifiers as large as ImageNet and performed purely in Numpy. The largest storage and memory consumption is for keeping the N logits used during the I-Max learning phase.

At test time, there is negligible memory and storage constraints, as only $(2M - 1)$ floats need to be saved for the M bin representatives $\{\phi_m\}_0^{M-1}$ and $M - 1$ bin edges $\{g_m\}_1^{M-1}$. The complexity at test time is merely logic operations to compute the bin assignments of each logit and can be done using numpy's efficient 'quantize' function. I-Max offers a real-time post-hoc calibrator which adds an almost-zero complexity and memory cost relative to the computations of the original classifier.

We will release our code soon.

7.7. Experimental setup

7.7.1. Setup and details

Datasets and models We evaluate post-hoc calibration methods on four benchmark datasets, i.e., ImageNet [Deng et al., 2009], CIFAR 10/100 [Krizhevsky & Hinton, 2009] and SVHN [Netzer et al., 2011], and across various modern DNNs architectures. More details are reported in Sec. A4.1.

7.7.2. Training and evaluation details

We perform class-balanced random splits of the data test set, unless stated otherwise: the calibration and evaluation set sizes are both 25k for ImageNet, and 5k for CIFAR10/100. Different to ImageNet and CIFAR10/100, the test set of SVHN is class imbalanced. We evenly split it into the calibration and evaluation set of size 13k. All reported numbers are the means across 5 random splits; stds can be found in the appendix. Note that some calibration methods only use a subset of the available calibration samples for training, showing their sample efficiency. Further calibrator training details are provided in Sec. A4.1.

We empirically evaluate MI, Accuracy (top-1 and 5 ACCs), ECE (class-wise and top-1), Brier and NLL; the latter are shown in the appendix. Analogous to Nixon et al. [2019], we use thresholding

Algorithm 1: I-Max Binning Calibration

```
Input: Number of bins $M$, logits $\{\lambda_n\}_1^N$ and binary labels $\{y_n\}_1^N$
Result: bin edges $\{g_m\}_0^M$ ($g_0 = -\infty$ and $g_M = \infty$) and bin representations $\{\phi_m\}_0^{M-1}$
Initialization: $\{\phi_m\} \leftarrow$ Kmeans++($\{\lambda_n\}_1^N$, $M$) (see Sec. 7.5) ;
for $iteration = 1,2,\dots,I$ do
    for $m = 1,2,\dots,M-1$ do
        $g_m \leftarrow \log\left\{\frac{\log\left[\frac{1+e^{\phi_m}}{1+e^{\phi_{m-1}}}\right]}{\log\left[\frac{1+e^{-\phi_{m-1}}}{1+e^{-\phi_m}}\right]}\right\}$;
    end
    for $m = 0,1,\dots,M-1$ do
        $\mathcal{S}_m \triangleq \{\lambda_n\} \cap [g_m, g_{m+1})$ ;
        $\phi_m \leftarrow \log\left\{\frac{\sum_{\lambda_n \in \mathcal{S}_m} \sigma(\lambda_n)}{\sum_{\lambda_n \in \mathcal{S}_m} \sigma(-\lambda_n)}\right\}$;
    end
end
```

when evaluating the class-wise ECE (${}_{\mathrm{CW}}\mathrm{ECE}_{\mathrm{thr}}$). Without thresholding, the empirical class-wise ECE score may be misleading. When a class-k has a small class prior (e.g. 0.01 or 0.001), the empirical class-wise ECE score will be dominated by prediction samples where the class-k is not the ground truth. For these cases, a properly trained classifier will often not rank this class-k among the top classes and instead yield only small calibration errors. While it is good to have many cases with small calibration errors, they should not wash out the calibration errors of the rest of the cases (prone to poor calibration) through performance averaging. These include (1) class-k is the ground truth class and not correctly ranked and (2) the classifier mis-classifies some class-j as class-k. The thresholding remedies the washing out by focusing more on crucial cases (i.e. only averaging across cases where the prediction of the class-k is above a threshold). In the following experiments, our primary choice of threshold is to set it according to the class prior for the reason that the class-k is unlikely to be the ground truth if its a-posteriori probability becomes lower than its prior after observing the sample.

While empirical ECE estimation of binning schemes is simple, we resort to HDE with 100 equal size evaluation bins [Wenger et al., 2020] for scaling methods. Sec. 7.14 also reports the results attained by HDE with additional binning schemes and KDE. For HDE-based ones, we notice that with 100 evaluation bins, the ECE estimate is insensitive to the choice of binning scheme.

7.7.3. No extra normalization after K class-wise calibrations

There is a group of calibration schemes that rely on one-vs-rest conversion to turn multi-class calibration into K class-wise calibrations, e.g., histogram binning (HB), Platt scaling and Isotonic regression. After per-class calibration, the calibrated prediction probabilities of all classes no longer fulfill the constraint, i.e., $\sum_{k=1}^{K} q_k \neq 1$. An extra normalization step was taken in [Guo et al., 2017] to regain the normalization constraint. Here, we note that this extra normalization is unnecessary and partially undoes the per-class calibration effect. For HB, normalization will make its outputs continuous like any other scaling methods, thereby suffering from the same issue at ECE evaluation. One-vs-rest strategy essentially marginalizes the multi-class predictive distribution over each class. After such marginalization, each class and its prediction probability shall be treated independently, i.e. calibration of each class (class-wise calibration), thus no longer being constrained by the multi-class normalization constraint. This is analogous to train a CIFAR or ImageNet classifier with $\mathrm{sigmoid}$ rather than $\mathrm{softmax}$ cross entropy loss, e.g., [Ryou et al., 2019]. At training and test time, each class prediction probability is individually taken from the respective $\mathrm{sigmoid}$-response without normalization. The class with the largest response is then top-ranked, and normalization itself has no influence on the ranking performance.

7.7.4. Tasks

We compare the performance of multiple post-hoc calibration methods on the following tasks:

1. **Binning-based calibration performance** (Sec. 7.8): Comparison of binning calibration methods.
2. **Importance of bin edges for accuracy preservation** (Sec. 7.9): An analysis explaining the importance of correctly choosing the bin edges for an accuracy preserving binning calibrator.
3. **Scaling vs. I-Max binning** (Sec. 7.10): Comparison between post-hoc scaling and binning calibration methods.
4. **sCW strategy for scaling methods** (Sec. 7.11): Using the shared class-wise strategy for OvR-based scaling methods.
5. **Imbalanced multi-class setting** (Sec. 7.12): Calibration performance for the class-imbalanced dataset setting.
6. **Ablation on the number of bins and calibration set size** (Sec. 7.13): Analyzing changes to the I-Max hyper-parameters: number of bins and calibration set size.
7. **Empirical ECE estimation of scaling methods under multiple evaluation schemes** (Sec. 7.14): Using multiple ECE approximation schemes to evaluate the calibration performance of continuou output scaling methods.

7.8. Binning-based calibration

In Tab. 7.1, we compare three binning schemes: Eq. size, Eq. mass and I-Max binning. The classification accuracy performances of the binning schemes are proportional to their MI; Eq. mass binning is highly sub-optimal at label information preservation, and thus shows a severe accuracy drop. Eq. size binning accuracy is more similar to that of I-Max binning, but still lower, in particular at Acc_{top5}. Also note that I-Max approaches the MI theoretical limit of $I(y;\lambda)$=0.0068. Advantages of I-Max become even more prominent when comparing the NLLs of the binning schemes. For all ECE evalution metrics, I-Max binning improves on the baseline calibration performance, and outperforms Eq. size binning. Eq. mass binning is out of this comparison scope due to its poor accuracy deeming the method impractical. Overall, I-Max successfully mitigates the negative impact of quantization on ACCs while still providing an improved and verifiable ECE performance. Additionally, one-for-all sCW I-Max achieves an even better calibration with only 1k calibration samples, instead of the standard CW binning with 25k calibration samples, highlighting the effectiveness of the sCW strategy.

Furthermore, it is interesting to note that ${}_{\mathrm{CW}}$ECE of the Baseline classifier is very small, i.e., 0.000442, thus it may appear as the Baseline classifier is well calibrated. However, ${}_{\mathrm{top1}}$ECE is much larger, i.e., 0.0357. Such inconsistent observations disappear after thresholding the class-wise ECE with the class prior. This example confirms the necessity of thresholding the class-wise ECE.

In Sec. 7.13 we perform additional ablations on the number of bins and calibration samples. Accordingly, a post-hoc analysis investigates how the quantization error of the binning schemes change the ranking order. Observations are consistent with the intuition behind the problem formulation (see Sec. 7.2.2) and empirical results from Tab. 7.1 that MI maximization is a proper criterion for multi-class calibration and it maximally mitigates the potential accuracy loss.

Table 7.1.: ACCs and ECEs of Eq. mass, Eq. size and I-Max binning for the case of ImageNet (InceptionResNetV2). Due to the poor accuracy of Eq. mass binning, its ECEs are not considered for comparison (e.g. not bold even though it produces the lowest CW-ECE value). The MI is empirically evaluated based on KDE analogous to Fig. 7.4, where the MI upper bound is $I(y;\lambda)$=0.0068. For the other datasets and models, we refer to A5.

Binn.	sCW(?)	size	MI ↑	$\mathrm{Acc}_{\mathrm{top1}}$ ↑	$\mathrm{Acc}_{\mathrm{top5}}$ ↑	${}_{\mathrm{CW}}\mathrm{ECE}$ ↓	${}_{\mathrm{CW}}\mathrm{ECE}_{\mathrm{cls-prior}}$ ↓	${}_{\mathrm{top1}}\mathrm{ECE}$ ↓	NLL ↓
Baseline	✗	-	-	**80.33**	**95.10**	0.000442	0.0486	0.0357	0.8406
Eq. Mass	✗	25k	0.0026	7.78	27.92	0.000173	0.0016	0.0606	3.5960
	✓	1k	0.0026	5.02	26.75	0.000165	0.0022	0.0353	3.5272
Eq. Size	✗	25k	0.0053	78.52	89.06	0.000310	0.1344	0.0547	1.5159
	✓	1k	0.0062	80.14	88.99	0.000298	0.1525	0.0279	1.2671
I-Max	✗	25k	**0.0066**	80.27	95.01	0.000346	0.0342	0.0329	0.8499
	✓	1k	**0.0066**	80.20	94.86	**0.000296**	**0.0302**	**0.0200**	**0.7860**

7.9. Importance of bin edges for accuracy preservation

In Tab. 7.1 of Sec. 7.8, we compared three different binning schemes by measuring their ACCs and ECEs. The observation on their accuracy performance is aligned with our mutual information maximization viewpoint introduced in Sec. 7.2.2 and Fig. 7.3. Here, we re-present Fig. 7.3 and provide an alternative explanation to strengthen our understanding on how the location of bin edges affects the accuracy, e.g., why Eq. Size binning performed acceptable at the top-1 ACC, but failed at the top-5 ACC. Specifically, Fig. 7.5 shows the histograms of raw logits that are grouped based on their ranks instead of their labels as in Fig. 7.3. As expected, the logits with low ranks (i.e., rest below top-5 in Fig. 7.5) are small and thus take the left hand side of the plot, whereas the top-1 logits are mostly located on the right hand side. Besides sorting logits according to their ranks, we additionally estimate the density of the ground truth (GT) classes associated logits, i.e., GT in Fig. 7.5. With a properly trained classifier, the histogram of top-1 logits shall largerly overlap with the density curve GT, i.e., top-1 prediction being correct in most cases.

From the bin edge location of Eq. Mass binning, it attempts to attain small quantization errors for logits of low ranks rather than top-5. This will certainly degrade the accuracy performance after binning. On contrary, Eq. Size binning aims at small quantization error for the top-1 logits, but ignores top-5 ones. As a result, we observed its poor top-5 ACCs. I-Max binning nicely distributes its bin edges in the area where the GT logits are likely to locate, and the bin width

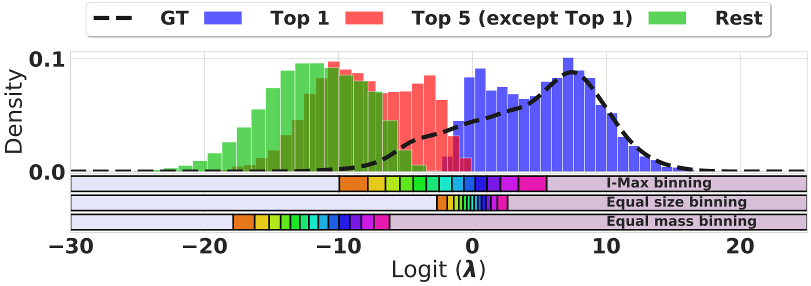


Figure 7.5.: Histogram of CIFAR100 (WRN) logits in S constructed from 1k calibration samples, using the same setting as Fig. 7.3. Instead of categorizing the logits according to their two-class label $y_k \in \{0,1\}$ as in Fig. 7.3, here we sort them according to their ranks given by the CIFAR100 WRN classifier. As a baseline, we also plot the KDE of logits associated to the ground truth classes, i.e., GT.

becomes smaller in the area where the top-5 logits are close by (i.e., the overlap region between the red and blue histograms). Note that, any logit larger than zero must be top-1 ranked, as there can exist at most one class with prediction probability larger than 0.5. Given that, the bins located above zero are no longer to maintain the ranking order, rather to reduce the precision loss of top-1 prediction probability after binning.

Table 7.2.: Comparison of sCW binning methods in the case of ImageNet - InceptionResNetV2. As sCW binning creates ties at top predictions, the ACCs initially reported in Tab. 1 of Sec. 4.1 use the class index as the secondary sorting criterion. Here, we add Acc*$_{\text{top1}}$ and Acc*$_{\text{top5}}$ which are attained by using the raw logits as the secondary sorting criterion. As the CW ECEs are not affected by this change, here we only report the new $_{\text{top1}}$ECE*.

Binn.	Acc$_{\text{top1}}$ ↑	Acc*$_{\text{top1}}$ ↑	Acc$_{\text{top5}}$ ↑	Acc*$_{\text{top5}}$ ↑	$_{\text{top1}}$ECE ↓	$_{\text{top1}}$ECE* ↓	NLL ↓
Baseline	**80.33**	-	**95.10**	-	0.0357	-	0.8406
Eq. Mass	5.02	**80.33**	26.75	**95.10**	0.0353	0.7884	3.5272
Eq. Size	80.14	80.21	88.99	**95.10**	0.0279	0.0277	1.2671
I-Max	80.20	**80.33**	94.86	**95.10**	**0.0200**	**0.0202**	**0.7860**

The second part of our post-hoc analysis is on the sCW binning strategy. When using the same binning scheme for all per-class calibration, the chance of creating ties in top-k predictions is much higher than CW binning, e.g., more than one class are top-1 ranked according to the calibrated prediction probabilities. Our reported ACCs in Tab. 7.1 are attained by simply returning the first found class, i.e., using the class index as the secondary sorting criterion. This is certainly a suboptimal solution. Here, we investigate on how the ties affect ACCs of sCW binning. To this end, we use raw logits (before binning) as the secondary sorting criterion. The resulting $\text{ACC}^*_{\text{top1}}$ and $\text{ACC}^*_{\text{top5}}$ are shown in Tab.7.2. Interestingly, such a simple change reduces the accuracy loss of Eq. Mass and I-Max binning to zero, indicating that they can preserve the top-5 ranking order of the raw logits but not in a strict monotonic sense, i.e., some $>$ are replaced by $=$. As opposed to I-Max binning, Eq. Mass binning has a poor performance at calibration, i.e., the really high NLL and ECE. This is because it trivially ranks many classes as top-1, but each of them has a very and same small confidence score. Given that, even though the accuracy loss is no longer an issue, it is still not a good solution for multi-class calibration. For Eq. Size binning, resolving ties only helps restore the baseline top-5 but not top-1 ACC. Its poor bin representative setting due to unreliable empirical frequency estimation over too narrow bins can result in a permutation among the top-5 predictions.

Concluding from the above, our post-hoc analysis confirms that I-Max binning outperforms the other two binning schemes at mitigating the accuracy loss and multi-class calibration. In particular, there exists a simple solution to close the accuracy gap to the baseline, at the same time still retaining the desirable calibration gains.

7.10. Scaling vs. I-Max binning

In Tab. 7.3, we compare I-Max binning to benchmark scaling methods. Namely, matrix scaling with L_2 regularization [Kull et al., 2019] has a large model capacity compared to other parametric scaling methods, while TS [Guo et al., 2017] only uses a single parameter and MnM [Zhang et al., 2020] uses three temperatures as an ensemble of TS (ETS). As a non-parametric method, GP [Wenger et al., 2020] yields state of the art calibration performance. Additional 6 scaling methods can be found in Sec. A6. Benefiting from its model capacity, matrix scaling achieves the best accuracy. I-Max binning achieves the best calibration on CIFAR-100; on ImageNet, it has the best ${}_{\mathrm{CW}}\mathrm{ECE}$, and is similar to GP on ${}_{\mathrm{top1}}\mathrm{ECE}$. For a broader scope of comparison, we refer to Sec. A6.

To showcase the complementary nature of scaling and binning, we investigate combining binning with GP (a top performing non-parametric scaling method, though with the drawback of high complexity) and TS (a commonly used scaling method). Here, we propose to bin the raw logits and use the GP/TS scaled logits of the samples per bin for setting the bin representatives, replacing the empirical frequency estimates. As GP is then only needed at the calibration learning phase, complexity is no longer an issue. Being mutually beneficial, GP helps improving ACCs and ECEs of binning, i.e., marginal ACC drop 0.16% (0.01%) on $\mathrm{Acc}_{\mathrm{top1}}$ for ImageNet (CIFAR100) and 0.24% on $\mathrm{Acc}_{\mathrm{top5}}$ for ImageNet; and large ECE reduction 38.27% (49.78%) in ${}_{\mathrm{CW}}\mathrm{ECE}_{\mathrm{cls-prior}}$ and 66.11% (76.07%) in ${}_{\mathrm{top1}}\mathrm{ECE}$ of the baseline for ImageNet (CIFAR100).

Table 7.3.: ACCs and ECEs of I-Max binning (15 bins) and scaling methods. All methods use 1k calibration samples, except for Mtx. Scal. and ETS-MnM, which requires the complete calibration set, i.e., 25k/5k for ImageNet/CIFAR100. Additional 6 scaling methods can be found in Sec. A6.

	CIFAR100 (WRN)			ImageNet (InceptionResNetV2)			
Calibrator	$\mathrm{Acc}_{\mathrm{top1}}$ ↑	${}_{\mathrm{CW}}\mathrm{ECE}_{\mathrm{cls-prior}}$ ↓	${}_{\mathrm{top1}}\mathrm{ECE}$ ↓	$\mathrm{Acc}_{\mathrm{top1}}$ ↑	$\mathrm{Acc}_{\mathrm{top5}}$ ↑	${}_{\mathrm{CW}}\mathrm{ECE}_{\mathrm{cls-prior}}$ ↓	${}_{\mathrm{top1}}\mathrm{ECE}$ ↓
Baseline	81.35	0.1113	0.0748	80.33	95.10	0.0486	0.0357
Mtx Scal. w. L_2	**81.44**	0.1085	0.0692	**80.78**	**95.38**	0.0508	0.0282
TS	81.35	0.0911	0.0511	80.33	95.10	0.0559	0.0439
GP	81.34	0.1074	0.0358	80.33	95.11	0.0485	*0.0186*
ETS-MnM	81.35	0.0976	0.0451	80.33	95.10	0.0479	0.0358
I-Max	81.30	*0.0518*	*0.0231*	80.20	94.86	*0.0302*	0.0200
I-Max w. TS	81.34	**0.0510**	0.0365	80.20	94.87	0.0354	0.0402
I-Max w. GP	81.34	0.0559	**0.0179**	80.20	94.87	**0.0300**	**0.0121**

7.11. sCW strategy for scaling methods

Though without quantization loss, some scaling methods, i.e., Beta [Kull et al., 2017], Isotonic regression [Zadrozny & Elkan, 2002], and Platt scaling [Platt, 1999], even suffer from more severe accuracy degradation than I-Max binning. As they also use the one-vs-rest strategy for multi-class calibration, we find that the proposed shared CW binning strategy is beneficial for reducing their accuracy loss and improving their ECE performance, with only 1k calibration samples, see Tab. 7.4.

Table 7.4.: ACCs and ECEs of scaling methods using the one-vs-rest conversion for multi-class calibration. Here we compare using 1k samples for both CW and one-for-all sCW scaling.

		CIFAR100 (WRN)			ImageNet (InceptionResNetV2)			
Calibrator	sCW(?)	Acc_{top1} ↑	$_{\text{CW}}\text{ECE}_{\text{cls-prior}}$ ↓	$_{\text{top1}}\text{ECE}$ ↓	Acc_{top1}	Acc_{top5} ↑	$_{\text{CW}}\text{ECE}_{\text{cls-prior}}$ ↓	$_{\text{top1}}\text{ECE}$ ↓
Baseline	-	81.35	0.1113	0.0748	80.33	95.10	0.0489	0.0357
Beta	✗	81.02	0.1066	0.0638	77.80	86.83	0.1662	0.1586
Beta	✓	**81.35**	0.0942	0.0357	**80.33**	**95.10**	0.0625	0.0603
I-Max w. Beta	✓	81.34	**0.0508**	**0.0161**	80.20	94.87	**0.0381**	**0.0574**
Isot. Reg. (IR)	✗	80.62	0.0989	0.0785	77.82	88.36	0.1640	0.1255
Isot. Reg. (IR)	✓	81.30	0.0602	0.0257	**80.22**	**95.05**	0.0345	0.0209
I-Max w. IR	✓	**81.34**	**0.0515**	**0.0212**	80.20	94.87	**0.0299**	**0.0170**
Platt Scal.	✗	81.31	0.0923	0.1035	**80.36**	94.91	0.0451	0.0961
Platt Scal.	✓	**81.35**	0.0816	0.0462	80.33	**95.10**	0.0565	0.0415
I-Max w. Platt	✓	81.34	**0.0511**	**0.0323**	80.20	94.87	**0.0293**	**0.0392**

7.12. Imbalanced multi-class setting

Lastly, we turn our experiments to an imbalanced multi-class setting. The adopted SVHN dataset has non-uniform class priors, ranging from 6% (e.g. digit 8) to 19% (e.g. digit 0). We reproduce Tab. 7.3 for SVHN, yielding Tab. 7.5. In order to better control the bias caused by the calibration set merging in the imbalanced multi-class setting, the former one-for-all sCW strategy in the balanced multi-class setting changes to sharing I-Max among classes with similar class priors. Despite the class imbalance, I-Max and its variants perform best compared to the other calibrators, being similar to Tab. 7.3. This shows that I-Max and the sCW strategy both can generalize to imbalanced multi-class setting.

In Tab. 7.5, we additionally evaluate the class-wise ECE at multiple threholds. We ablate various thresholds settings, namely, 1) 0 (no thresholding); 2) the class prior; 3) $1/K$ (any class with prediction probability below $1/K$ will not be the top-1); and 4) a relatively large number 0.5 (the case when the confidence on class-k outweighs NOT class-k). We observe that I-Max and its variants are consistently top performing across the different thresholds.

7.13. Ablation on the number of bins and calibration set size

In Tab. 7.1 of Sec. 7.8, sCW I-Max binning is the top performing one at the ACCs, ECEs and NLL measures. In this part, we further investigate on how the number of bins and calibration set size influences its performance. Tab. 7.6 shows that in order to benefit from more bins we shall accordingly increase the number of calibration samples. More bins help reduce the quantization loss, but increase the empirical frequency estimation error for setting the bin representatives. Given that, we observe a reduced ACCs and increased ECEs for having 50 bins with only 1k calibration samples. By increasing the calibration set size to 5k, then we start seeing the benefits

Table 7.5.: ACCs and ECEs of I-Max binning (15 bins) and scaling methods. All methods use 1k calibration samples, except for Mtx. Scal. and ETS-MnM, which requires the complete calibration set, i.e., 13k for SVHN. Here, we also report the class-wise ECEs using four different thresholds.

Calibrator	$\mathrm{Acc}_{\mathrm{top1}} \uparrow$	$_{\mathrm{top1}}\mathrm{ECE} \downarrow$	$_{\mathrm{CW}}\mathrm{ECE}_{0} \downarrow$	$_{\mathrm{CW}}\mathrm{ECE}_{\frac{1}{K}} \downarrow$	$_{\mathrm{CW}}\mathrm{ECE}_{\text{cls-prior}} \downarrow$	$_{\mathrm{CW}}\mathrm{ECE}_{\frac{1}{2}} \downarrow$
Baseline	97.08	0.0201	0.0052	0.0353	0.0356	0.0260
Mtx. Scal. w. L_2	**97.09**	0.0188	0.0050	0.0346	0.0349	0.0250
ETS-MnM	97.08	0.0152	0.0054	0.0379	0.0382	0.0256
TS	97.08	0.0106	0.0041	0.0323	0.0327	0.0206
GP	97.08	0.0104	0.0043	0.0340	0.0341	0.0212
I-Max	96.88	0.0164	0.0043	0.0244	0.0245	0.0176
I-Max w. TS	97.06	0.0088	0.0025	0.0156	0.0155	0.0112
I-Max w. GP	97.06	**0.0074**	**0.0024**	**0.0148**	**0.0147**	**0.0110**

of having more bins to reduce quantization error for better ACCs. Next, we further exploit scaling method, i.e., GP [Wenger et al., 2020], for improving the sample efficiency of binning at setting the bin representatives. As a result, the combination is particularly beneficial to improve the ACCs and top-1 ECE. Overall, more bins are beneficial to ACCs, while ECEs favor less number of bins.

Table 7.6.: Ablation on the number of bins and calibration samples for sCW I-Max binning, where the basic setting is identical to Tab. 7.1.

Binn.	Bins	$\mathrm{Acc}_{\mathrm{top1}} \uparrow$	$\mathrm{Acc}_{\mathrm{top5}} \uparrow$	$_{\mathrm{CW}}\mathrm{ECE}_{\frac{1}{K}} \downarrow$	$_{\mathrm{top1}}\mathrm{ECE} \downarrow$	$\mathrm{Acc}_{\mathrm{top1}} \uparrow$	$\mathrm{Acc}_{\mathrm{top5}} \uparrow$	$_{\mathrm{CW}}\mathrm{ECE}_{\frac{1}{K}} \downarrow$	$_{\mathrm{top1}}\mathrm{ECE} \downarrow$
Baseline	-	**80.33**	95.10	0.0486	0.0357	**80.33**	95.10	0.0486	0.0357
		1k Calibration Samples				5k Calibration Samples			
GP		**80.33**	**95.11**	0.0485	0.0186	**80.33**	**95.11**	0.0445	0.0177
I-Max	10	80.09	94.59	0.0316	0.0156	80.14	94.59	0.0330	0.0107
	15	80.20	94.86	0.0302	0.0200	80.21	94.90	0.0257	0.0107
	20	80.10	94.94	**0.0266**	0.0234	80.25	94.98	**0.0220**	0.0133
	30	80.15	94.99	0.0343	0.0266	80.25	95.02	0.0310	0.0150
	40	80.11	95.05	0.0365	0.0289	80.24	95.08	0.0374	0.0171
	50	80.21	94.95	0.0411	0.0320	80.23	95.06	0.0378	0.0219
I-Max w. GP	10	80.09	94.59	0.0396	0.0122	80.14	94.59	0.0330	**0.0072**
	15	80.20	94.87	0.0300	**0.0121**	80.21	94.88	0.0256	0.0080
	20	80.23	94.95	0.0370	0.0133	80.25	95.00	0.0270	0.0091
	30	80.26	95.04	0.0383	0.0141	80.27	95.02	0.0389	0.0097
	40	80.27	**95.11**	0.0424	0.0145	80.26	95.08	0.0402	0.0108
	50	80.30	95.08	0.0427	0.0153	80.28	95.08	0.0405	0.0114

7.14. Empirical ECE estimation of scaling methods under multiple evaluation schemes

As discussed in the Chap. 6.4, scaling methods suffer from not being able to provide verifiable ECEs, see Fig. 7.1. Here, we discuss alternatives to estimate their ECEs. The current literature can be split into two types of ECE evaluation: histogram density estimation (HDE) and kernel density estimation (KDE).

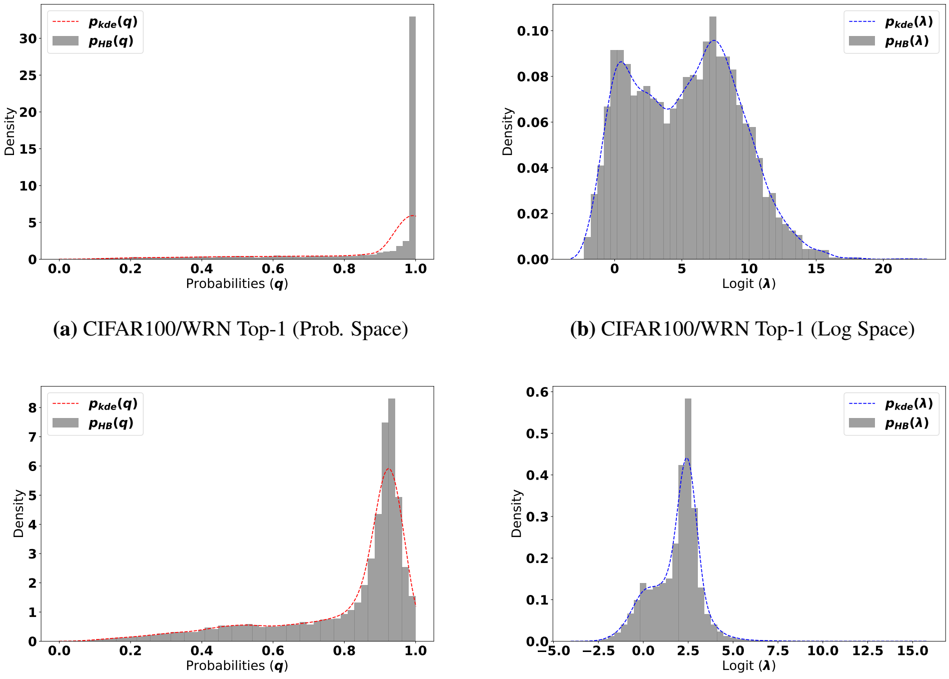
7.14.1. HDE-based ECE evaluation

HDE bins the prediction probabilities (logits) for density modeling. The binning scheme has different variants, where changing the bin edges can give varying measures of the ECE. Two bin edges schemes have been discussed in the literature (Eq. size and Eq. mass) as well as a new scheme was introduced (I-Max). Alternatively, we also evaluate a binning scheme which is based on KMeans clustering to determine the bin edges.

7.14.2. KDE-based ECE evaluation

Recent work [Zhang et al., 2020] presented an alternative ECE evaluation scheme which exploits KDEs to estimate the distribution of prediction probabilities $\{q_k\}$ from the test set samples. Using the code provided by [Zhang et al., 2020], we observe that the KDE with the setting in their paper can have a sub-optimal fitting in the probability space. This can be observed from Fig. 7.6a and Fig. 7.6c, where the fitting is good for ImageNet/Inceptionresnetv2 though when the distribution is significantly skewed to the right (as in the case of CIFAR100/WRN) the mismatch becomes large. We expect that the case of CIFAR100/WRN is much more common in modern DNNs, due to their high capacities and prone to overfitting.

Equivalently, we can learn the distribution in its log space by the bijective transformation, i.e., $\lambda = \log q - \log(1-q)$ and $q = \sigma(\lambda)$. As we can observe from Fig. 7.6b and Fig. 7.6d, the KDE fitting for both models is consistently good.



(a) CIFAR100/WRN Top-1 (Prob. Space)

(b) CIFAR100/WRN Top-1 (Log Space)

(c) ImageNet/Inceptionresnetv2 Top-1 (Prob. Space)

(d) ImageNet/Inceptionresnetv2 Top-1 (Log Space)

Figure 7.6.: Distribution of the top-1 predictions and its log-space counterparts, i.e., $\lambda = \log q - \log(1-q)$.

Zhang et al. [2020] empirically validated their KDE in a toy example, where the ground truth ECE can be analytically computed. By analogy, we reproduce the experiment and further compare it with the log-space KDE evaluation. Using the same settings as in Zhang et al. [2020], we assess the ECE evaluation error by KDE, i.e., $|\mathrm{ECE}_{\mathrm{gt}} - \mathrm{ECE}_{\mathrm{kde}}|$, in both the log and probability space, achieving $\mathrm{prob}\ 0.0020$ vs. $\log 0.0017$ for the toy example setting $\beta_0 = 0.5; \beta_1 = -1.5$ (see Zhang et al. [2020] for details of their experiment). For an even less calibrated setting, $\beta_0 = 0.2; \beta_1 = -1.9$, we obtain $\mathrm{prob}\ 0.0029$ vs. $\log 0.0020$. So the log-space KDE-based ECE evaluation ($\mathrm{kdeECE}_{\log}$) has lower estimation error than in the probability space.

7.14.3. Alternative ECE evaluation schemes

Concluding from the above, Tab. 7.7 shows the ECE estimates attained by HDEs (from four different bin setting schemes) and KDE (from Zhang et al. [2020], but in the log space). As we can see, the obtained results are evaluation scheme dependent. On contrary, I-Max binning with and without GP are not affected, and more importantly, their ECEs are better than that of scaling methods, regardless of the evaluation scheme.

7.15. Conclusion

We proposed I-Max binning for multi-class calibration, which maximally preserves the label-information under quantization, reducing potential accuracy losses. Using the shared class-wise (sCW) strategy, we also addressed the sample-inefficiency issue of binning and scaling methods that rely on one-vs-rest (OvR) for multi-class calibration. Our experiments showed that I-Max yields consistent class-wise and top-1 calibration improvements over multiple datasets and model architectures, outperforming HB and state-of-the-art scaling methods. Combining I-Max with scaling methods offers further calibration performance gains, and more importantly, ECE estimates that can converge to the ground truth in the large sample limit.

Future work will investigate extensions of I-Max that jointly calibrate multiple classes, and thereby directly model class correlations. Interestingly, even on datasets such as ImageNet which contain several closely related classes, there is no clear evidence that methods that do model class correlations, e.g. Mtrx. Scal. capture uncertainties better. In fact, I-Max empirically outperforms such methods, although all classes are calibrated independently under the OvR assumption. Non-OvR based methods may fail due to various reasons such as under-parameterized models (e.g. TS), limited data (e.g. Mtrx. Scal.) or complexity constraints (e.g. GP). Joint class calibration therefore strongly relies on new sample efficient evaluation measures that estimate how accurately class correlations are modeled, and which can be included as additional optimization criteria.

In the context of the literature, I-Max makes a major contribution in addressing literature gap LG3. Through a post-hoc uncertainty calibration step, it offers a competitive solution for improving the quality of the confidence estimates of *both* new classification models as well as already-train models which might already be deployed in real-world systems. A big advantage of such a solution is that it does not require re-training of the entire model, which has been notoriously exhausting for deep learning classifiers requiring extensive computational resources and data. I-Max additionally covers a large part of the literature gap by allowing calibration improvements for *any* classification model regardless of the data modality or model architecture, and requires no adaptions when the classification setting is changed. For further performance gains, in the

Table 7.7.: ECEs of scaling methods under various evaluation schemes for ImageNet InceptionResNetV2. Overall, we consider five evaluation schemes, namely (1) dECE: equal size binning; (2) mECE: equal mass binning, (3) kECE: MSE-based KMeans clustering; (4) iECE: I-Max binning; 5) kdeECE: KDE. The HDEs based schemes, i.e., (1)-(4), use 10^2 bins. Note that, the ECEs of I-Max binning (as a calibrator rather than evaluation scheme) are agnostic to the evaluation scheme. Furthermore, BBQ suffers from severe accuracy degradation.

Calibrator	$\mathrm{ACC}_{\mathrm{top}_1}$	$_{\mathrm{cw}}\mathrm{dECE}_{\frac{1}{K}}$ ↓	$_{\mathrm{cw}}\mathrm{mECE}_{\frac{1}{K}}$ ↓	$_{\mathrm{cw}}\mathrm{kECE}_{\frac{1}{K}}$ ↓	$_{\mathrm{cw}}\mathrm{iECE}_{\frac{1}{K}}$ ↓	$_{\mathrm{cw}}\mathrm{kdeECE}_{\frac{1}{K}}$ ↓	Mean ↓
Baseline	80.33	0.0486	0.0459	0.0484	0.0521	0.0749	0.0540
	25k Calibration Samples						
BBQ	53.89	0.0287	0.0376	0.0372	0.0316	0.0412	0.0353
Beta	80.47	0.0706	0.0723	0.0742	0.0755	0.0828	0.0751
Isotonic Reg.	80.08	0.0644	0.0646	0.0652	0.0655	0.0704	0.0660
Platt	80.48	0.0597	0.0593	0.0613	0.0634	0.1372	0.0762
Vec Scal. w. L2 reg.	80.53	0.0494	0.0472	0.0498	0.0531	0.0805	0.0560
Mtx Scal. w. L2 reg.	**80.78**	0.0508	0.0488	0.0512	0.0544	0.0898	0.0590
	1k Calibration Samples						
TS	80.33	0.0559	0.0548	0.0573	0.0598	0.1003	0.0656
GP	80.33	0.0485	0.0450	0.0475	0.0520	0.0580	0.0502
I-Max	80.20	0.0302					
I-Max w. GP	80.20	**0.0300**					

Calibrator	$\mathrm{ACC}_{\mathrm{top}_1}$	$_{\mathrm{top1}}\mathrm{dECE}$ ↓	$_{\mathrm{top1}}\mathrm{mECE}$ ↓	$_{\mathrm{top1}}\mathrm{kECE}$ ↓	$_{\mathrm{top1}}\mathrm{iECE}$ ↓	$_{\mathrm{top1}}\mathrm{kdeECE}$ ↓	Mean ↓
Baseline	80.33	0.0357	0.0345	0.0348	0.0352	0.0480	0.0376
	25k Calibration Samples						
BBQ	53.89	0.2689	0.2690	0.2690	0.2689	0.2756	0.2703
Beta	80.47	0.0346	0.0360	0.0360	0.0357	0.0292	0.0343
Isotonic Reg.	80.08	0.0468	0.0434	0.0436	0.0468	0.0437	0.0449
Platt	80.48	0.0775	0.0772	0.0771	0.0773	0.0772	0.0773
Vec Scal. w. L2 reg.	80.53	0.0300	0.0298	0.0300	0.0303	0.0365	0.0313
Mtx Scal. w. L2 reg.	**80.78**	0.0282	0.0287	0.0286	0.0289	0.0324	0.0293
	1k Calibration Samples						
TS	80.33	0.0439	0.0452	0.0454	0.0443	0.0679	0.0493
GP	80.33	0.0186	0.0182	0.0186	0.0190	0.0164	0.0182
I-Max	80.20	0.0200					
I-Max w. GP	80.20	**0.0121**					

next chapter we present a novel during-training calibration technique which optimizes both the generalization and calibration performance of deep learning classifiers.

Chapter 8.

OMADA: On-manifold adversarial data augmentation

Contribution: An alternative approach to improve the calibration performance is to adopt new training strategies which guide the deep learning classifiers training procedure. In this chapter, we present a novel augmentation strategy which directly aims at improving the training procedure of the classifiers for better quality uncertainty estimation. Particularly, to improve the uncertainty calibration, we propose On-Manifold Adversarial Data Augmentation or *OMADA*, which specifically attempts to generate challenging examples by following an on-manifold adversarial attack path in the latent space of an autoencoder that closely approximates the decision boundaries between classes. On a variety of datasets and for multiple network architectures, OMADA consistently yields more accurate and better calibrated classifiers than baseline models, and outperforms competing approaches such as Mixup, as well as achieving similar performance to (at times better than) post-processing calibration methods such as temperature scaling. Variants of OMADA can employ different sampling schemes for ambiguous on-manifold examples based on the entropy of their estimated soft labels, which exhibit specific strengths for generalization, calibration of predicted uncertainty, or detection of out-of-distribution inputs.

8.1. During-training uncertainty calibration

Current uncertainty calibration research mainly focus on post-hoc calibration (i.e. discussed in Chap. 7), though an alternative body of work center their research on improving the calibration performance during the training process [Thulasidasan et al., 2019; Mueller et al., 2019]. The latter approach, which we refer to as *during-training calibration*, aims to exploit the modeling capacity of deep learning classifiers for calibrating their predictions and addressing their over-confidence problem, in addition to learning discriminative features from the data. These include regularization techniques [Hein et al., 2019], data augmentation [Thulasidasan et al., 2019], modification of the training labels [Mueller et al., 2019] or loss [Mukhoti et al., 2020]. In this chapter, we focus of utilizing the power of data augmentation to improve the calibration performance of deep learning classifiers during the training process by learning from a manifold-based augmentation strategy. In contrast to post-hoc calibration techniques which are specifically tasked to yield better calibration errors, we showcase the benefits of an effective during-training calibration strategy for additional performance gains in terms of generalization (e.g. Accuracy and negative-log likelihood (NLL)) and other uncertainty tasks (e.g. outlier detection and sparsification). Given that post-hoc and during-training calibration can be viewed as orthogonal techniques, we demonstrate their effective combination for further calibration performance gains.

Over-confidence and hard labels

The softmax outputs of modern DNNs, although accurate in their class predictions, have proven to perform poorly as indicators of uncertainty. Overfitting to the training data with one-hot encoded or *hard* labels [Thulasidasan et al., 2019], and over-confidence of ReLU networks for out-of-data inputs [Hein et al., 2019] have been identified as potential root causes for this behavior. Estimating the predictive uncertainty in deep learning is thus an active and challenging research topic. The ultimate goal is to obtain *calibrated* uncertainties [Guo et al., 2017], i.e. indicators that directly quantify the likelihood of a correct prediction.

Since in all practical machine learning scenarios there is no access to the true data generating distribution, a reasonable starting point for uncertainty estimation is to assume that only data points in the vicinity of training data points can be predicted with high certainty [Zhang et al., 2018]. In fact, this is closely related to the problem of generalization, and various data augmentation techniques improve classification accuracy on unseen data by generating new training samples obtained by applying simple transformations to the original training samples without modifying the labels. In this article we propose a novel approach *On-manifold Adversarial Data Augmentation*, or *OMADA*, which yields calibrated uncertainty predictions by augmenting the training dataset with ambiguous samples generated by adversarial attacks [Goodfellow et al., 2015], but constrained to lie on an estimated training data manifold[1] [Stutz et al., 2019] (see Fig. 8.1). The adversarial attack targets a latent space classifier. Unlike typical image-space classifiers that directly process the data samples, the latent space classifier is built on top of an autoencoder (encoder-decoder) based generative model (see Fig. 8.2), and processes the latent codes of data samples created by the encoder. Restricting the adversarial attack to the latent-space ensures that the generated perturbations are semantically meaningful. OMADA can be viewed as a complementary approach to image-space attacks, which require the choice of an appropriate distance metric and ϵ-ball in image space to keep the perturbed images realistic. OMADA instead considers only neighborhoods on the manifold by utilizing the generative model.

The encoder and decoder of the generative model are jointly trained to approximate the true data distribution. By constraining the augmented samples to lie on the data manifold, we can closely approximate the true decision boundaries between classes by the latent-space classifier, while avoiding confusing the image-space classifier by injecting out-of-distribution samples into the training set.

We perform extensive experiments and comparisons against alternative methods from the literature for supervised classification. Augmenting the supervised classification training with OMADA, we observe significant improvements for calibration and accuracy across multiple benchmark datasets such as CIFAR-100, CIFAR-10 and SVHN and diverse network architectures such as DenseNet [Huang et al., 2017], Wide ResNet [Zagoruyko & Komodakis, 2016], VGG [Simonyan & Zisserman, 2015], and ResNeXt [Xie et al., 2017]. Among all compared methods for uncertainty calibration, OMADA is the only one that consistently performs well across all benchmarks and architectures. The consistent failure on specific network architectures of simpler methods suggests that the quality of the generated confusing examples is extremely

[1] A manifold is a low-dimensional embedding/representation of high-dimensional data, such as images.

important to avoid inducing undesired artefacts during training. Furthermore, we test the (image space) classifier on out-of-distribution samples. Using the confidence of the predictions as the metric to detect unseen data, OMADA outperforms multiple baseline methods in outlier detection performance in terms of the area under the ROC curve and Mean-Maximal Confidence (MMC). In summary, the results suggest that realistic but ambiguous on-manifold samples between one or more class clusters aid in resolving the notorious mis-calibration [Guo et al., 2017] and over-confidence [Hein et al., 2019] characteristics associated with DNNs.

Our contributions include (1) a novel approach OMADA to create on-manifold ambiguous samples for data augmentation in supervised classification; (2) extensive empirical comparisons of a wide spectrum of alternative methods in the literature on various uncertainty evaluation metrics, and on out-of-distribution detection tasks; (3) extensive evaluation on a number of diverse network architectures; (4) significant improvement over the benchmark methods on prediction calibration and outlier detection. For example, on CIFAR-100, OMADA results in up to a 9.8x reduction in calibration error against standard training, up to a 5.9x reduction compared to Mixup, and up to a 3x reduction compared to temperature scaling.

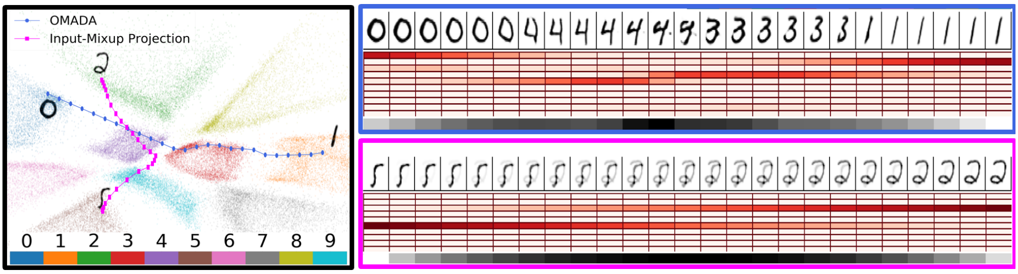


Figure 8.1.: Visualization of an MNIST encoder-decoder latent space with two trajectories traversing between pairs of clusters. On the right we visualize the decoded image path for OMADA (top) and the Input-Mixup images (bottom) along with their corresponding soft labels (10 rows below images, red intensity corresponds to likelihood for classes 0 to 9), and the class entropy (bottom row, black shows high entropy). The OMADA trajectory starts at the cluster of “0" and smoothly transitions to the target class “1". It can be seen that the path favors routes which stick to the boundary regions of class clusters (e.g. going around the red cluster of “3"s). Alternatively, we visualize the projection of Input-Mixup images onto the same manifold, for linear input interpolations between the digits “5" and “2". It can be seen that the images generated by OMADA are more confusing, and more importantly that the soft labels assigned to each image depend on the location on the manifold. This is in contrast to Mixup, where the soft label will always be non-zero for all classes except the start and end interpolation points, regardless of whether the mixed images have similar features to other classes. More OMADA trajectories (sampled more finely) can be found in Fig. B1 in the appendix.

8.2. Ambiguous sample and soft label generation

The core of OMADA is constructing realistic, yet ambiguous samples for data augmentation, and input-dependent soft labels for improving the calibration of classifiers. This section explains, in detail, how to create on-manifold ambiguous training samples, and how to exploit them for the target classification task. Fig. 8.2 sketches the three main independen training phases of OMADA: generative modeling, latent space adversarial attacks, and classifier training with on-manifold data augmentation.

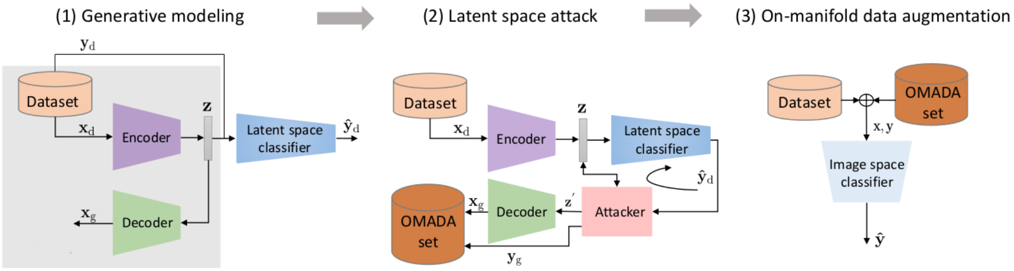


Figure 8.2.: Illustration of the three phases of OMADA: (1) generative model, used in step (2) for latent space adversarial attacks to create the OMADA set, used in step (3) to train the classifier in image space using on-manifold data augmentation.

Generative modeling for data manifold approximation

In order to model the complex high-dimensional space the data lies in, generative models are used to approximate the inaccessible ground truth data distribution. We choose BigBi-GAN [Donahue & Simonyan, 2019] because it has achieved state-of-the-art results on image synthesis and representation learning tasks, and exploit its design for learning the training data manifold.

BigBi-GAN model

The BigBi-GAN model (Fig. 8.2-(1), [Donahue & Simonyan, 2019]) consists of an encoder $\mathcal{E}_\rho(z|x)$ and decoder $\mathcal{G}_\phi(x|z)$. The encoder encodes the input sample x from the training set by a latent code z that follows the standard normal distribution P_z. The decoder attempts to reconstruct the input from the latent code z. The discriminator is trained to distinguish decoder outputs from real samples. The decoder competes against the discriminator by synthesizing increasingly realistic samples. As the discriminator is only needed for training BigBi-GAN, it is omitted in Fig. 8.2-(1).

Latent space classifier

In order to generate augmentations which are ambiguous challenging samples, we explore decision boundary regions on the manifold. Given that the current setting of the generative model is unsupervised, we further introduce a latent space classifier $\mathcal{C}_\gamma(y|z)$ that exploits the label information to cluster the latent codes $\{z\}$ of $\{x_d\}$ according to the classes $\{y_d\}$ (see Fig. 8.1). Namely, given the labeled training samples $\{x_i,y_i\}_{i=1}^N$, the classifier is trained by cross entropy minimization to predict the labels y_i from the latent code $z_i = \mathcal{E}_\rho(x_i)$ obtained by applying the encoder to x_i. The cross entropy multi-class classification loss is added to the original encoder-decoder training loss of BigBi-GAN. The three networks are jointly trained to fool the discriminator, and the discriminator is trained to detect real samples from fake ones. We further observe from Fig. 8.1 that sampling from the boundaries between two class clusters yields ambiguous samples at the decoder output. Such generated samples lie in the support of the model distribution, which well approximates the data manifold, and are considered as on-manifold samples. The trained encoder $\mathcal{E}_\rho(z|x)$, decoder $\mathcal{G}_\phi(x|z)$ and latent space classifier $\mathcal{C}_\gamma(y|z)$ provide all of the necessary tools for OMADA to generate ambiguous samples and corresponding soft labels, which is described in the following section.

8.3. Latent space adversarial attack

OMADA uses the generative model to synthesize samples which specifically have higher class ambiguity. Since ambiguous samples should reflect characteristics from two or more classes, their latent codes are expected to lie close to the class decision boundaries of the latent space classifier. As these latent codes of interest have an infinitely-small chance of being selected using conventional random sampling from the prior distribution on z, an alternative, novel sampling technique is required. We propose to use adversarial attacks on the latent space classifier to provide a targeted way to raise class ambiguity.

We start to explore the latent space from the source latent code z^s of an arbitrary training sample x^s and move in a direction to approach a target class y^o. Here, y^o is a one-hot vector encoding the class label. An adversarial attack, e.g. the projected gradient descent (PGD) method [Kurakin et al., 2017], is used to find a small perturbation z_{pert} on z^s such that the latent space classifier classifies $z^s + z_{\text{pert}}$ as y^o rather than y^s. Using the cross entropy loss, the perturbation z_{pert} is attained by solving the following minimization problem:

$$z_{\text{pert}} = \text{argmin}_{\|\delta\|_{\text{inf}}} \sum_{c=1}^{K} (-y_c^o \text{log} \mathcal{C}_\gamma (z^s + \delta)_c), \tag{8.1}$$

where K denotes the number of classes and y_c^o is the c^{th} entry of the one-hot vector y^o. Unlike standard adversarial attacks, here we do not need to constrain δ to lie within an ϵ ball. This is because the decoder is trained to produce realistic samples from any $z \sim P_{\text{z}}$ and the support of the prior distribution P_{z} is the whole latent space. As depicted in Fig. 8.2-(2), the work horse of our second phase training is the attack model to solve (8.1) in an iterative manner (for all adversarial attacks we perform 1k steps, using an L_{inf} norm with a step size $\alpha = 0.01$). The other networks in Fig. 8.2-(2) are not changed after phase (1).

By iterating to solve (8.1), the intermediate results for δ added to z^s create an attack path in the latent space (Fig. 8.1). Compared to simple linear interpolation in latent space, the proposed adversarial attack path has an important advantage: The adversarial loss (8.1) penalizes paths that pass through the class clusters except the target one. As shown in Fig. 8.1, the attacker mainly explores the empty regions between class clusters (i.e., decision boundaries of the latent space classifier) to reach the target, therefore being more efficient than linear interpolation in creating ambiguous samples. Feeding the latent codes along the attack path into the decoder $\mathcal{G}_\phi(z)$, Fig. 8.1 depicts a series of synthetic samples that smoothly diverge from the source x^s and approach a sample belonging to the target class y^o. The samples in-between realistically exhibit the features from both the source class y^s and the target class y^o, and possibly other classes if they are encountered on the attack path. In this work, we randomly sample source and target classes, though alternative heuristics can also be used. For example, based on the confusion matrix of a Baseline classifier, source and target class pairs can be selected based on the confusions among classes.

The labels of the samples can be obtained by applying the latent space classifier to the latent codes, i.e., $\mathcal{C}_\gamma(z^s + \delta)$. Unlike the one-hot encoded hard label vectors, the softmax responses can take on *soft* values between $[0,1]$. Since the perturbation δ may traverse through multiple class boundaries to reach the target, the soft labels are not simply based on y^s and y^o, and can have non-zero mass on other classes. Fig. 8.1 shows that the soft labels are semantically coherent with the samples synthesized by the decoder. Comparing with Mixup [Zhang et al., 2018], which linearly interpolates both the samples and their labels, the proposed adversarial attack always produces on-manifold ambiguous samples and labels them according to the class-specific features.

Using the attacker together with the BigBi-GAN pretrained models to create our OMADA augmentation set, we investigate two ways to sample the latent codes from the attack path in the latent space. The first, and default mode, samples uniformly along the path. The second approach favors samples whose soft labels yield large entropies. After proper normalization, we use the entropies of each latent code's soft label vector along the path to parameterize a probability mass function (pmf), and then sample the latent code according to such a constructed pmf.

8.4. On-manifold data augmentation

In order to solve the classification task, we train a DNN in the original input space x. As shown in Fig. 8.2-(3), the only difference is that we augment the original dataset with the OMADA set generated in Step (2) by sampling on data-manifold ambiguous samples together with their soft labels. Combining the two datasets has two effects. Firstly, the enlarged training set improves the generalization performance and reduces model uncertainty. As the size of the OMADA set can be unlimited by repeatedly sampling the latent space, it also prevents overfitting and memorization. Secondly, the DNN learns from the soft labels of OMADA to make soft predictions in addition to hard ones, tempering overconfidence in the training process and achieving an improved calibration performance at test time. In the subsequent experiment section, we find that soft labeling of ambiguous samples is particularly helpful to detect out-of-distribution samples.

8.5. Experimental setup

8.5.1. Setup and details

We evaluate and compare OMADA against multiple benchmark methods in the literature across 3 datasets, i.e., CIFAR-100, CIFAR-10 [Krizhevsky & Hinton, 2009], SVHN [Netzer et al., 2011] and 4 models, i.e., DenseNet ($L = 100$, $k = 12$) [Huang et al., 2017], Wide-ResNet 28-10 (WRN) [Zagoruyko & Komodakis, 2016], ResNeXt-29 [Xie et al., 2017], and VGG-16 [Simonyan & Zisserman, 2015]. We use the same generative model (i.e. OMADA augmentations) for all models for a given dataset. The benchmark methods from the literature primarily address data augmentation, label smoothing, and combinations of the two, similar to our proposed method, as well as stochastic Bayesian approximation methods and post-hoc calibration techniques.

The following is the list of methods we compare against: Baseline (trained without any data augmentation), Mixup ($\alpha = 0.1$) [Zhang et al., 2018], Manifold Mixup ($\alpha = 2.0$) [Verma et al., 2019], ϵ-smoothing ($\epsilon = 0.1$) [Szegedy et al., 2016], and CEDA [Hein et al., 2019]. Unless otherwise noted, hyperparameters are taken from the original publications. For Mixup, α is chosen based on the results from [Thulasidasan et al., 2019]. Further details about hyperparameters for individual methods can be found in Sec. B3. We also compare to stochastic Bayesian approximation methods, such as MC-Drop [Gal & Ghahramani, 2016] and Ensembles [Lakshminarayanan et al., 2017], and a number of post-hoc calibration techniques. The post-hoc calibration techniques include methods discussed in Chap. 7 such as temperature scaling (TS) [Guo et al., 2017], Gaussian process scaling (GP) [Wenger et al., 2020] and I-Max.

8.5.2. Deep neural network training

The training hyperparameters (learning rate, etc.) for each network are listed in the appendix (Sec. B3); these hyperparameters do not vary across datasets and methods. At the end of training, the model weights used for evaluation are chosen from the epoch with the best validation accuracy. Each reported result is the mean over 5 independent runs with the same hyperparameters. For all OMADA-trained classifiers we evenly balance each batch with 50% of the real training samples and 50% of the on-manifold adversarial samples. In order for these classifiers to be comparable to other baselines, we ensure that each epoch has the same number of updates as the Baseline (i.e. for each epoch the OMADA-trained classifiers only observe 50% of all real training samples).

8.5.3. Tasks

While the experimental investigation is primarily focused on uncertainty calibration, we also evaluate and compare the performance on multiple other metrics and tasks. In this section we present an overview of all experiments and their details. All experiments can be broken into the following tasks:

1. **Uncertainty calibration** (Sec. 8.6): Comparison of OMADA with all label smoothing methods at the task of uncertainty calibration. The calibration evaluation follows the evaluation discussed in Chap. 6. As a reminder, calibration measures the mis-match between the confidence and accuracy of the classifiers predictions.

2. **During-training vs. post-hoc calibration** (Sec. 8.7): A natural combination for further calibration gains involves applying *post-hoc* calibration techniques after employing *during-training* calibration. We applying post-hoc calibration to all classifiers and compare their performance.

3. **Generalization performance** (Sec. 8.8): Extensive comparison of all label smoothing methods for the following metrics: Accuracy, negative log-likelihood (NLL), and Brier score (mean-squared-error of predictions and hard labels).

4. **OMADA variants** (Sec. 8.9): Ablation of all OMADA variants and effect of changes to labels during training.

5. **Outlier detection** (Sec. 8.10): Comparison of label smoothing methods for the task of outlier detection. Outlier detection focuses on identifying out-of-distribution (OOD) inputs at test time based on thresholding the predicted uncertainty. This can evaluated using a completely different dataset, corrupted data, or classes from the same dataset not seen during training. The outlier detection experiments in this work focus on the former case; the classifiers are trained on CIFAR-10, and the predicted softmax is used to try and identify anomalous SVHN images at inference time. The metric used for evaluating outlier detection performance is the area under the receiver operating characteristic curve (OOD-AUC). Intuitively, this measures the ability of the uncertainty measure to binary classify an input as in-distribution or out-of-distribution over various thresholds. Additionally, one can evaluate the confidences assigned to OOD data using the mean-maximal-confidence (OOD-MMC), which is optimized when classifiers assign low confidences to unknown OOD data.

6. **Stochastic Bayesian neural network (BNN) approximations** (Sec. 8.11): Calibration performance of Bayesian approximation methods and comparison against OMADA.

7. **Sparsification** (Sec. 8.12): Investigation of the correlation between the prediction uncertainty and correctness by Sparsification [Yonatan & Ran, 2017]. Sparsification measures the correlation between the uncertainty estimates (in this case the softmax responses) and their true error. The Sparsification metric is optimized when the classifiers produce higher confidences for correctly classified samples than mis-classified samples.

8.5.4. Note on evaluation labels

We note that even though the training process utilizes the soft labels, they are not used for any evaluations. Despite the soft labels improving classifier trainings, they cannot provide a fair evaluation setup. In order to keep the evaluations fair for all methods, we only use the soft labels for training the label smoothing classifiers and use hard labels for all evaluations.

8.6. Uncertainty calibration

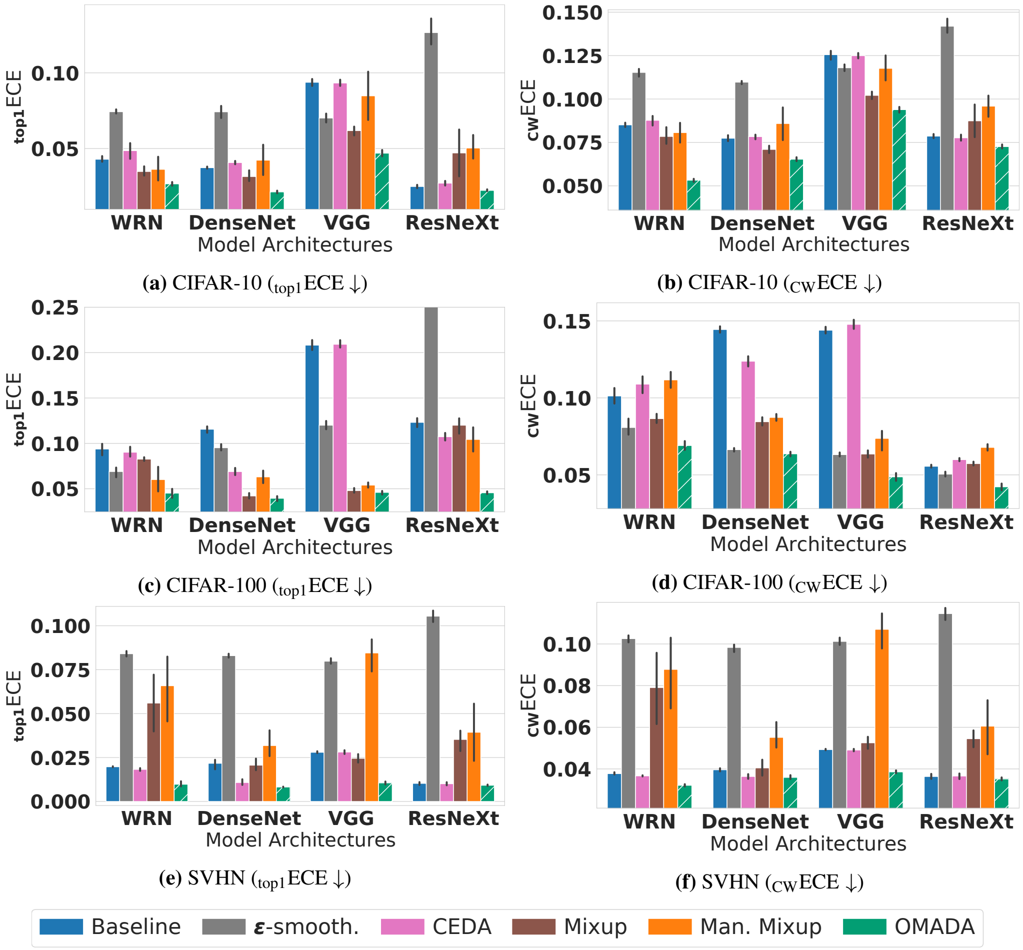
We first compare the calibration of OMADA against a range of baselines and competing methods. In Fig. 8.3, we evaluate the ${}_{\text{top1}}$ECEand ${}_{\text{CW}}$ECEperformance of the label smoothing methods for (a-b) CIFAR-10, (c-d) CIFAR-100 and (d-e) SVHN for all four model architectures on the in-distribution test set. We find that the observations are very similar for both ${}_{\text{top1}}$ECEand ${}_{\text{CW}}$ECE, with OMADA showing significant improvements across all datasets and model combinations compared to the baseline and all other methods. We observe larger performance gains for OMADA for harder datasets such as CIFAR-100 (Fig. 8.3(c-d)), as well as SVHN (Fig. 8.3(e-f)) where the dataset contains multiple class instances (digits) in the same image, introducing high uncertainty.

The stability of OMADA's performance across model architectures is remarkable. The selected architectures range from low capacity networks such as DenseNet to larger networks such as WRN and ResNeXt, as well as a network architecture with multiple dense layers (VGG). None of the compared methods achieves such low calibration errors across this diverse set of network architectures and multiple datasets, which demonstrates the benefits of OMADA for model-agnostic calibrated classifier training.

During the training of all classifiers in Fig. 8.3, no other augmentations were used as the baseline for all trained classifiers. The reason for this choice is to have a controlled setting where each method's effect on both calibration and generalization can be isolated. In order to be comparable against the literature, which often uses standard augmentation as the baseline [Thulasidasan et al., 2019], we reproduce Fig. 8.3 to show the results of all methods when trained using standard augmentation in Sec. B4.1 in the appendix and the observations are remain consistent.

8.7. Post-hoc vs. during-training calibration

In Chap. 7, we presented a novel post-hoc calibration technique and compared it against other previous state-of-the-art post-hoc techniques. These methods are applied on already-trained classifiers to refine their predictions for improved calibration. Given that *post-hoc* and *during-training* calibration techniques can be combined, in this section we analyze their combinations. We apply temperature scaling (TS) [Guo et al., 2017], Gaussian process scaling (GP) [Wenger et al., 2020] and I-Max (Chap. 7) to all label smoothing methods. We visualize the ${}_{\text{top1}}$ECEperformance for



(a) CIFAR-10 ($_{\text{top1}}$ECE ↓)

(b) CIFAR-10 ($_{\text{CW}}$ECE ↓)

(c) CIFAR-100 ($_{\text{top1}}$ECE ↓)

(d) CIFAR-100 ($_{\text{CW}}$ECE ↓)

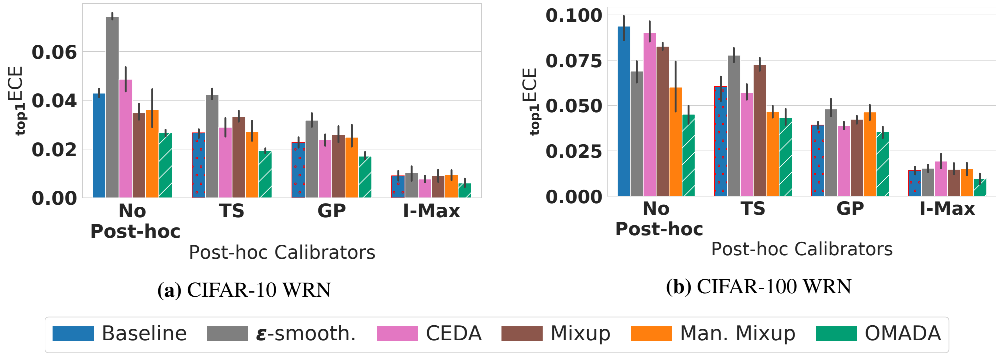
(e) SVHN ($_{\text{top1}}$ECE ↓)

(f) SVHN ($_{\text{CW}}$ECE ↓)

Figure 8.3.: $_{\text{top1}}$ECEand $_{\text{CW}}$ECEcalibration performance of label-smoothing methods on in-distribution test dataset (a-b) CIFAR-10, (c-d) CIFAR-100 and (e-f) SVHN. Hatched bars indicate the best-performing method. Error bars are ± 1 std. dev. over 5 runs. Across all datasets and model architectures, OMADA achieves better performance than all other methods at both calibration metrics.

CIFAR-10 and CIFAR-100 on WRN in Fig. 8.4. We find that applying post-hoc calibration techniques to *all* label smoothing methods improves the calibration performance (i.e. applying any post-hoc technique is better than no post-hoc calibration) with the best performance gains seen with I-Max. Using I-Max for all label smoothing methods, the best performance is achieved by OMADA, showing that the best combination of *post-hoc* and *during-training* calibration is OMADA with I-Max for improving the ECE. We also observe that OMADA, without any post-hoc calibration, can yield better performance than simple post-hoc methods such as TS on the Baseline (e.g. CIFAR-100).

Interestingly, we find that some label smoothing methods (e.g. ϵ-smoothing and Mixup) perform worse than the Baseline (i.e. blue bars with red dots) after applying post-hoc calibration. This is an unintuitive effect, though further investigation showed that usually in the case where the calibration error before post-hoc calibration is already fairly low. This can happen as the NLL loss which is optimized for most post-hoc methods, such as TS and GP, is not directly correlated with ECE. For further explanation, and an example of this phenomenon, see Sec. B4.2 in the appendix where we use temperature scaling as an example for the analysis.



(a) CIFAR-10 WRN

(b) CIFAR-100 WRN

Figure 8.4.: ${}_{\text{top1}}$ECEperformance of all classifiers after applying post-hoc calibration methods (Temperature Scaling (TS), GP Scaling (GP) and I-Max) on WRN classifier for (a) CIFAR-10 and (b) CIFAR-100. Hatched bars indicate the best-performing method. Error bars are ± 1 std. dev. over 5 runs. Even though the over-all performance of each label smoothing method can be improved with post-hoc calibration, the performance ranking of these methods can change after post-hoc calibration. For example, ϵ-smoothing and Mixup after TS or GP scaling do not perform as well as applying TS or GP directly on the Baseline classifier (i.e. blue bars with red dots). We find that OMADA performs much better after post-hoc calibration with the best performance seen after applying I-Max. For each post-hoc calibration method, OMADA consistently yields better performance than directly applying post-hoc calibration to the Baseline classifier (i.e. trained without any during-training calibration). We also observe that OMADA, without any post-hoc calibration, also performs better than simple post-hoc techniques like TS.

8.8. Generalization performance

In addition to improving the calibration performance, *during-training* calibration techniques also have the potential to improve the generalization performance of the classifiers. In Tab. 8.1, we evaluate the Accuracy, negative log-likelihood (NLL) and Brier score of all CIFAR-10 and CIFAR-100 classifiers. Unlike methods such as ϵ-smoothing and CEDA which can degrade the generalization performance for some dataset-model settings, we observe that Mixup, Manifold Mixup and OMADA consistently offer improved generalization. We observe the best generalization gains for OMADA across all datasets and model architectures. This indicates that the increased calibration performance obtained by OMADA does not come at the expense of generalization, but rather significantly increases the generalization performance. Despite being trained with soft labels, OMADA also yields the best performance on metrics which are evaluated with the *hard* labels (e.g. NLL and Brier score). The former metric measures the cross entropy divergence between the predictions and *hard* labels, and the latter measures the mean-squared-error between the predictions and *hard* labels. This shows that OMADA maintains high confidences for the correctly-classified samples which is required for optimizing the NLL and Brier score.

Table 8.1.: Generalization performance (Accuracy, NLL and Brier) of all classifiers for the in-distribution test datasets CIFAR-10 and CIFAR-100. The numbers reflect the mean across the 5 splits and the best performing method is made bold. OMADA shows significant gains on the Accuracy, NLL and Brier compared to all other methods. In addition to improving the calibration performance, we observe that OMADA also yields good generalization performance.

	WRN			DenseNet			VGG			ResNeXt		
Method	Acc ↑	NLL ↓	Brier ↓	Acc ↑	NLL ↓	Brier ↓	Acc ↑	NLL ↓	Brier ↓	Acc ↑	NLL ↓	Brier ↓
	CIFAR-10											
Baseline	91.44	0.297	0.134	92.80	0.259	0.113	86.10	0.641	0.228	92.17	0.242	0.117
ϵ-smooth.	91.36	0.366	0.139	92.35	0.336	0.127	86.30	0.530	0.211	92.34	0.366	0.137
CEDA	91.16	0.315	0.140	92.80	0.263	0.115	86.19	0.634	0.228	92.56	0.239	0.114
Mixup	92.58	0.268	0.115	93.35	0.236	0.104	87.62	0.449	0.194	93.32	0.243	0.105
Man. Mixup	93.28	0.275	0.110	93.58	0.255	0.103	88.86	0.436	0.185	93.14	0.248	0.105
OMADA	**95.72**	**0.161**	**0.070**	**94.56**	**0.171**	**0.083**	**89.73**	**0.388**	**0.163**	**93.57**	**0.209**	**0.099**
	CIFAR-100											
Baseline	69.26	1.217	0.431	72.58	1.126	0.399	58.04	2.109	0.615	72.24	1.245	0.414
ϵ-smooth.	69.21	1.366	0.434	72.35	1.138	0.395	61.51	1.785	0.535	74.46	1.752	0.544
CEDA	69.88	1.179	0.425	73.72	0.943	0.367	57.94	2.088	0.617	75.02	1.091	0.376
Mixup	71.10	1.192	0.409	74.19	0.971	0.359	62.43	1.574	0.502	73.44	1.174	0.398
Man. Mixup	71.60	1.130	0.400	76.23	0.906	0.340	61.41	1.566	0.511	73.24	1.109	0.389
OMADA	**74.45**	**0.978**	**0.358**	**76.50**	**0.848**	**0.330**	**64.32**	**1.397**	**0.477**	**74.90**	**0.886**	**0.349**

8.9. OMADA variants

In this section, we show the performance of different OMADA variants, to investigate the effects of adding ambiguous images and soft labels independently. First, we investigate an alternative sampling method, which preferentially samples images from the path with high label entropies (i.e. higher chance of sampling pure boundary region samples), as opposed to uniformly sampling them. We call this variant OMADA-SE (*sample from entropy*). Furthermore, we study the effect of the soft labels produced by the latent-space classifier by training the networks with the generated ambiguous samples from OMADA and OMADA-SE, but changing the labels. We either harden the soft labels based on the maximum class probability (OMADA*-H), or change the class labels to be uniform across all classes (OMADA*-U). We investigate the resulting network calibration ($_{\text{top1}}$ECEand $_{\text{CW}}$ECE), Accuracy, NLL and the outlier detection performance ($_{\text{OOD}}$AUC) of the variants. The results are shown in Tab. 8.2 for CIFAR-10 on WRN and DenseNet.

We observe that using this alternative sampling method performs very competitively on multiple tasks, especially on outlier detection. The effect of hardening the labels yields surprisingly good results on $_{\text{top1}}$ECE, where it sometimes improves calibration over the corresponding soft label variant, suggesting that the ambiguous images generated by the on-manifold attacks alone are enough to improve the network's confidence estimates. However, this gain comes at the cost of Accuracy and NLL, suggesting that the soft labels help generalization. This observation will be the focus of future research.

The effect of hardening labels is different for the sampling variants; as OMADA-SE contains more samples with higher entropy soft labels, the change in label density is much more drastic than in OMADA, which also produces samples far away from decision boundaries (i.e. the soft label is already relatively hard). This is illustrated especially in the outlier detection performance: here, OMADA-H and OMADA-SE-H suffer in comparison to their soft-label counterparts.

Next, we study the effect of assigning uniform class labels for each ambiguous sample generated by the adversarial attack. The results show that the soft labels of the ambiguous samples are required to attain competitive calibration and generalization performance for in-distribution data. However, for out-of-distribution samples, where the OOD-AUC is optimized when predicting near uniform class labels on OOD data, the OMADA*-U networks do very well. This is consistent with observations from CEDA [Hein et al., 2019], where uniform class labels are also used to improve detecting OOD samples (shown in Fig. 8.5). In summary, changing the soft labels increases performance on some tasks, but degrades performance across other tasks; the best choice of labels is then task-dependent. On average, the soft labeled methods (OMADA and OMADA-SE) perform stably across tasks.

Table 8.2.: Performance of OMADA ablation methods on calibration (${}_{\text{top1}}$ECE, ${}_{\text{CW}}$ECE), generalizatioon (Accuracy, NLL), and outlier detection (${}_{\text{OOD}}$AUC). OMADA-SE refers to the sample entropy variant of OMADA. *-H refers to the respective hard label variant and *-U refers to the respective uniform label variant. Bold entries indicate the best-performing method. We report the mean over 5 independent runs for each method.

Method	Acc ↑	NLL ↓	${}_{\text{top1}}$ECE ↓	${}_{\text{CW}}$ECE ↓	${}_{\text{OOD}}$AUC ↑	Acc ↑	NLL ↓	${}_{\text{top1}}$ECE ↓	${}_{\text{CW}}$ECE ↓	${}_{\text{OOD}}$AUC ↑
			WRN					DenseNet		
OMADA	95.72	0.161	0.027	0.053	0.930	94.56	**0.171**	0.021	0.065	0.936
OMADA-H	95.26	0.163	**0.024**	0.059	0.928	93.80	0.186	**0.019**	0.070	0.899
OMADA-U	95.67	0.187	0.031	0.052	0.978	93.87	0.205	0.032	0.069	0.972
OMADA-SE	**95.96**	**0.153**	0.026	**0.050**	**0.987**	**95.00**	0.195	0.032	**0.061**	**0.983**
OMADA-SE-H	95.95	0.156	0.027	0.051	0.967	94.28	0.209	0.033	0.066	0.918
OMADA-SE-U	95.15	0.183	0.029	0.059	0.938	94.68	0.217	0.035	0.064	0.981

8.10. Outlier detection

In order to put the outlier detection abilities of OMADA-SE (the best variant across multiple tasks) into context, we compare the ${}_{\text{OOD}}$AUCto the already-investigated label smoothing methods (Fig. 8.5). OMADA-SE outperforms all other methods on both DenseNet and WRN, albeit with a small gap to CEDA on DenseNet. The good performance of CEDA is not surprising, as it is implicitly trained to predict lower confidence on out-of-distribution samples (in CEDA these are random noise images). Interestingly, soft-labels alone are not enough to result in good outlier detection, as evidenced by the poor performance of ϵ-smoothing.

Another measure for evaluating the over-confidence of networks is to measure the ${}_{\text{OOD}}$MMC. For out-of-distribution (OOD) samples we want the network to assign a confidence of $\frac{1}{\#\text{ classes}}$, reflecting maximum uncertainty. The mean-maximal-confidence on OOD data measures how well the network performs the task of assigning a low confidence to unseen samples. For ${}_{\text{OOD}}$MMC, we observe that OMADA-SE performs better than all other methods except CEDA, which has the lowest ${}_{\text{OOD}}$MMCon DenseNet. As CEDA augments the training dataset with out-of-distribution images and uniform class labels, it has an advantage for the ${}_{\text{OOD}}$MMCmetric which is optimized by predicting high entropy labels (i.e. uniform class labels). Though for WRN, OMADA-SE again performs best compared to all other methods.

8.11. Stochastic Bayesian neural network (BNN) approximations

We compare the calibration, generalization and outlier detection performance of the OMADA variants to two stochastic Bayesian neural network (BNN) approximations: Monte Carlo (MC)

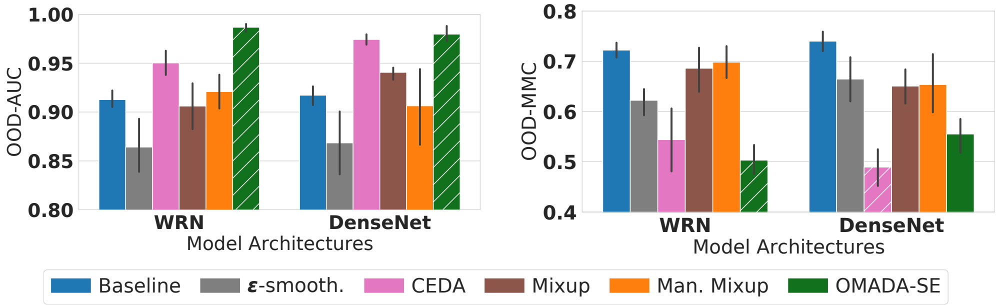
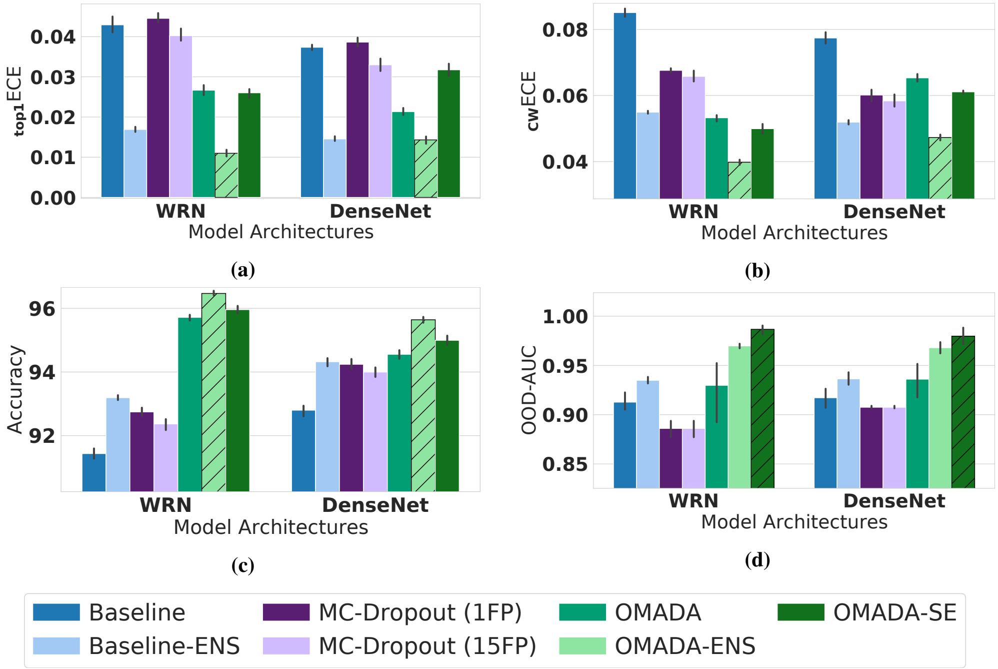


Figure 8.5.: Outlier detection performance of label smoothing methods for CIFAR-10 on WRN and DenseNet. We evaluate the (a) ${}_{\text{OOD}}$AUCand (b) ${}_{\text{OOD}}$MMC. We observe that OMADA-SE also yields good outlier detection performance, especially for WRN. For ${}_{\text{OOD}}$MMC, we observe that the best performing methods are CEDA and OMADA-SE. Both methods have significantly lower confidence assigned to OOD data compared to all other methods. Hatched bars indicate the best-performing method. Error bars are ± 1 std. dev. over 5 runs.

Dropout [Gal & Ghahramani, 2016] and Ensembles [Lakshminarayanan et al., 2017]. These are commonly used to obtain uncertainty estimates and have been shown to improve uncertainty calibration, though are not real-time solutions and are computationally expensive at test-time. As with post-hoc calibration techniques, these methods are both orthogonal to OMADA, and can be easily combined. Fig. 8.6 shows the performance (${}_{\text{top1}}$ECE, ${}_{\text{CW}}$ECE, Accuracy and ${}_{\text{OOD}}$AUC) for Ensemble and MC-Dropout (with 1 and 15 forward passes) for CIFAR-10 on DenseNet and WRN. Each ensemble entry reports the mean and standard deviation across 5 ensembles, where each ensemble contains 5 classifiers. As none of the classifiers reported in the paper are trained with Dropout, we specially train DenseNet and WRN with dropout (0.20 and 0.30, respectively) in order to compare against MC-Dropout. As these classifiers can be considered to have a different model architecture compared to their no-dropout counter-parts, we report the single deterministic forward pass through the network (Dropout-1FP) and compare this to the 15 stochastic forward passes (Dropout-15FP). For calibration, we observe that ensembles (*-ENS) greatly improve the ${}_{\text{top1}}$ECEand ${}_{\text{CW}}$ECEfor both models, and that MC-Dropout is only effective at ${}_{\text{CW}}$ECEbut still not as good as ensembles. The relatively poor performance observed by MC-Dropout (e.g. outlier detection becoming worse than the baseline) is consistent with the literature which found also found MC-Dropout to underperform [Foong et al., 2019; Ovadia et al., 2019], especially on out-of-distribution tasks [Shafaei et al., 2019; Ritter et al., 2018]. We observe that even though Baseline-ENS yields better performance than OMADA, after applying ensembles, OMADA-ENS yields the best performance overall. This shows that the benefits of ensembles can best be seen after combining with OMADA. It should be noted that one caveat of ensembles is the increased computational complexity required at test-time.

In terms of Accuracy, we observe that both ensembles and MC-Dropout improve compared to the Baseline classifier, though are still significantly lower than all OMADA variants (with and without ensembles). On the other hand, at ${}_{\text{OOD}}$AUCusing ensembles help but MC-Dropout performs worse than even the Baseline. We find that the best outlier detection performance is observed by OMADA-ENS and OMADA-SE. OMADA-ENS improves on OMADA alone, though interestingly it does not perform as well as OMADA-SE (which has much more samples with high entropy soft labels and confusing samples).



(a)

(b)

(c)

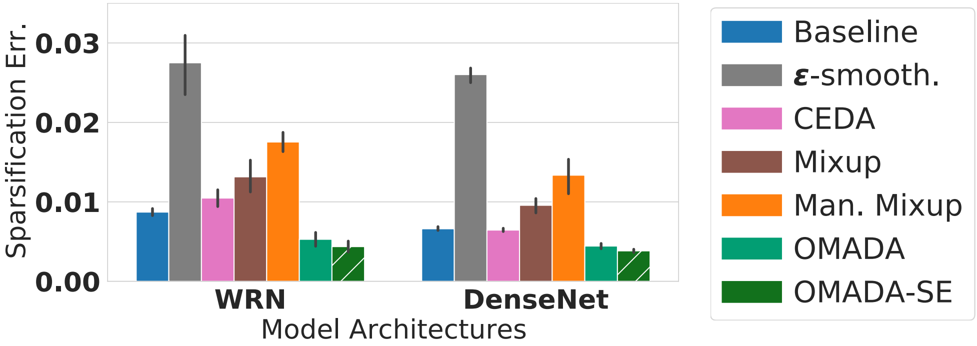
(d)

Figure 8.6.: Comparison of OMADA classifiers with stochastic BNN approximations (Ensembles and MC-Dropout) for CIFAR-10 on DenseNet and WRN. We compare the calibration performance ((a) $_{\text{top1}}$ECEand (b) $_{\text{CW}}$ECE), (c) generalization Accuracy and (d) outlier detection ($_{\text{OOD}}$AUC). Methods which involve an ensemble are denoted with "*-ENS" (Ensemble of 5 networks) and report the mean and standard deviation across 5 sets of ensembles. The Dropout networks are specially trained networks with dropout and we report the results when using 1 deterministic forward pass (MC-Dropout (1FP)) and 15 stochastic forward passes (MC-Dropout (15FP)). Hatched bars indicate the best-performing method. Error bars are ± 1 std. dev. over 5 runs.

8.12. Sparsification

In this section, we report the Sparsification errors on the in-distribution test set. Sparsification evaluates how well a given uncertainty estimate correlates with the true error; intuitively, we want the classifiers to be more confident on correctly-classified predictions, and less confident on incorrect predictions [Yonatan & Ran, 2017]. This is calculated by selectively calculating the classification accuracy on increasingly large subsets of the test set. Samples are added to the subset based on their uncertainty with the more certain samples are added first. This metric is optimized with predicting all incorrectly-classified samples with higher uncertainty than all correctly-classified samples. The final error is calculated by computing the divergence to this ideal case. A lower Sparsification error is desired. These results can be seen in Fig. 8.7. We observe that OMADA-SE significantly improves its Sparsification error compared to all other label smoothing methods.

8.13. Conclusion



(a) CIFAR-10 Sparsification Error

Figure 8.7.: Sparsification error of the label smoothing methods for CIFAR-10 on WRN and DenseNet. For Sparsification we observe that OMADA-SE has a significantly lower error compared to all other methods. Error bars are ± 1 std. dev. over 5 runs.

Importance of the Generative Model OMADA relies on a meaningful approximation of the data manifold, which allows finding ambiguous examples that lie close to the class decision boundaries. Utilizing GANs is currently the best suited solution for complex high dimensional data, and OMADA demonstrates a beneficial use of the learned latent space apart from realistic data synthesis. GANs and other generative models are notorious for their training instability, but this is also a very active research field, where better tools and algorithms become available quickly, and more and more pre-trained models are made available. In our case OMADA delivered the best results with the most realistic generative model, the state-of-the-art BigBiGAN [Donahue & Simonyan, 2019], and using our principled targeted sampling provided more robust improvements across all networks and architectures than simpler techniques working in image space. Hence, we expect that both uncertainty estimates and accuracy will improve with even better generative models and better approximations of the manifolds that will become available in the future, without requiring changes to the sampling principles of OMADA. In terms of computational complexity, the main costs are only incurred when initially training the generative model. OMADA and the core idea is of course not limited to GANs, but can utilize any autoencoder-based generative model which might be better suited for other data modalities (e.g. VAEs [Kingma & Welling, 2014], VAE-GAN [Rosca et al., 2017] and VQ-VAE2 [Razavi & van den Oord an Oriol Vinyals, 2019]).

Benefits of manifold-based calibration OMADA can be directly applied to any data modality for which a generative model can be trained, including text, time series, spectrograms, or point clouds, e.g. for radar or LIDAR data. Other methods such as Mixup [Thulasidasan et al., 2019] perform linear interpolation in the image space, which is not feasible if the dimensionality of the data changes, and might result in meaningless intermediate images (e.g. when interpolating between spectra). CEDA [Hein et al., 2019] generates uniform random-noise images for calibration, but this might not be a good approximation of out-of-data samples for non-image data.

Principled Sampling of Ambiguous Data Points OMADA's principled sampling technique for data augmentation maximizes the chances of sampling truly ambiguous on-manifold samples that improve calibration, as apposed to naive sampling (which has only a tiny chance of sampling points near the boundary). The targeted on-manifold adversarial attack of OMADA deliberately samples novel training points with multiple class-specific features, sharing the characteristics of all neighboring classes. By training the image-space classifiers on such ambiguous samples a high uncertainty in those regions is learned, rather than an over-confident prediction for a single class.

Model-agnostic Calibration One striking observation of our experiments (e.g. Fig. 8.3) is that the calibration quality of all competing methods varies strongly if tested on different network architectures, whereas OMADA consistently ranks among the top performers. We chose the tested model architectures because of their varying capacities, layer types (e.g. VGG's MLPs, Residual blocks in ResNeXt), and their different peak accuracies. Different downstream tasks (e.g. semantic segmentation or object detection), different data modalities (e.g. point clouds or word embeddings), and different hardware constraints for embedded solutions might require different types of network architectures. Therefore it is highly desirable to have a method which is agnostic to the diversity of network architectures found in practice.

In this article we have introduced the concept of on-manifold adversarial data augmentation for uncertainty estimation by leveraging recent advances in generative modeling. By combining a latent space classifier on the approximated data manifold with on-manifold adversarial attacks we derive a novel sampling procedure, which generates samples specifically in challenging regions close to decision boundaries. The OMADA dataset is generated by using the decoder network to project back into the image space, and using soft labels derived from the latent space classifier. Through a range of carefully chosen experiments, we study the effect of OMADA as a data augmentation method for training an independent image space classifier.

An extensive set of experiments show significant improvements across multiple datasets and diverse network architectures, as well as on multiple tasks. The stability of the OMADA results for ECE across multiple networks is a particularly desirable property, as most alternative methods fail to perform well across all investigated networks. OMADA can be combined with post-processing methods such as temperature scaling [Guo et al., 2017], and we are confident that further beneficial combinations and extensions of the key concept of OMADA will be discovered in future research. Furthermore, we show that OMADA always results in increased classification accuracy compared to baselines with and without data augmentation, and all other competing methods. Finally, OMADA-SE is presented as a method to focus on data generation in boundary regions, thereby outperforming all other methods for outlier detection.

This is a first step towards improving uncertainty quantifications for deep networks through on-manifold adversarial samples. Initial results show significant improvements of the networks ability to assign confidence to its predictions on in-distribution samples as well as out-of-distribution samples. Further studies are required to investigate the behavior of these networks on data which marginally leaves the data manifold (e.g. unseen transformations or corruptions).

In combination with I-Max (Chap. 7), OMADA covers another large part of litearture gap LG3 by offering a data augmentation strategy for optimizing the uncertainty calibration of deep learning classifiers during the training process. This chapter offered a real-time uncertainty estimation technique which additionally optimizes generalization and outlier detection performance. Different to post-hoc calibration techniques, such solutions allow classification models in the litearture to jointly optimize multiple objectives while implicitly optimizing calibration during training. In order to study uncertainty calibration in the context of radar classification, the next part will focus on improving the confidence estimates of the radar classifiers used in Part I.

Part III.

Radar uncertainty calibration

Chapter 9.

Problem formulation and literature review

With a focus on radar classification deep learning and uncertainty calibration of deep learning classifiers in Part I and Part II, respectively, we now shift the focus to the combination of the two topics. In this part, we will focus on studying the predictive uncertainty estimation quality of radar classifiers from Part I.

9.1. Problem formulation

Consistent with the literature (see review and discussion in Sec. 6.6), the uncertainty calibration methods presented, in Chap. 7 and Chap. 8, both tackle the well known over-confidence and mis-calibration issues of deep learning classifiers. In summary, these results show that despite the highly *accurate* performance of these classification models, their application to real-world tasks has been greatly limited due to their unreliability in unknown situations. Despite, the majority of such research focusing on vision classifiers, these characteristics transfer to all neural network-based classification models regardless of the data modality.

As described earlier (in Chap. 3), radar spectra measurements are very susceptible to noise, especially for objects measured at a distance and small non-reflective objects which yield low signal-to-noise ratio (SNR) measurements. Some situations with low SNR measurements do not provide sufficient information in the spectra to accurately and confidently conclude on a classification (will be empirically shown in Chap. 11). Therefore, it is important that the classifier predictions can be used to distinguish such situations as it is important for downstream decision-making systems to be aware of the uncertainty inherent in the measurements. For example, it should become clear from the uncertainty estimates when the classifier is guessing.

An additional level of difficulty in radar stems from challenges in directly applying deep learning for radar classification (e.g. dataset size, diversity, imbalance; see Sec. 3.2). Already strenuous to yield highly-accurate radar classifiers, these challenges additionally worsen the over-confidence and mis-calibration issues of deep learning models. For example, the small dataset size make the classifiers more prone to over-fitting and subsequently yielding more over-confident predictions. The sensitivity of the sensor to view point changes makes clear that radar classifiers will frequently encounter novel unseen measurements and it is important that these situations can be detected through the uncertainty estimates of the classifier.

In the following Chap. 10, we shall investigate a first study of the uncertainty estimation of deep learning-based radar classifiers, and present a first solution for addressing these issues using post-hoc calibration techniques discussed in Chap. 7 (e.g. I-Max calibration). We conclude this part with Chap. 11, which proposes a novel soft label training strategy for improving uncertainty calibration using radar-specific knowledge to estimate the soft labels. The new findings confirm that the quality of the uncertainty estimates of radar classifiers are, similar to or even worse than vision classifiers, over-confident and mis-calibrated. This part, firstly, identifies a large gap (LG3) in literature of deep learning-based radar classification which is a vital open problem but remains, thus far, largely unaddressed; despite the increased interest in exploiting deep learning for radar classification in safety-critical applications. Secondly, this part also presents the first attempts in addressing this literature gap, which is a prerequisite for safe, robust and reliable classification based on radar sensors.

Using the solutions proposed in Chap. 10 and Chap. 11, we find that these issues of over-confidence in radar classifiers can be largely mitigated. This includes application of general uncertainty methods developed for vision (e.g. Chap. 7), as well as a novel technique which is designed to incorporate radar knowledge for uncertainty estimation. In summary, the results in this part show that these problems can be successfully addressed to allow reliable radar predictions with accurate uncertainty estimation.

9.2. Literature review

As radar classification uncertainty is still an open problem and mostly an unaddressed gap in the literature, this literature review is rather short. The work done in this part is among the first to investigate uncertainty estimation for radar classification.

For coastal surveillance, the authors of Jochumsen et al. [2015] used a modified version of a Gaussian mixture model (GMM) [Reynolds, 2009], the DeltaGMM [Jochumsen et al., 2014] model, and used the predicted uncertainty to filter unwanted targets. Estimated based on the target locations, the classification was based on the speed uncertainty of the targets. The authors also observed over-confidence in the predictions from the GMM-based classifier. This over-confidence is expected to become worse for deep learning classifiers, which have much larger modeling capacities and tend to use this capacity to memorize the *over-confident* ground truth labels. On synthetic radar signals, the authors of [Meronen et al., 2020] proposed using MC-Dropout [Gal & Ghahramani, 2016] with a new non-linear activation function and visualized the entropy distribution of out-of-distribution samples showing the over-confidence phenomena, but no outlier detection was studied.

A more detailed literature review of deep learning algorithms for radar can be found in Sec. 3.3, and the literature of uncertainty estimation is discussed in Sec. 6.6. We are not aware of any application of the uncertainty methods discussed in Sec. 6.6 to radar classification.

In addition to a lack of methods to improve the uncertainty estimation of radar classifiers, the radar literature also lacks works which *evaluate* the performance beyond accuracy (i.e. LG2). In the next chapter, we study the uncertainty calibration of the radar classifiers trained in Part I and discuss the negative consequences in neglecting to detect over-confidence and mis-calibration in radar classifiers. The focus of the remaining chapters in this part is to provide the first works in the radar literature to offer solutions for tackling these problems. We present the improvements which can be made by simply applying post-hoc solutions from the machine learning domain (Chap. 7) to the radar domain in Chap. 10, as well as offer a novel radar-specific solution for further performance gains in Chap. 11.

Chapter 10.

Post-hoc calibration for radar classifiers

Contribution: Deep learning (DL) has recently attracted increasing interest to improve object type classification for automotive radar. In addition to high accuracy, it is crucial for decision making in autonomous vehicles to evaluate the reliability of the predictions; however, decisions of DL networks are non-transparent. Current DL research has investigated how uncertainties of predictions can be quantified, and in this chapter, we evaluate the potential of these methods for safe, automotive radar perception. In particular, we evaluate how uncertainty quantification can support radar perception under (1) domain shift, (2) corruptions of input signals, and (3) in the presence of unknown objects. We find that in agreement with phenomena observed in the literature, deep radar classifiers are overly confident, even in their wrong predictions. This raises concerns about the use of the confidence values for decision making under uncertainty, as the model fails to notify when it cannot handle an unknown situation. Accurate confidence values would allow optimal integration of multiple information sources, e.g. via sensor fusion. We show that by applying state-of-the-art post-hoc uncertainty calibration, the quality of confidence measures can be significantly improved, thereby partially resolving the over-confidence problem. Our investigation shows that further research into training and calibrating DL networks is necessary and offers great potential for safe automotive object classification with radar sensors.

10.1. Overconfidence and mis-calibration

Although DNNs typically have softmax-outputs that provide a pseudo-confidence for each prediction, it is known that they predict values that are not usable as true measures of uncertainty, and require uncertainty calibration [Guo et al., 2017]. Without uncertainty calibration, DNNs often exhibit extreme overconfidence in their predictions, regardless of their correctness, as well as for unknown inputs [Hein et al., 2019; Kristiadi et al., 2020]. A well-calibrated network, instead, will associate every output with an uncertainty measure that approximates the probability of misclassfication.

In this chapter, we investigate for the first time the weaknesses of DNNs in providing accurate confidence measures for radar object type classification, and the potential consequences. Our work focuses on the utility of computationally efficient post-hoc uncertainty calibration methods (discussed and introduced in Chap. 7) to improve the confidence predictions of radar classifiers. Such solutions can be easily applied to already trained networks, and require no further retraining of the network parameters. We additionally investigate the role that the construction of the test set plays in detecting over-confidence. If training and test splits are similar in their distribution, seemingly high accuracy might be misleading in evaluating the true generalization performance on unseen real-world data.

Radar perception requires dealing with multiple factors that introduce uncertainty, e.g. 1) corrupted signals due to noise, interference, and environmental effects, 2) ambiguities between different objects when viewed from different angles and distances, 3) variability within different instances of the same object class, and 4) domain shifts between the datasets used for training and testing. As radar signals are hard to interpret for humans, it is crucial to have objective measures that quantify the effects of such confounding factors.

We focus our experiments on evaluating the generalization and calibration performance of DNNs which classify 7 different categories of objects. Additional tests investigate how artificially introduced input corruptions, which attempt to mimic expected real-world perturbations, affect both accuracy and confidence estimates. As expected, stronger corruptions reduce the accuracy of DNNs, but we also observe an increasing deterioration of confidence estimates. Finally, tests show that the over-confidence phenomena remains even for objects never previously seen.

Observations reported in this article highlight that current DL approaches for radar classification struggle to produce trustworthy confidence estimates for their predictions, even if on average their classification accuracy is very high. This indicates that further research is needed to provide downstream decision making modules with high quality uncertainty estimates, in addition to highly accurate predictions. Our work suggests that accuracy on a fixed test set alone is insufficient to predict the performance under real-world conditions, and calibrated confidence measures, in particular on corrupted and unseen data, should always be evaluated in conjunction.

10.2. Post-hoc calibration

In Part II, we discussed a family of approaches, to improve uncertainty calibration of DNNs, that are relatively simple to implement: post-hoc calibration (also see Chap. 7). The simplest approach, temperature scaling (TS [Guo et al., 2017]), uniformly scales all confidences by a learned temperature T, obtained by optimizing the negative log-likelihood on the validation set (in effect uniformly reducing/increasing confidence). While extremely simple to use, this approach has limited expressive power, and more recent approaches yield better calibration gains. Two such methods are used in this paper to demonstrate the effectiveness of post-hoc methods to improve network calibration. The first is an improved scaling method using a latent Gaussian processes (GP) [Wenger et al., 2020], and the second is a discrete binning method based on maximizing the mutual information between the network outputs and the labels, I-Max (Chap. 7). It should be noted that these families of methods take as input the predicted probabilities of the deep network and further refine them to generate more accurate confidences. For more details on each of these methods, please refer to Chap. 7.

10.3. Experiments setup

In order to improve the calibration performance of radar classifiers, we compare three post-hoc calibration techniques (TS, GP and I-Max) on two challenging radar test datasets introduced in Chap. 4.

10.3.1. Deep neural network training

We use the same architecture and training procedures as described in Chap. 5.2. We train the classifiers only using Env1-Train and select the model with the best accuracy on the independent validation set Env1-Valid. We evaluate the performance of the classifiers on two unseen test datasets: Env2-Test and Env3-Test. We train 10 independent classifiers with different initializations and report their mean and standard deviations for all experiments.

10.3.2. Methods

We compare the Baseline classifier (with no post-hoc calibration) with temperature scaling (TS) [Guo et al., 2017], Gaussian process scaling (GP) [Wenger et al., 2020] and I-Max (Chap. 7). Additionally, the combination of GP and I-Max (i.e. I-Max with GP) which offers a real-time solution and exploits the benefits of both GP and I-Max.

10.3.3. Tasks

While the experimental investigation is primarily focused on uncertainty calibration, we also evaluate and compare the performance on multiple other metrics and tasks. In this section we present an overview of all experiments and their details. All experiments can be broken into the following tasks:

1. **Over-confident predictions** (Sec. 10.4): Showing the over-confidence issue of DNN classifiers through visualization of the Baseline classifiers' prediction distribution.

2. **Uncertainty calibration** (Sec. 10.5): The calibration evaluation follows the evaluation discussed in Chap. 6. As a reminder, calibration measures the mis-match between the confidence and accuracy of the classifiers predictions. For a qualitative comparison we visualize the reliability diagrams.

3. **Quantitative evaluations of all metrics** (Sec. 10.6): Extensive comparison of all methods for the following metrics: Accuracy, ${}_{\mathrm{top1}}$ECE, ${}_{\mathrm{CW}}$ECE, and mean-maximal-confidence of incorrect samples (MMC$_{\mathrm{inc}}$).

4. **Spectra corruptions** (Sec. 10.6.1): Evaluation of the performance of the classifiers under dataset shifts by studying their behavior after synthetically corrupting the spectra under various corruptions.

5. **Out-of-distribution spectra** (Sec. 10.6.2) A study of the behaviour of the Baseline classifier to out-of-distribution spectra samples.

10.4. Over-confidence of predictions

While systems deployed for real-world tasks always require highly accurate predictions, we show the importance of detecting when these predictions are over-confident on wrong, unseen and unknown data. In Fig. 10.1, we depict the distribution of the confidences of the top predicted class, categorized into correct (blue) and incorrect (red) classifications. Ideally, correctly classified samples should exhibit higher confidences than the incorrectly classified samples. In Fig. 10.1a, it can be seen that for the Baseline classifier most of the correctly classified samples have high confidences (skewness towards a confidence of 1.0), but similar trends can also be seen for the incorrect samples, i.e. misclassifications are assigned relatively high confidences. This phenomenon is harmful as the model outputs confident predictions, regardless of whether the classifier was able to correctly recognize the object or not. Using a post-hoc uncertainty calibration method such as GP [Wenger et al., 2020], the severity of this harmful effect can be largely reduced (i.e. the red distribution is shifted significantly left).

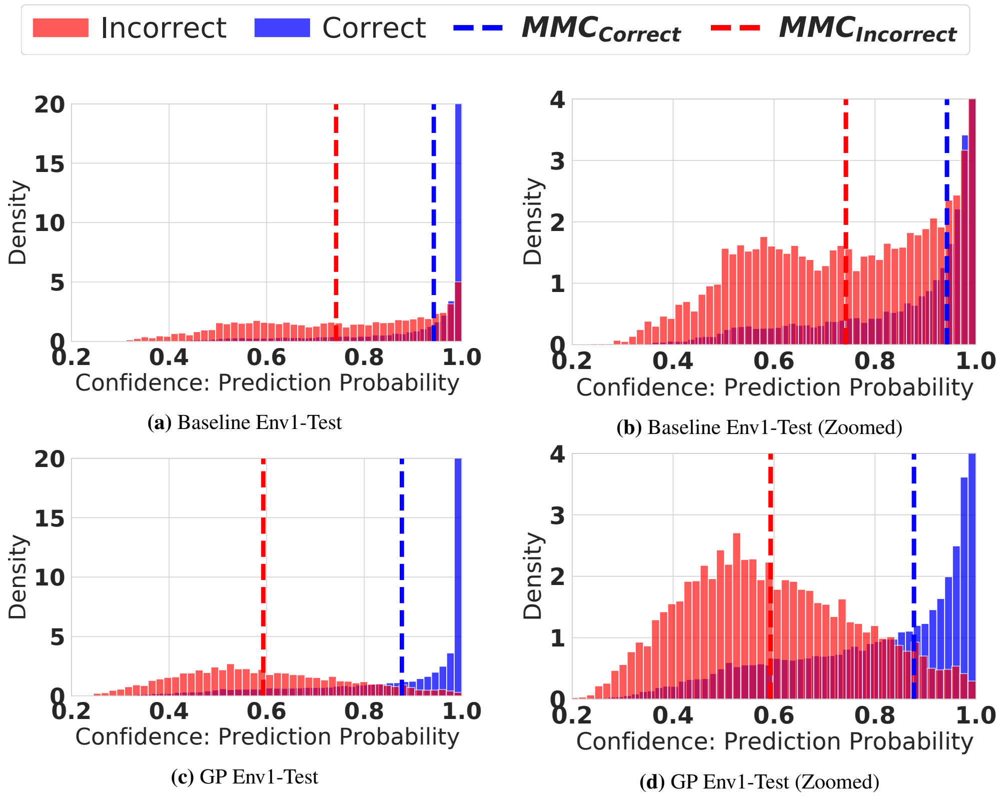
The dashed vertical lines in the figure show the mean-maximal-confidence (MMC) of the respective distributions, and comparing these it becomes clear that GP significantly reduces the confidences of most negative samples. Even though post-hoc uncertainty calibration does not mitigate the problem, it still provides the first steps in the direction, and significantly improves the separation between confident correct and over-confident incorrect predictions. We also can observe that in addition to lowering the confidences of the negative samples, GP also marginally reduces the confidences of the correct samples (i.e. slight left shift in the blue line), but this is not unexpected. Being correct does not necessarily correlate with 100% confidence, despite the labels indicating only a binary presence or absence of the object. It is important to note, that the labels are mere approximations of the object class but do not reflect the true ambiguity (labels indicate presence of object regardless of the amount of noise present or difficulty in recognizing the objects). Given that measurement noise can have different severities and DL classifiers could still recognize the objects with lower SNR, the confidence for such correct predictions should still be lower than 100% as there is higher ambiguity.

10.5. Uncertainty calibration

In Fig. 10.2, we depict reliability diagrams for the two test datasets. These diagrams indicate how well the predicted confidences correspond to the true accuracy, i.e. how calibrated the uncertainty estimates are. A perfectly calibrated classifier yields the black dashed $y = x$ line; curves below reflect over-confidence, and curves above under-confidence. While the Baseline classifier outputs severely overconfident predictions, the post-hoc calibration methods significantly improve the calibration. Simple methods like TS help reduce the confidence, but further improvements can be made using more sophisticated methods such as GP and I-Max. However, even after uncertainty calibration, the predictions on Env2 are still overly confident. This shows that when encountered with realistic but unseen changes (e.g. a new environmental setup), that the predictive accuracy and calibration become worse and uncalibrated predictions make it harder to detect such situations where the classifier is unsure.

10.6. Quantitative evaluations of all metrics

In order to quantitatively show the benefit of uncertainty calibration methods, we examine the accuracy, $_{\text{top1}}$ECE, $_{\text{CW}}$ECE, and MMC$_{\text{inc}}$. These results, evaluated on the two test datasets (Env2 and Env3) are shown in Tab. 10.1. Additionally, we also evaluate the performance on an extra test set, Env1-Test. This test set comes from the exact same environment as the training dataset (e.g. Env1-Train), but involves different repetitions than the training dataset. Therefore, the exact spectra measurements in Env1-Test have not been seen during training but the distributions is very similar to Env1-Train.



(a) Baseline Env1-Test

(b) Baseline Env1-Test (Zoomed)

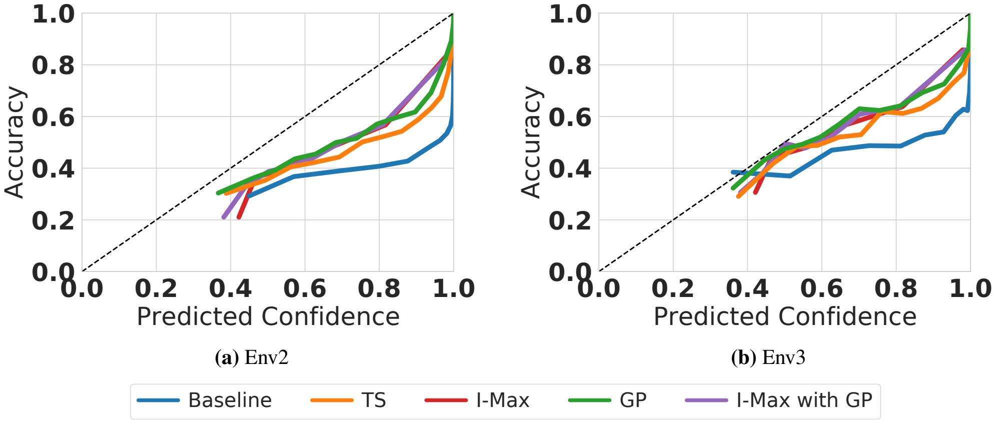
(c) GP Env1-Test

(d) GP Env1-Test (Zoomed)

Figure 10.1.: Confidence distribution of the top predicted class of Env2 by the (a) Baseline classifier, and after (c) post-hoc calibration (i.e. GP). Panels (b) and (d) are zoom-ins of panels (a) and (c), respectively. The distribution of the correctly classified samples are depicted in blue and the mis-classified samples in red. Even though the classifier correctly assigns high confidence for the correct samples (skewness of the blue distribution to the right), it also assigns high confidences for the mis-classified samples. GP uncertainty calibration remedies this by significantly lowering the confidence on mis-classified samples in the test set. The vertical dashed lines show the MMC of the respective distributions.

Firstly, it is noted that on all metrics, significantly better performance is achieved on Env1-Test than on Env2 and Env3. This is owed to the fact that the classifier can easily generalize to Env1-Test, as it shares significant similarities to the training set Env1-Train. However, it is important to note the impressive learning capabilities of the Baseline classifier to recognize most of the observations from Env2 and Env3. These include measurements from a novel scene, some novel object instances (e.g. unseen car model) and viewing angles. It should be noted that the exceptionally good performance on Env1-Test (even for the Baseline classifier) is not an excellent indicator of the generalization performance. Comparing the performance on Env1-Test and Env2/Env3, we notice that the classifier exhibits over-fitting and that only using Env1-Test for evaluation can over-estimate the classifiers ability to generalize to unseen data. Therefore, for future evaluations we do not consider Env1-Test for evaluation.

We observe that all uncertainty calibrations methods help improve on the uncertainty metrics compared to the Baseline. Even a simple scaling (TS) of all confidences (correct and incorrect)



(a) Env2

(b) Env3

Figure 10.2.: Reliability diagrams of the Baseline and uncertainty calibration methods. The baseline classifier confidences are severely over-confident for both test sets (larger distance to the diagonal line). The uncertainty calibration methods help to improve this by reducing the confidence. Among these, GP and I-Max show the best calibration performance. We find that I-Max and I-Max with GP perform similarly.

shows significant improvements. Larger gains are observed for I-Max and I-Max with GP, with the former offering slightly better performance. Interestingly, we observe that TS and GP both perform similarly across all metrics, with TS offering marginally better performance. Additionally, as GP involves multiple stochastic forward passes (preventing its use in real-time systems), TS and I-Max variants have large advantages in terms of test time, allowing easy integration into real-time systems with negligible extra computation time relative to the DL classifier. Overall, we find that all post-hoc calibration techniques help improve the calibration performance.

10.6.1. Spectra corruptions

In Fig. 10.3, we study the generalization of the classifiers to unseen dataset shifts. We synthetically corrupt the ROI spectra, presenting the classifier with input distributions unseen during training. More details of the spectra corruptions can be found in Sec. 4.5. We average the metrics for all corruption types across the 3 severities. In Fig. 10.3a, we highlight the effect of the corruptions on the classifier's confidence estimates. As the corruptions get more severe, the ECE increases. A main factor to this increase in calibration error is illustrated in Fig. 10.3b, which shows the MMC over the same corruption severities. The network fails to reduce its confidences despite becoming less accurate. In summary, we again observe the trend of over-confidence, and, as before, the uncertainty calibration methods can yield significant improvements relative to the Baseline.

10.6.2. Out-of-distribution spectra

To test even further the limits of the classifiers confidence to identify cases when it should be uncertain, we also evaluate on out-of-distribution data (i.e. objects never seen during training), for which it is impossible to classify correctly. The ideal prediction in this case should be a uniform prediction over all classes, however, the network still assigns very high confidences

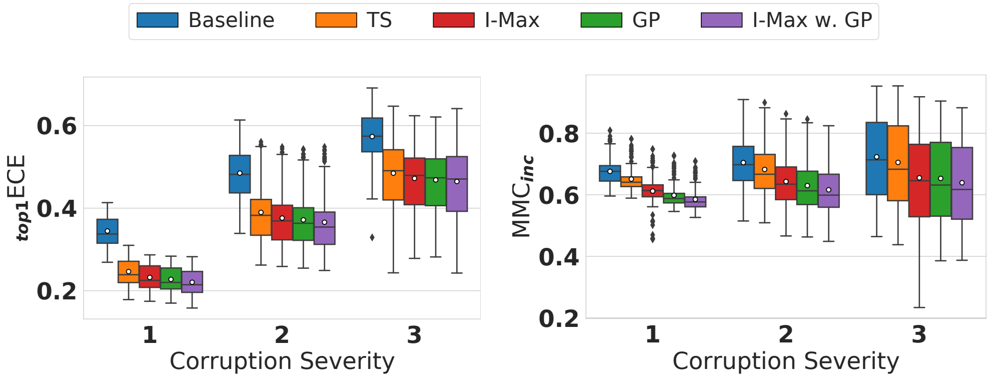
Table 10.1.: Comparison between the Baseline and post-hoc uncertainty calibration methods on Accuracy, ${}_{\text{top1}}$ECE, ${}_{\text{CW}}$ECEand MMC$_{\text{inc}}$for the test datasets Env2 and Env3. We additionally create and evaluate a special test set (i.e. Env1-Test) which comes from the same environment as the training dataset but is unseen during the training. Consistent with observations from Chap. 7, we observe that all post-hoc methods help improve on the uncertainty metrics compared to the baseline, whereas accuracy remains similar. GP offers the best over-all performance (with similar performance by I-Max), though it involves multiple stochastic forward passes and is not feasible on a real-time system. Alternatively, I-Max offers competitive uncertainty calibration improvements and is significantly faster; allowing easy integration into any real-time system. Given that Env1-Test shares many similaries to the training dataset (because both are measured from the exact same environment), the classifiers perform much better at this test set compared to Env2 and Env3. Showing that the classifiers can generalize better when for the same environment but is still able to obtain impressive performance on challenging unseen environments.

Method	Time (μs) ↓	Acc ↑	${}_{\text{top1}}$ECE ↓	${}_{\text{CW}}$ECE ↓	MMC$_{\text{inc}}$ ↓
			Env2		
Baseline	0.00 ± 0.00	**52.50** ± 0.33	0.227 ± 0.025	0.170 ± 0.017	0.666 ± 0.030
TS	0.75 ± 0.01	**52.50** ± 0.33	0.196 ± 0.012	0.147 ± 0.007	0.629 ± 0.014
GP	35.2 ± 1.01	**52.50** ± 0.32	0.198 ± 0.011	0.148 ± 0.006	0.632 ± 0.012
I-Max	0.76 ± 0.01	51.14 ± 0.78	**0.193** ± 0.014	**0.142** ± 0.010	**0.621** ± 0.045
I-Max w. GP	0.76 ± 0.01	52.05 ± 0.33	0.194 ± 0.012	0.144 ± 0.008	0.625 ± 0.015
			Env3		
Baseline	0.00 ± 0.00	57.05 ± 0.27	0.191 ± 0.018	0.143 ± 0.013	0.671 ± 0.022
TS	0.75 ± 0.01	57.05 ± 0.27	0.163 ± 0.004	0.123 ± 0.003	0.636 ± 0.004
GP	35.2 ± 1.01	**57.06** ± 0.27	0.164 ± 0.003	0.124 ± 0.002	0.639 ± 0.003
I-Max	0.76 ± 0.01	56.09 ± 0.35	**0.162** ± 0.014	**0.117** ± 0.012	**0.627** ± 0.052
I-Max w. GP	0.76 ± 0.01	57.05 ± 0.26	**0.162** ± 0.005	0.118 ± 0.004	0.633 ± 0.005
			Env1-Test		
Baseline	0.00 ± 0.00	**81.22** ± 0.38	0.037 ± 0.012	0.042 ± 0.006	0.742 ± 0.023
TS	0.75 ± 0.01	**81.22** ± 0.38	0.017 ± 0.002	0.031 ± 0.002	0.598 ± 0.005
GP	35.2 ± 1.01	**81.22** ± 0.38	0.018 ± 0.002	0.031 ± 0.002	0.600 ± 0.004
I-Max	0.76 ± 0.01	78.95 ± 0.88	**0.016** ± 0.031	**0.017** ± 0.006	**0.571** ± 0.102
I-Max w. GP	0.76 ± 0.01	81.16 ± 0.37	**0.016** ± 0.002	0.018 ± 0.003	0.593 ± 0.005

(Fig. 10.4). As rare unseen object types can commonly be encountered by a real-world system, assigning lower confidences to them is vital; a classifier should not confidently predict on what it does not know. However, this is far from the case in the tested experimental setup. In line with the previous results, post-hoc methods (GP) can significantly reduce these confidences. Even though the post-hoc calibration methods can correctly reduce the confidences of the OOD data, the outlier detection performance remains similar. A topic of future research is to directly optimize the outlier detection performance in order to yield reliable low confident predictions for cases where the classifiers are encountered with unknown objects.

10.7. Conclusion

As we observe rapidly growing interest in using deep learning for radar solutions, our intention is to study the behavior of radar classifiers beyond pure classification accuracy. Ultimately neural networks are optimized for high accuracy, but our results show that it is important to also critically investigate the associated prediction confidences. Through the use of measurements from an environment unfamiliar to the classifiers, we show that they fail to reflect higher uncertainties



(a) ${}_{\text{top1}}$ECE ↓ **(b)** MMC$_{\text{inc}}$

Figure 10.3.: Boxplot visualizations of (a) ${}_{\text{top1}}$ECEand (b) MMC$_{\text{inc}}$evaluated on corrupted spectra samples from Env2. All corruptions have been averaged out per severity. As expected, the ${}_{\text{top1}}$ECEbecomes worse with higher severity and for each case all post-hoc methods improve the calibration performance. This means that the uncertainty estimates associated for these corruption samples are more accurate than the Baseline network confidences.

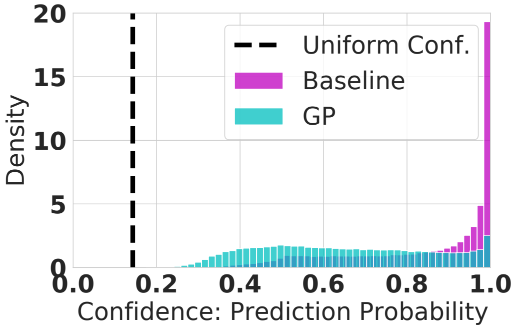


Figure 10.4.: Even for the OOD dataset (which contains only outlier objects) the classifier exhibits extreme overconfidence. The desired confidence is achieved with a uniform confidence across all classes (i.e. dashed vertical line).

to unseen but realistic real-world changes in the spectra. In an attempt to incrementally induce ambiguity to the spectra, a controlled set of corruptions presented to the classifiers, showed that confidences remained high, although predictions became highly inaccurate. Lastly, we mimic realistic real-world encounters by attempting to classify unknown objects. Clearly, no neural network can correctly classify these objects, but remarkably the confidence in the necessarily wrong predictions remained very high. Displaying that highly-accurate classifiers fail to reflect uncertainty in their predictions, shows that more work is required for DL-based radar solutions to address the challenges of real-world automotive applications. This should not diminish the fact that DL-based models are highly attractive because of their accuracy, and clearly learn important features, which are hard to model otherwise. We conclude that future work should not only address increasing accuracy, but should guide classifiers towards using their expressive powers for learning to model reliable uncertainty values.

As a first solution, we propose to use uncertainty calibration methods, and showed that these can be effective in improving confidence calibration across multiple setups, and reduce the severe overconfidence behavior to some extent. Next steps could address the simulation of even more realistic corruptions and augmentations to further improve the evaluation and support the development of highly-accurate and calibrated solutions.

Chapter 11.

P-smoothing: Improving uncertainty calibration of radar classifiers with soft labels

Contribution: Inspired by during-training calibration techniques discussed in Chap. 8, we aim to improve the uncertainty calibration of radar classifiers using a novel label smoothing technique to train with soft labels. As data augmentation of radar spectra remains an open problem, we focus our attention on adopting the use of soft-labels to improve the training process of radar classifiers. Unlike zero-entropy hard labels (i.e. labels which reflect zero uncertainty), which are commonly available during training of deep learning classifiers, using soft-labels can partially address the notorious over-confidence issues of deep learning classifiers. These soft-labels allow the reflection of the uncertainty inherent in a sample which can be leveraged during training to learn better calibrated networks. Unfortunately, obtaining soft-labels which accurately quantify the uncertainty of a sample can often be arduous, expensive or at times even impossible to obtain. In this work, we leverage radar specific knowledge to address the problem of creating soft labels for radar spectra and showcase how this key information can ameliorate the training process of deep learning radar classifiers by improving its uncertainty calibration. Treating the number and amplitudes of the reflected peaks in the spectra as one measure of the object information captured by the sensor, we exploit the prior knowledge that this information reduces at larger distances and that it differs for the various object types (e.g. small objects reflect fewer signals). We rely on the range of the object and the average measured power in the spectra as two indicators of the uncertainty inherent in the samples, and use these two indicators to define a sample dependant smoothing function to refine the hard labels. As a result of this soft information, the training process can differentiate spectra with few or low power reflections in the spectra (which are often harder to classify) and assign them with lower confidences, to induce uncertainty in its predictions. Though, at the same time maintaining the higher confidences for easy, noise-free, samples. We extensively showcase the performance gains of using these soft labels on uncertainty calibration, as well as multiple other tasks and metrics.

11.1. Introduction

Supervised learning has shown great performance in the presence of labeled data, however its developments mostly focus on improving the generalization accuracy instead of the robustness or reliability of the predictions. As a result, predictions from high capacity models, such as deep learning-based ones, tend to be highly accurate but poor representatives of the predictive uncertainty. Classifiers inhibiting such characteristics have limited use in practice, as decision-making systems fail to distinguish between incorrect over-confident predictions (i.e. which can be harmful) and correct high-confident predictions (i.e. easy samples which should be given high confidences).

Among multiple reasons behind these notorious over-confidence characteristics, one fundamental reason emanates from the dataset used during training. Almost all supervised datasets consist of hard one-hot label vectors (i.e. binary labels for each class) and often only have a single ground truth class label (i.e. hard probability vectors summing to 1). In contrast to the inaccessible ground truth distribution which does reflect the uncertainty inherent in the samples, these approximations (i.e. hard labels) induce an unwanted over-confidence bias.

In this chapter, we identify that the over-confidence which emanates from using hard labels can be fixed by using soft labels [Mueller et al., 2019; Hinton et al., 2015; Thulasidasan et al., 2019; Patel et al., 2020]. In addition to improving the uncertainty calibration, they have also been shown to improve the generalization performance [Szegedy et al., 2016; Hinton et al., 2015; Zhang et al., 2018]. Despite soft labels improving the over-all performance of deep learning classifiers, no work has exploited using soft labels for training radar spectra classifiers. In addition to being the first works to apply soft label training for radar object classifiers, we propose two novel techniques to compute sample-specific smoothing factors to refine the hard labels. These two techniques are designed to be proxy measures for the class ambiguity present in the samples. Even though this work focuses on radar spectra classification, the proposed technique to compute sample-specific smoothing factors for soft labels could potentially translate to other data modalities. The main requirement being the development of some measure which is able to quantify the rise in ambiguity in different situations.

11.1.1. Uncertainty reflected in hard vs. soft labels

Unlike hard labels, soft labels are continuous probability vectors more like the continuous ground truth distribution. In order to motivate the use of soft labels we present a small example. For the task of image classification of animals, leopards overwhelmingly share multiple features with jaguars. Even though both cats have their discriminative features (i.e. valid hard labels can be obtained and learned), under certain corruptions such as motion blur or occlusions, even human animal experts would reflect uncertainty in their classification. Soft labels similarly allow the reflection of the uncertainty by increasing the entropy of the label by distributing the mass of the ground truth label to other classes, rather than assigning all the mass to a single class (e.g. leopard with confidence 1.0).

Using this analogy, we can describe ambiguous situations in radar spectra classification even in the absence of corruptions or sensor malfunctions. According to the radar range equation [Skolnik, 2008], the power measured by the receiving antennas depends on the amount of transmitted power, the range, and reflecting characteristics of the objects. As the power transmitted to all objects in the field-of-view remains roughly uniform, the received power is some inaccessible complex function of the range and reflecting characteristics of the object. The received power is measured in the spectra from which deep learning classifiers attempt to approximate this complex function by finding features which lead to the accurate classification of the object. As the received power decreases and less class-specific information is available in the spectra, the object class becomes ambiguous; ultimately making it harder to classify as the classifiers rely on much less information to determine the classification. This ambiguity can increase at large distances where all objects reflect relatively fewer or lower power peaks than from closer distances. Another ambiguity factor relies on the ability of the object to reflect the signals back to the radar. In these cases, the classifiers have to rely on much less information in order to determine a classification, and the goal of this work is to ensure that the predictive uncertainty of the predictions is indicative of such situations.

In Fig. 11.1 we visualize ROI spectra samples from multiple objects (car, motorbike, construction barrier, pedestrian and stop sign) measured at different ranges (15m, 25m, and 35m) which show how both these effects control the amount of received power. Even though it has been shown that deep learning classifiers can correctly classify spectra with few small peaks, an autonomous decision-making system still relies on the predictive uncertainty to determine the reliability of the classification. The reliable quantification of this uncertainty becomes relevant in unfamiliar or unseen situations where the system needs to decide to use this information to execute planned actions or rather practice abstention. Currently, with pure hard labels the classifiers cannot learn the correct distribution which best reflects the predictive uncertainty.

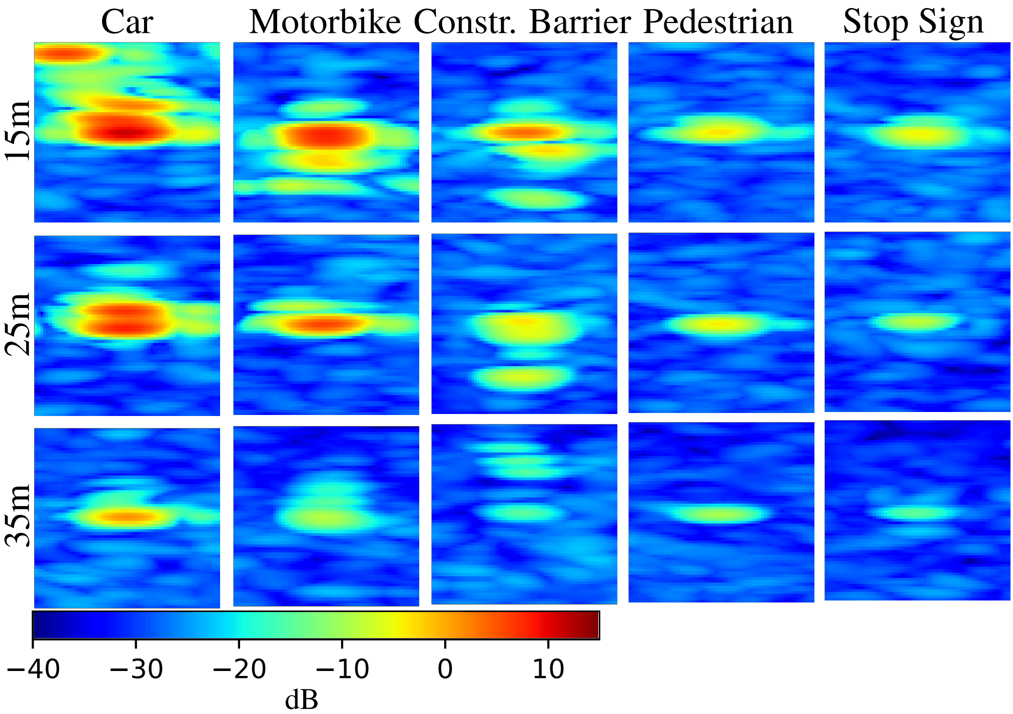


Figure 11.1.: Range-Azimuth spectra samples of 5 objects each measured at 3 different distances. We highlight two observations: (1) the number and amplitudes of the peaks reduce for each object when measured from farther away and (2) for a given range the total amount of power received differs across each object. As expected objects with large metallic components (e.g. car and motorbike) are much more reflective than pedestrians and stop signs which only have few small or no metallic components. The construction barrier, which is also large but not as reflective, also sends back significant signals though much less than the car and motorbikes.

11.1.2. Over-confidence with hard label training

During the early epochs of network training, hard labels quickly succeed in learning discriminative features needed to minimize the commonly used negative log likelihood (NLL) loss. As the training proceeds and increasingly more challenging discriminative features are learned, the loss and classification error (i.e. accuracy) continue to improve. Though, eventually at later phases of the training, the accuracy performance gains begin to saturate, whereas the training loss can continue to improve. At this stage, in addition to the networks exhibiting the issue of over-fitting, another problem is that the predictions become increasingly mis-calibrated due to the correct but zero-entropy hard label vectors [Thulasidasan et al., 2019]. This is referred to as NLL over-fitting in [Mukhoti et al., 2020]. At this point, further minimization of the NLL loss, even without the presence of over-fitting (i.e. validation set loss also continues to improve), forces the predictions to become overly confident. As a cross entropy measure between the two distributions, the NLL loss, as desired, minimizes the divergence between the predictive distribution and the ground truth distribution. Though, due to the lack of a reliable estimate of the ground truth *distribution*, the NLL is limited to the over confident hard labels. In order to obtain reliable and trustworthy predictions for unseen data at test time, merely minimizing the NLL with hard labels might not be enough as the trained networks fail to learn the quality of reflecting predictive uncertainty.

11.2. Soft labels in the radar domain

Obtaining soft-labels which accurately quantify the uncertainty of a sample can often be arduous (i.e. humans are not good at accurately quantifying uncertainty), expensive (i.e. obtaining soft labels by aggregating multiple annotator hard labels for large datasets is costly) or at times even impossible to obtain (i.e. human annotators are not always involved in the labeling process). Therefore, the dataset creation process predominantly focuses on diversifying and increasing the data samples, with little to no attention given to the associated (ground truth) label. As a result, these supervised learning algorithms mainly show continuous success in processing and extracting discriminating features from large complex datasets (i.e. improving the accuracy performance) but not its reliability.

Unlike in vision and most other domains, the input of the network of radar spectra classifiers have a different meaning. The input represents the amount of power/signal information reflected back from the objects and measured by the radar sensor, therefore larger amplitudes ("pixel values") represent more power reflected back. For smaller distant objects such as the stop sign at 35m, it is clearly visible in Fig 11.1 that there is only a single small received peak and that this peak is very close to the noise floor. For such samples, the radar classifiers are tasked with the challenge of providing a prediction based on this single relatively small peak. The smaller and fewer the peaks become (e.g. stop sign at even larger distances), the harder the classification tasks should become as a similar small peak can also be received from other small objects (e.g. pedestrian) or large objects at even larger distances (e.g. car at 80m).

The difficulty in predicting these samples can be observed in Fig. 11.2, where we evaluate the accuracy and calibration performance of the Baseline (i.e. trained with hard labels) and Ours (i.e. our best label smoothing method presented in Sec. 11.3) on the test set at varying distances. Additionally, we evaluate the test set corrupted with speckle noise at severity 1 (i.e. the dotted curves). We observe that overall the performance degrades at larger distances. More information on this experiment and a more through comparison of all methods and other metrics can be seen in Sec. 11.8.

Using this observation, we exploit the range and received power to refine the hard label of the spectra to better reflect the uncertainty or difficulty in predicting the spectra. One version of our label smoothing only uses the range information, and the other uses the mean received power in the spectra (i.e. mean in linear scale). We note that there are multiple ways to determine the uncertainty associated with a radar spectra, though we find that using these simple measures are sufficient to significantly improve the calibration performance of the classifiers. Using other measures instead of the range and mean received power are part of future works.

11.2.1. Exploiting range and received power information for label smoothing

We further motivate the design choices of using the range and received power for determining the amount of label smoothing. In Fig. 11.3, we plot the average power received of each region-of-interest (ROI) spectra and plot it against the range for multiple object classes for the entire dataset. Due to the loss of power with distance, it is seen that the received power greatly decreases when objects are farther away. Furthermore, large objects have significantly more reflections than smaller ones. We leverage this information as a proxy to the uncertainty of the spectra and use it to smooth the hard labels.

11.3. Methodology

In this section, we first formally motivate the need for training the classifiers with soft labels and then introduce the concept of label smoothing [Szegedy et al., 2016]. We then present two ways to improve the label smoothing called R-smoothing and P-smoothing.

11.3.1. Hard labels vs. soft labels

In a classification setting, the ground truth data distribution $P_*(x,y)$ is unknown in most practical situations, and as a result it is approximated by a limited training dataset, $D = \{x_i,y_i\}_{i=1}^N$ (i.e. empirical distribution $P_d(x,y) = 1/N \sum_{(i=1)}^N \delta(x = x_i,y = y_i)$). Basically, the empirical distribution is formed by assembling delta functions located on each example [Zhang et al., 2018]. This density estimate can be further improved by replacing the delta function with some estimate of the density in the vicinity of the training points (vicinal distribution [Chapelle et al., 2001]). Using expert domain knowledge, the vicinity or neighborhood around each example in the training data can be defined, allowing additional virtual samples and labels to be drawn. This process is also known as data augmentation (known to boost generalization performance), which creates points by transforming training inputs x, though it often leaves the target outputs y unchanged. Due to the delta functions in the dataset creation process, the target labels y, for a C-class classification problem, are hard (i.e. one-hot encodings - $y \in \{y' : y' \in \{0,1\}^C, 1^T y' = 1\}$) estimates of the true conditional distribution $P_*(y|x)$ (also known as the predictive uncertainty). On the other hand soft labels have values between 0 and 1 (i.e. $y \in \{y' : y' \in [0,1]^C, 1^T y' = 1\}$)). These labels can better reflect the true confidences (or uncertainties) for each class. For example, a highly ambiguous sample from a 3-class classification problem could have a soft label of

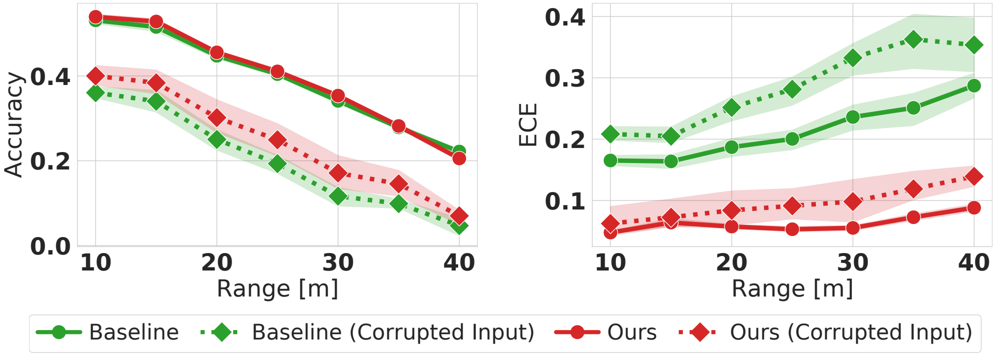


Figure 11.2.: Accuracy and calibration evaluation of the Baseline and our soft-label classifier ("Ours") on the test set and the corrupted test set (speckle noise at severity 1) at varying ranges. The performance significantly degrades at larger distances with better performance seen by our label smoothing method. More information can be found in Sec. 11.8.

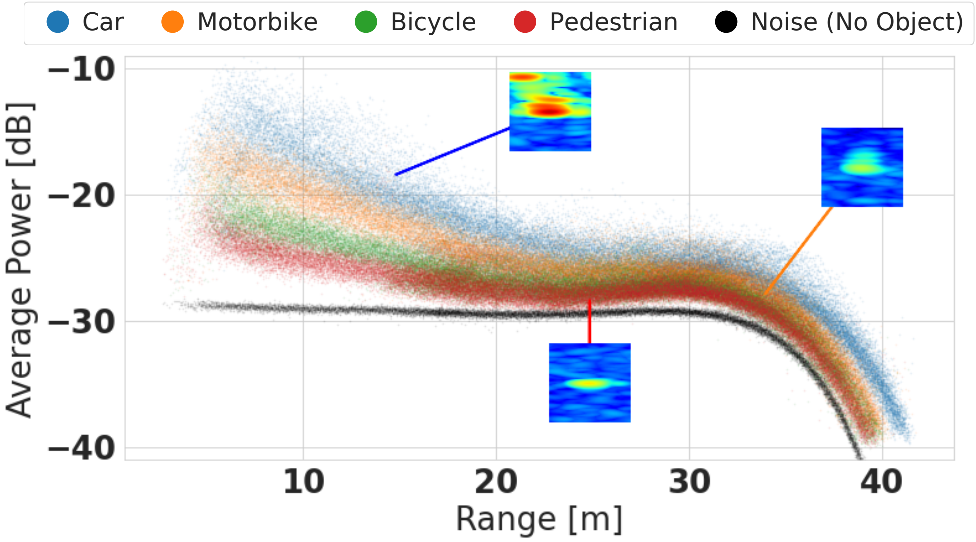


Figure 11.3.: We visualize the average received power of all ROI spectra in the dataset and plot it against the ground truth range. Similar to observations in Fig. 11.1, we observe the received power degrades over range and that overall some objects reflect more power than others. We also plot the average power of the Noise, which are ROI spectra where no objects were present.

$y = [0.5, 0.3, 0.2]$ which better reflects the confidences of each class. Whereas, the same highly ambiguous sample actually has the hard label of $y = [1, 0, 0]$.

The hard labels have an adverse effect, as the training data does not capture the true uncertainty of the target classes. This results in over-confident predictions (as shown and discussed in Part II) as the classifiers are trained solely on zero-entropy label signals; penalizing the network for reflecting any sign of uncertainty during training. These predictions tend to be highly mis-calibrated (i.e. the accuracy and uncertainty of the predictions are not well correlated; see Chap. 6) as the NLL loss used with hard labels tend to push towards zero-entropy predictions, regardless of their correctness. Alternatively, using softer labels would discourage over-confidence for uncertain situations which ultimately pose greater difficulty in correctly predicting them. They also have the benefit of producing better calibrated predictions as achieving optimal calibration highly relies on better estimation of the ambiguity between classes.

One problem which remains is the acquisition of these soft labels. One approach can be to ask annotators to assign confidences for every label, though this is difficult as humans are not good at quantifying uncertainty. Alternatively, aggregating hard labels from multiple annotators for each sample is another way, though it can become very expensive and impractical. Specifically for radar-based spectra classifiers, where the labeling is automated and spectra samples are not recognizable by humans, the acquisition of *accurate* soft labels is exceptionally hard. Consequently, we remain with a supervised learning dataset comprising of hard labels which lack any information about the data uncertainty. More formally, the models are trained with a data set $(x,y) \sim P_{\text{data}}$, where the target labels y are sampled from $P_{\text{data}}(y|x)$ (i.e. $\text{argmax}_y P_*(y|x)$), where P_* is the ground truth distribution approximated by P_{data} using training data. We present a new manner in which we tackle the problem of soft label estimation: estimating these soft labels directly from the data and other meta-data (e.g. range).

11.3.2. Connection to OMADA

In Chap. 8, we presented a solution (i.e. OMADA) to diversify the training dataset through sampling both the input x and y. In that work, we focused on using the powers of generative modeling to approximate the data manifold and learned the decision boundary regions on the manifold. We used a generator to synthesize visually ambiguous samples from these decision boundary regions, as well as obtained a soft label for each ambiguous sample. In a similar fashion, a generative model can be used for generating ambiguous spectra and soft labels, though we keep this as future work as generative modeling for radar spectra is still an open problem. We, instead, focus our attention on the ambiguity already inherent in the radar spectra and present ways to better reflect this through soft labels estimation.

11.3.3. Label smoothing

In [Szegedy et al., 2016], the authors introduced the concept of label smoothing, where they use a fixed value to smooth *all* hard labels *equally* (called ϵ-smoothing). This regularized the training procedure and partially addressed the issue of over-confidence, as classifiers trained with softer labels exhibit less over-confidence; even for smoothing values as small as $0.01 - 0.1$. It achieved this by simply lowering the confidences of *all* samples, regardless of the ease or difficulty in predicting the samples. So even though it might not learn a over-confident classifier anymore, it did not entirely and consistently solve the problem of calibration (see Chap. 8, Fig. 8.3).

Label smoothing can be seen as a mixture between the hard one-hot labels y and the class prior distribution over labels υ, with weights $1 - \epsilon$ and ϵ, respectively. More formally, the label y is smoothed to

$$\tilde{y} = (1 - \epsilon)y + \epsilon\upsilon, \tag{11.1}$$

where the value of ϵ determines the amount of smoothing. Compared to assigning ϵ to a small *fixed* value (with best validation set performance using $\epsilon = 0.1$ and also used in Szegedy et al. [2016] and Chap. 8), we propose to use an adaptive ϵ for each sample for further calibration performance gains.

11.3.4. Estimating soft labels for radar spectra: R-smoothing and P-smoothing

Building on the label smoothing technique, we propose **R-smoothing**, which uses the range R of the object to determine the value of ϵ,

$$\epsilon_R = 1 - \mathrm{e}^{-\alpha \frac{R - r_{\text{min}}}{r_{\text{max}} - r_{\text{min}}}}, \tag{11.2}$$

where $\alpha > 0$ controls the smoothing factor, r_{min} and r_{max} are the minimum and maximum range values for spectra samples in the training dataset and are used to normalize the range to $[0{,}1]$.

We note that this does not consider any input specific information or leverage any object specific features. As it only uses the range to determine the amount of label smoothing, it is agnostic to the object class. In order to incorporate spectra specific information to guide the amount of label smoothing, we propose another technique which utilizes the average power measured in the spectra. Using the average power of the ROI has been done before in [Visentin, 2019], where the measure was used to filter out all noisy samples, below some pre-defined threshold, before learning a classifier only for the resulting samples. Instead of using this measure as a filter to remove (noisy) samples, we propose to use it to determine the degree to which the hard labels should be smoothed. This allows the network to still learn from these noisy samples, though the confidences for these samples are encouraged to be significantly lower.

In order to consider the spectra during the label smoothing process, we additionally propose **P-smoothing**, which uses the average received power ($\pi(x) = \frac{1}{P}\sum_{p=1}^{P} x_p$ for the linear scale input x with P pixels) to determine the smoothing factor ϵ,

$$\epsilon_\pi = 1 - \mathrm{e}^{-\alpha\left(1 - \frac{\pi(x) - \pi_{\text{min}}}{\pi_{\text{max}} - \pi_{\text{min}}}\right)}, \tag{11.3}$$

where $\alpha > 0$ controls the smoothing factor, π_{min} and π_{max} are the minimum and maximum average power for spectra samples in the training dataset. The spectra of highly reflective large objects will have more received power, which ultimately result in small smoothing values ϵ_π (i.e. lower uncertainty reflected in the soft labels), and alternatively objects with weaker signal returns will result in large amounts of smoothing (i.e. higher uncertainty reflected in the soft labels). The distribution of the average power measure across range can be seen in Fig. 11.3.
In order to ensure that the soft labels still assigns more than 50% to the ground truth class, we bound the hyper-parameter α by $0 < \alpha < -\log 0.5$. This ensures that the training procedure with the soft labels, still uses labels which assign the majority of the mass to the correct ground truth class (according to the hard labels).

11.4. Experimental setup

We show the efficacy of the two methods R-smoothing and P-smoothing on two challenging radar test datasets introduced in Chap. 4.

11.4.1. Deep neural network training

We use the same architecture and training procedures as described in Chap. 5.2. We train the classifiers only using Env1-Train and select the model with the best accuracy on the independent validation set Env1-Valid. We evaluate the performance of the classifiers on two unseen test datasets: Env2-Test and Env3-Test. We train 10 independent classifiers with different initializations and report their mean and standard deviations for all experiments.

11.4.2. Tasks

While the experimental investigation is primarily focused on uncertainty calibration, we also evaluate and compare the performance on multiple other metrics and tasks. In this section we present an overview of all experiments and their details. All experiments can be broken into the following tasks:

1. **Uncertainty calibration** (Sec. 11.5): The calibration evaluation follows the evaluation discussed in Chap. 6. As a reminder, calibration measures the mis-match between the confidence and accuracy of the classifiers predictions. For a qualitative comparison we visualize the reliability diagrams.

2. **Quantitative evaluations of all metrics** (Sec. 11.6): Extensive comparison of all methods for the following metrics: Accuracy, $_{\mathrm{top1}}$ECE, $_{\mathrm{CW}}$ECE, mean-maximal-confidence of incorrect samples (MMC$_{\mathrm{inc}}$), negative log-likelihood (NLL), Brier score (mean-squared-error of predictions and labels).

3. **Ablation on smoothing hyper-parameter** α (Sec. 11.7): Ablation of α for R-smoothing and P-smoothing, as well as the fixed ϵ for ϵ-smoothing, and comparison against the Baseline (with $\alpha = \epsilon = 0.0$) on multiple metrics

4. **Study of the performance over object range** (Sec. 11.8): Study of the effect of the label smoothing methods on multiple metrics at varying object ranges.

5. **Spectra corruptions** (Sec. 11.9): Evaluation of the performance of the classifiers under dataset shifts by studying their behavior after synthetically corrupting the spectra under various corruptions.

6. **Thresholded accuracy** (Sec. 11.10): This metric computes a set of accuracy measures which are evaluated by filtering out all samples with predicted confidence above a set of thresholds. Classifiers which output high-quality uncertainty estimates assign higher confidences for correctly classified samples than incorrectly classified samples, and ideally uniform predictions for the incorrect samples. This measure quantifies how well a classifier is able to achieve this.

7. **Predicted entropy of trained classifiers** (Sec. 11.11): Comparison of the predicted entropy of the Baseline, ϵ-smoothing and P-smoothing. Visualization of a scatter plot which shows how the high-complexity models are able to closely reproduce the labels. This motivates the need for paying special attention to the label assigned to the sample to ensure a calibrated classifier is learned.

8. **Label smoothing + post-hoc calibration** (Sec. 11.12): Combination of the label smoothing methods presented in this chapter with post-hoc calibration methods presented in Chap. 7.

9. **Label smoothing + spectra decay** (Sec. 11.13): Combination of the label smoothing methods with the methods presented in Chap. 5 (namely, distance-to-center (DTC) map and spectra decay).

11.4.3. Note on evaluation labels

We note that even though the training process utilizes the soft labels, they are not used for any evaluations. Despite the soft labels improving classifier trainings, they cannot provide a fair evaluation setup. In order to keep the evaluations fair for all methods, we only use the soft labels for training the label smoothing classifiers and use hard labels for all evaluations.

11.5. Uncertainty calibration

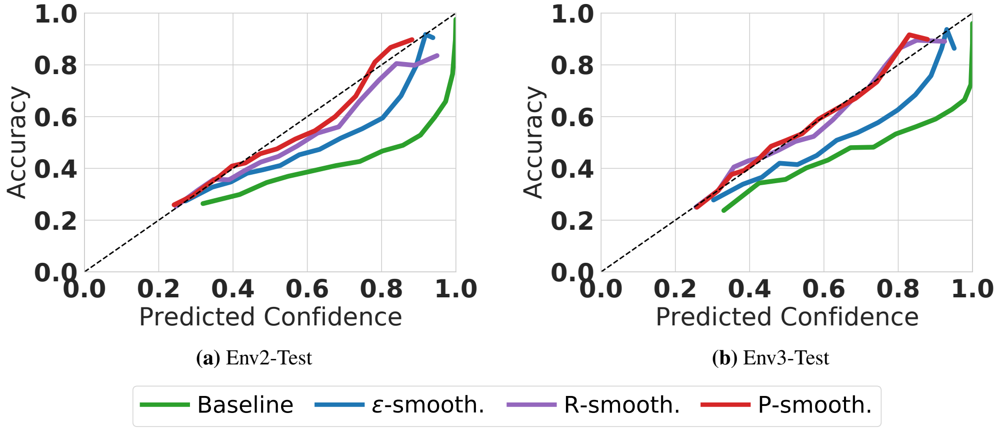
In Fig. 11.4, we plot the reliability diagrams for the training methods evaluated on the two test sets. Reliability diagrams have been explained in more detail in Chap. 6. In short, a classifier has perfect calibration when its predictive confidences match the accuracy (i.e. the black dashed diagonal line), and is over-confident (under-confident) when below (above) this line. The significant over-confidence of the Baseline can be seen by skewness of the curve to the right and the largest distance to the ideal calibration curve. We observe that all label smoothing methods improve the calibration performance and that R-smoothing and P-smoothing show the best performance (significantly lower distance to the diagonal dashed line). As this is merely a qualitative evaluation to visually compare the calibration performance, in the next section we provide more quantitative ECE evaluations.

11.6. Quantitative evaluations of all metrics

In Tab. 11.1 we quantitatively evaluate the calibration performance and multiple other metrics on the two test sets, Env2-Test and Env3-Test. We observe that all label smoothing methods improve the Baseline across all metrics, with the best performance seen with R-smoothing and P-smoothing. Even though the soft labels used by R-/P-smoothing were designed to improve the calibration performance and address the issues of over-confidence, we also observe performance gains on other metrics such as accuracy, NLL and Brier score. In addition to significantly improving the calibration performance, the classifiers are also less over-confident on the incorrectly classified samples (i.e. lower MMC_{inc}values). It is interesting to see that addressing the mis-calibration and over-confidence issues, of the Baseline classifier, result in significant performance gains at NLL (i.e. the loss used during training), Brier score (i.e. mean-squared-error to the hard labels) and marginal performance gains at the accuracy. This shows that the classifiers generalize better when using the modified soft labels instead of the ground truth hard labels.

11.7. Ablation on smoothing hyper-parameter α

In Fig. 11.5, we study the effect of the hyper-parameter α of R-/P-smoothing and ϵ of ϵ-smoothing and compare them against the Baseline (i.e. $\alpha = \epsilon = 0.0$). We observe that increasing α improves the $_{\text{top1}}$ECEand NLL performance for R-/P-smoothing, but only observe performance gains for ϵ-smoothing for $\epsilon \leq 0.2$. The classifiers perform worse for ϵ-smoothing for larger ϵ because they



(a) Env2-Test

(b) Env3-Test

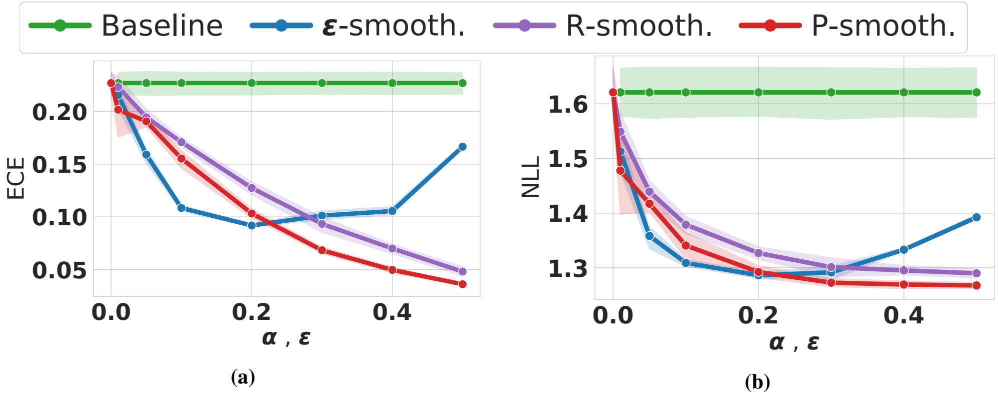
Figure 11.4.: Reliability diagrams of the Baseline and label smoothing methods for (a) Env2-Test and (b) Env3-Test. The baseline network confidences are severely over-confident for both test sets (larger distance to the diagonal line and skewness to the right). We observe that using soft labels instead of the hard labels used by the Baseline, greatly improves the confidence calibration. Among these, overall P-smoothing shows the best calibration performance.

Table 11.1.: Quantitative evaluation of the Baseline and label smoothing methods on Accuracy, $_{\text{top1}}$ECE, $_{\text{CW}}$ECE, MMC$_{\text{inc}}$, NLL and Brier score. Overall, R-smoothing and P-smoothing consistently show the best performance compared to the Baseline and ϵ-smoothing. Even though the R-/P-smoothing was designed to improve the calibration, we observe marginal gains on accuracy, as well as significant improvements at NLL (loss used during training) and Brier (mean squared error to the hard labels). Number report mean and standard deviation across 10 independent runs.

Method	Acc ↑	$_{\text{top1}}$ECE ↓	$_{\text{CW}}$ECE ↓	MMC$_{\text{inc}}$ ↓	NLL ↓	Brier ↓
			Env2-Test			
Baseline	52.50 ± 0.32	0.227 ± 0.03	0.170 ± 0.02	0.666 ± 0.03	1.621 ± 0.11	0.675 ± 0.02
ϵ-smooth.	53.05 ± 0.34	0.108 ± 0.01	0.111 ± 0.00	0.553 ± 0.01	1.309 ± 0.02	0.621 ± 0.01
R-smooth.	53.15 ± 0.48	0.048 ± 0.01	0.100 ± 0.01	0.486 ± 0.01	1.290 ± 0.02	0.608 ± 0.01
P-smooth.	**53.50** ± 0.61	**0.036** ± 0.00	**0.089** ± 0.00	**0.452** ± 0.01	**1.268** ± 0.01	**0.595** ± 0.01
			Env3-Test			
Baseline	57.05 ± 0.27	0.191 ± 0.02	0.143 ± 0.01	0.671 ± 0.02	1.572 ± 0.09	0.612 ± 0.01
ϵ-smooth.	57.35 ± 0.21	0.102 ± 0.01	0.095 ± 0.00	0.579 ± 0.01	1.183 ± 0.01	0.570 ± 0.00
R-smooth.	**57.78** ± 0.36	**0.033** ± 0.00	**0.087** ± 0.00	0.478 ± 0.01	**1.173** ± 0.01	**0.554** ± 0.00
P-smooth.	57.40 ± 0.40	**0.033** ± 0.00	0.088 ± 0.00	**0.467** ± 0.01	1.174 ± 0.01	**0.554** ± 0.00

begin to become under-confident. Even though naively reducing the confidence of all predictions is one solution to tackle the problem of over-confidence, this simple solution fails to output calibrated predictions. This shows that rather than simply uniformly smoothing all hard labels, a smarter adaptive choice for ϵ is needed.

We also see the advantage of using a label smoothing technique which considers the spectra input and not only the range input. P-smoothing performs better than R-smoothing, as P-smoothing determines the amount of smoothness based on the input spectra whereas R-smoothing only considers the object range. As can be seen in Fig. 11.3, using the average received power (used by P-smoothing) considers both the object type (seen by the partially separated object classes) and implicitly the range of the object (seen by the downward trend of the scatter points with larger range).



(a)

(b)

Figure 11.5.: Ablation of α for R-smoothing and P-smoothing and ϵ for ϵ-smoothing, compared to the Baseline trained with hard labels (i.e. $\alpha = \epsilon = 0.0$) evaluated for Env2-Test. We compare the (a) $_{\text{top1}}$ECEand (b) NLL and observe that for all $\alpha > 0$, R-/P-smoothing performs significantly better than the Baseline with larger α showing better performance. For ϵ-smoothing, we observe the best performance with $\epsilon \leq 0.2$, but the performance quickly degrades for larger ϵ. Unlike R-/P-smoothing, ϵ-smoothing cannot handle larger ϵ because the classifier becomes severely under-confident which is (similar to over-confident classifiers) not good for calibration. This shows the benefit of using an adaptive smoothing parameter which will only smooth the labels in cases where the class ambiguity is high and not all labels like ϵ-smoothing. P-smoothing performs better than R-smoothing, as P-smoothing determines the amount of smoothness based on the input spectra whereas R-smoothing only considers the object range.

11.8. Study of the performance over object range

Both R- and P-smoothing were designed to improve the performance of the classifiers at larger range by lowering the confidences of the labels for these spectra. We now study if this goal has been achieved. In order to study the influence of the training methods over range, we create subsets of the data at different range values and evaluate their performance. We create these subsets using 5m thresholds from 10m up until 40m and plot the evaluations of the different metrics in Fig. 11.6. For example, the point 20m represents all samples which fall into the subset $15 - 20$m.

We find that using the smoothing methods which considers the uncertainty associated with the spectra to smoothen the hard labels, provide better calibrated outputs, especially at larger distances. Unlike the Baseline and ϵ-smoothing, which becomes significantly worse at $_{\text{top1}}$ECEat larger distances, the calibration performance of R-/P-smoothing stay roughly constant at all ranges. As depicted in Fig. 11.2, the accuracy of all classifiers become significantly worse at larger ranges, therefore it becomes important to lower the confidence of these samples but at the same time maintain high accuracies for closer samples. In addition to the calibration performance gains, we observe that all label smoothing methods perform similarly at NLL and Brier score at larger distances, with R-/P-smoothing performing best at distances larger than 30m. This result shows that the design choice of smoothing the spectra of farther objects and low-power spectra more significantly helps maintain calibrated predictions for uncertain points which are predominantly found at larger ranges.

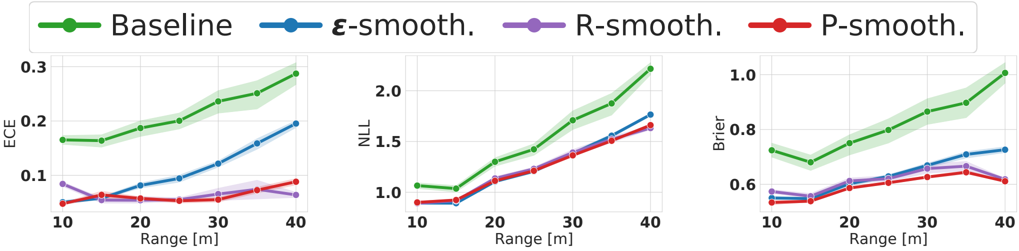


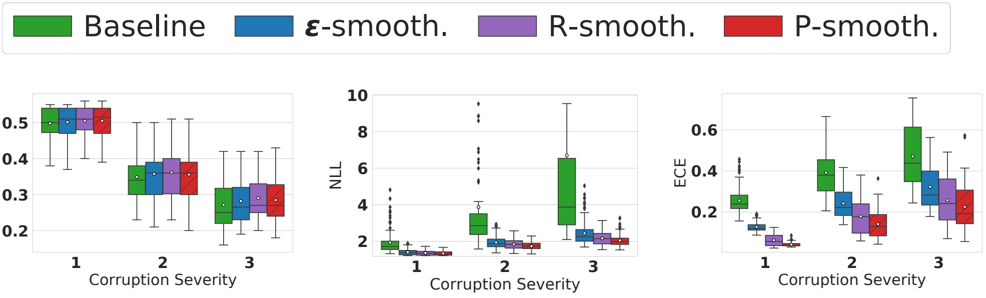
Figure 11.6.: Performance over range of all training methods. Creating subsets by grouping objects at similar distances to the sensor, we evaluate the (a) ECE, (b) NLL and (c) Brier Score performance. As expected, we observe that R-smoothing and P-smoothing have the best calibration performance, especially at larger distances. Additionally, we observe that for NLL and Brier, all label smoothing methods perform similarly, though the gains are better for R- and P-smoothing at large distances.

11.9. Spectra corruptions

In Fig. 11.7, we study the generalization of the classifiers to unseen dataset shifts. We synthetically corrupt the ROI spectra, presenting the classifier with input distributions unseen during training. More details of the spectra corruptions can be found in Sec. 4.5. We average the metrics for all corruption types across the 3 severities. Similar to previous observations of ECE on the test set, using P-smoothing also gives the best calibration performance under the corruption settings. We also observe that the accuracy performance is similar for all methods, though we observe significant improvements in the NLL performance of all label smoothing methods. Compared to the accuracy (which only cares whether a sample is correct or not), the NLL is a continuous measure which also cares about the over-confidence and this metric severely penalizes wrong predictions with high confidence. Overall, P-smoothing performs better and has lower variances at higher severities in terms of NLL. The high number of outliers (i.e. diamond markers) for the Baseline classifier for NLL, come from predicting significantly high confidences for the mis-classified samples. With increasing severity, the accuracy performance greatly degrades but the Baseline continues to be over-confident and is greatly penalized by the NLL loss. Using soft labels greatly helps with getting rid of the over-confidence on mis-classified samples (a problem which becomes significantly worse under dataset distribution shifts). Even using ϵ-smoothing can already greatly improve the over-confidence and mis-calibration of the Baseline classifier, though much larger performance gains can be achieved when considering a smarter technique to determine the amount of smoothness for each sample, like P-smoothing.

11.10. Threshold accuracy

For some safety-critical applications, a decision-making system could choose to ignore all predictions which the classifier is unsure about (e.g. abstaining from predicting these samples).



(a) (b) (c)

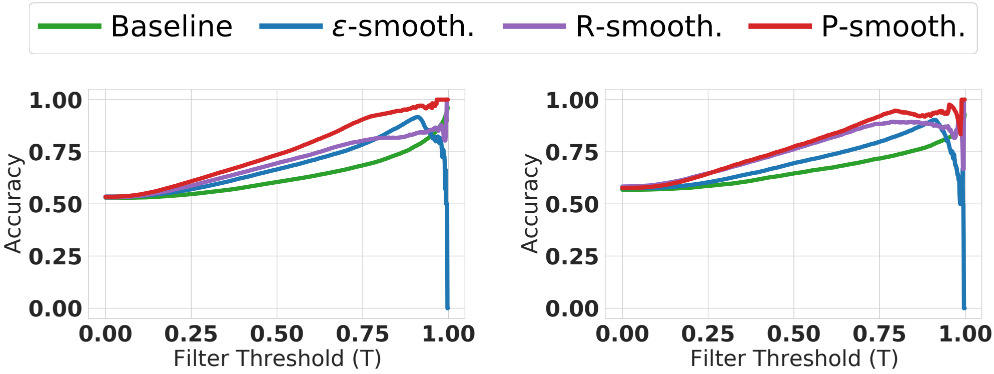
Figure 11.7.: Boxplot visualizations of (a) Accuracy, (b) NLL and (c) $_{\text{top1}}$ECEevaluated on corrupted spectra samples from Env2-Test. We observe that R-smoothing and P-smoothing both perform well across all metrics and corruption severities. All corruptions have been averaged out per severity. Hatched boxes indicate the best performing method per severity.

This is typically done by ignoring predictions with a predicted confidence lower than some threshold, where the threshold is controlled by the practitioner based on the application and the desired coverage. Threshold accuracy curves compare the accuracy performance on a set of thresholds, where better performance is obtained by classifiers which assign higher confidences for correctly classified samples than incorrectly classified ones. These curves measure how the accuracy would change if you filter out the samples assigned with low confidences (i.e. the samples for which the classifier was unsure and assigned low confidences). Ultimately, this measure compares the ability of the classifiers to rank correctly classified samples with higher confidences than incorrectly classified samples.

In Fig. 11.8, we plot the threshold accuracy curves for the different training methods evaluated on Env2-Test and Env3-Test. We observe that with larger more aggressive thresholding, the accuracy of P-smoothing remains higher than the rest. This shows that using P-smoothing to ignore low confidence samples would best maintain higher accuracies for the remaining samples. We also observe higher accuracies for R-smoothing, though at larger thresholds this becomes slightly worse than ϵ-smoothing, until the large drop of ϵ-smoothing. This shows that merely smoothing all labels based on the range, regardless of the power received in the spectra, can still lead to mis-classified samples assigned very high confidences. Poor confidence ranking performance can be observed for ϵ-smoothing with a large drop at the end of the curve. This occurs because incorrectly classified samples were assigned very high confidences, even higher than all other correctly classified samples.

11.11. Predicted entropy of trained classifiers

Previously, we have shown in Fig. 11.3 that the average received power (i.e. information) significantly reduces over range, as well as showed some example ROI samples (in Fig. 11.1) that much fewer signals are measured for objects at larger ranges. Additionally, in Fig. 11.6, we showed how these observations translate to weaker performance of all classifiers at larger range. Now, we qualitatively study the predictive uncertainties of the classifiers and how this changes over larger ranges, where the classifiers perform weaker. For samples at larger ranges, it is preferred to have high entropy predictions which indicate that the classifier has predicted



(a) Env2-Test **(b)** Env3-Test

Figure 11.8.: Threshold accuracy curves for all classifiers on (a) Env2-Test and (b) Env3-Test. These curves reflect the accuracy of the samples which remain after filtering out all samples with confidence smaller than the threshold T. We notice that as the threshold is increased (and more samples with low predictive confidences are removed), the accuracy of P-smoothing remains the highest among the training methods. This indicates that it is best at assigning correctly classified samples higher confidences than incorrectly classified ones. The large drop of ϵ-smoothing happens because it wrongly assigns the highest confidences to incorrectly classified samples.

an ambiguous sample. We note that even though there are samples at larger ranges which are correctly classified, the classifiers base their decision on less information and should thus reflect higher uncertainties for these samples.

In Fig. 11.9, we visualize a scatter plot of the predicted entropy for the Baseline (trained with hard labels), ϵ-smoothing and P-smoothing. In order to better observe the trend of the scatter points, an order-3 polynomial fitting the scatter points is depicted as curves. Additionally, we leave out the ϵ-smoothing scatter points as it makes the figure hard to interpret, though we keep the polynomial fit curve.

Firstly, the over-confidence issue of the Baseline is clearly visible in the plot. We notice that the predictive entropy very closely matches the hard labels (i.e. a zero-entropy vector) and produces predictions with entropy close to 0.0 (i.e. predicted confidences close to 1.0), regardless of the range of the objects and the level of ambiguity or noise in the spectra. Using ϵ-smoothing we observe a behavior similar to the Baseline but with larger entropy predictions (i.e. reflecting higher uncertainties). It is important to note that similar to the Baseline, ϵ-smoothing is also almost agnostic to the range and similarly ignores the uncertainty inherent in the data. This shows that ϵ-smoothing simply increases the entropy of *all* predictions roughly equally, even the nearby samples which could be assigned higher confidences.

Alternatively, P-smoothing learns a predictive entropy distributions which better correlate with the range and implicitly the uncertainty in classifying the inputs. It maintains lower entropies (i.e. higher confidences) for nearby objects for which the classifiers have more reflections and information in the spectra, and increasing the uncertainties for larger ranges where the classifiers are known to perform worse. Even though the entropy distribution can be controlled by the smoothing hyper-parameter α, the trend of having lower confidences for larger range objects remain consistent for all choices.

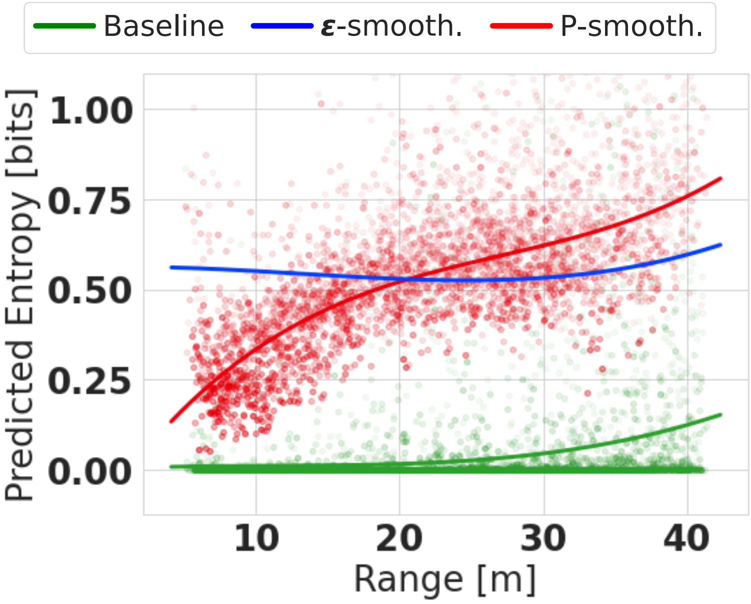


Figure 11.9.: Scatter plot of the predicted entropy over the object range for the Baseline, ϵ-smoothing and P-smoothing. The curve for each method depicts an order-3 polynomial fit to the scatter points to better see the trends of the different methods. In order to make the visualization less cluttered, we removed the scatter points of ϵ-smoothing because of the heavy overlap with P-smoothing. The predictive entropy distribution of the Baseline classifier is severely skewed towards zero which closely matches the entropy of the hard labels used during training. The predictive entropy distribution of ϵ-smoothing, which simply uniformly increases the entropy of the hard labels during training, closely matches that of the Baseline with a roughly fixed offset dependant on the amount of smoothing. We observe that the trend of P-smoothing seems to best correlate with the range, where larger range samples are assigned lower confidences. Even though there is no ground truth for the predictive entropy of the larger range samples, we showed in Fig. 11.6 how the performance of the classifiers get worse over range. Therefore, it is better for a classifier to assigned lower confidences to larger range objects than closer objects.

11.12. Label smoothing and post-hoc calibration

In Chap. 6, we introduced the concept of post-hoc and during-training calibration. The former technique learns to calibrate the classifiers after the training process is complete (e.g. I-Max (Chap. 7)), whereas the latter technique considers the object of calibration during training (e.g. OMADA (Chap. 8) and R-/P-smoothing from this chapter).

We now study the combination of the two types of calibration techniques by performing post-hoc calibration on classifiers trained in this chapter. In Tab. 11.2, we show the ECE performance of the 4 training procedures (i.e. Baseline, ϵ-smoothing, R-smoothing and P-smoothing) after applying post-hoc calibration. After the training of all classifiers, we apply the following post-hoc calibration techniques: Temperature scaling (TS [Guo et al., 2017]), Gaussian process (GP [Wenger et al., 2020]) and I-Max variants (Chap. 7). We note that all post-hoc calibration techniques for the label smoothing methods were learned using the same soft labels used during training. We found that the performance severely degraded when naively using the hard labels during post-hoc calibration (which is typically done) if soft labels were used during training.

We observe that the Baseline classifier benefits from all post-hoc calibration techniques, consistent with observations in Chap. 7. For the label smoothing methods, which are trained with soft labels to improve the calibration during training, post-hoc calibration is less effective. Among these we find that temperature scaling (TS) performs the worse and greatly degrades the calibration performance for classifiers which are well-calibrated (e.g. ϵ-, R-, P-smoothing), showing that simple uniform scaling of all predictions might only work for severely over-confident classifiers but are less effective for better calibrated classifiers. For ϵ-smoothing and R-smoothing only I-Max and I-Max with GP, respectively, improve the calibration performance, with the other techniques making the trained classifiers worse at calibration. For P-smoothing (the best label smoothing classifier), none of the post-hoc calibration techniques improve the performance. This result shows that post-hoc calibration has limited effects for classifiers which are already already well calibrated (e.g. P-smoothing), and further research is needed to exploit these kind of classifiers for further post-hoc performance gains.

Table 11.2.: Calibration performance on Env2-Test for combining label smoothing methods with post-hoc calibration from Chap. 7. After the training of all classifiers, we apply the following post-hoc calibration techniques: Temperature scaling (TS [Guo et al., 2017]), Gaussian process (GP [Wenger et al., 2020]) and I-Max variants (Chap. 7). For all 3 label smoothing methods, the post-hoc calibration was done using the soft labels used during training. We observe that for the Baseline classifier, all post-hoc techniques improves the performance. For ϵ-smoothing and R-smoothing only I-Max and I-Max with GP, respectively, improve the calibration performance, with the other techniques making the trained classifiers worse at calibration. For P-smoothing (the best classifier), none of the post-hoc calibration techniques improve the performance. Interestingly, we find that post-hoc calibration techniques have limited effect on classifiers which are already well calibrated (e.g. P-smoothing) but still observe the benefits when applying to the Baseline which is over-confident and mis-calibrated. The worst post-hoc calibration performance is seen by temperature scaling (TS) which is only helpful for the (over-confident and highly mis-calibrated) Baseline classifier but greately degrades the performance for classifiers with reasonable and better calibration performance (e.g. ϵ-, R-, P-smoothing). The best performing calibration methods are bold for each classifier (i.e. column block) and the overall best combination is underlined (i.e. P-smoothing with no post-hoc calibration).

Post-hoc	Baseline		ϵ-smooth.		R-smooth.		P-smooth.	
Calibration	${}_{\text{top1}}\text{ECE}\downarrow$	${}_{\text{CW}}\text{ECE}\downarrow$	${}_{\text{top1}}\text{ECE}\downarrow$	${}_{\text{CW}}\text{ECE}\downarrow$	${}_{\text{top1}}\text{ECE}\downarrow$	${}_{\text{CW}}\text{ECE}\downarrow$	${}_{\text{top1}}\text{ECE}\downarrow$	${}_{\text{CW}}\text{ECE}\downarrow$
No Post-hoc	0.227	0.170	0.108	0.111	0.048	0.100	$\underline{\mathbf{0.036}}$	$\underline{\mathbf{0.089}}$
TS	0.196	0.147	0.191	0.148	0.224	0.169	0.200	0.154
GP	0.198	0.148	0.122	0.118	0.051	0.101	0.071	0.096
I-Max	**0.193**	**0.142**	**0.104**	**0.102**	0.067	0.104	0.074	0.094
I-Max w. GP	0.194	0.144	0.110	0.109	**0.045**	**0.099**	0.068	0.092

11.13. Label smoothing and input spectra decay

Lastly, we apply the soft label techniques to methods designed to improve the generalization performance of radar classifiers. In Chap. 5, we introduced multiple new input augmentation techniques which improved the accuracy performance of the classifiers by considering radar specific information. We combine our label smoothing methods with the best performing input augmentation technique, namely the spectra decay. In Tab. 11.3, we compare the performance of the label smoothing methods without using spectra decay (i.e. Input: Default) and with using spectra decay (i.e. Input: Spectra Decay) on multiple metrics. Consistent with previous observations, we notice that P-smoothing is the best performing label smoothing variant even after using spectra decay. Furthermore, we observe that the best performance can be obtained by the combination of P-smoothing and using spectra decay during training. This shows that a calibration technique such as P-smoothing can be easily combined with generalization techniques to jointly optimize accuracy and ECE.

Table 11.3.: Combination of label smoothing methods with Spectra Decay (Chap. 5). Evaluation of the Accuracy, NLL, $_{\text{top1}}$ECEand $_{\text{CW}}$ECEperformance after applying a technique which improves the generalization performance of the radar classifiers by decaying the peripheral regions of the spectra. We observe that consistent with previous results, P-smoothing is the best label smoothing method across all metrics for both the classifiers trained with and without spectra decaying. We also notice significant generalization performance gains in terms of Accuracy and NLL, and small performance gains in terms of calibration with the combination of the two techniques. This shows that P-smoothing can be easily combined with other techniques to jointly optimize the generalization and calibration performance of the classifier. We obtain the best performance when combining P-smoothing with spectra decay. The best performing label smoothing method is underlined and the best overall combination is made bold.

Input	Method	Acc ↑	NLL ↓	$_{\text{top1}}$ECE ↓	$_{\text{CW}}$ECE ↓
Default	Baseline	52.4 ± 0.52	1.621 ± 0.11	0.227 ± 0.03	0.170 ± 0.02
Default	ϵ-smooth.	53.0 ± 0.47	1.309 ± 0.02	0.108 ± 0.01	0.111 ± 0.00
Default	R-smooth.	53.2 ± 0.63	1.290 ± 0.02	0.048 ± 0.01	0.100 ± 0.01
Default	P-smooth.	53.5 ± 0.71	1.268 ± 0.01	0.036 ± 0.00	0.089 ± 0.00
SpecDecay	Baseline	59.2 ± 0.42	1.415 ± 0.06	0.198 ± 0.02	0.156 ± 0.01
SpecDecay	ϵ-smooth.	59.2 ± 0.42	1.136 ± 0.02	0.076 ± 0.01	0.085 ± 0.00
SpecDecay	R-smooth.	59.2 ± 0.63	1.136 ± 0.01	0.037 ± 0.00	0.078 ± 0.00
SpecDecay	P-smooth.	**59.9** ± 0.57	**1.119** ± 0.01	**0.033** ± 0.00	**0.075** ± 0.00

11.14. Conclusion

In an attempt to improve the calibration performance of radar classifiers, we proposed to use a label smoothing technique to obtain soft labels for the training process. We developed two new label smoothing methods (R-smoothing and P-smoothing) which better correlate with the underlying ambiguity and uncertainty of the input spectra. The soft labels are based on the range and average received power of the spectra and both techniques have shown to greatly improve the predictive quality of the classifiers.

This was the first work aiming to improve the predictive uncertainty quality for radar classifiers during the training process (i.e. during-training calibration). We find that the presented label smoothing methods achieved this goal and greatly improve the calibration performance of the classifiers. In addition, as a side effect of addressing the mis-calibration and over-confidence issues of deep learning classifiers, we also observe consistent generalization performance gains. As a result, R-/P-smoothing both produced more reliable and accurate classifiers, enabling better integration of such deep learning classifiers into real-world systems which rely on the predictive uncertainties to perform actions.

This work has shown that deep learning classifiers, which have the power of learning complex features from data, highly benefit from radar specific knowledge and further improvements can be made by tailoring learning algorithms to leverage the radar-specific knowledge.

Even though the focus of this work was on *radar* classifiers, the core idea behind the presented label smoothing methods could potentially translate to other data modalities. The main requirement being the development of some measure which is able to quantify the rise in ambiguity in different situations. For radar we exploited the fact that classifiers rely on much less information for objects farther away, though this idea holds for vision and lidar classifiers as well. Lidar sensors, which are also able to measure the range, could directly apply R-smoothing to their training process. For vision, the number of pixels an object occupies in an image can be one possible proxy measure to the amount of information the classifier relies on. Alternatively, other image quality measures such as brightness, the amount of motion blur, compression-rates or the weather (detected from other sensors or directly from the images) could also be used to determine the amount of label smoothing for each sample.

Future research directions for improving the performance of the radar classifiers include intervening during the training epochs to improve the uncertainty and generalization performance by incorporating more radar specific knowledge. The classifiers tend to focus on learning easy discriminative features in the data and developing new ways to guide the model capacity to learn more complex and challenging features seems to be a promising direction. Additionally, similar to Chap. 8, we plan on addressing the problem of developing radar-specific data augmentations and using generative modeling to further improve the calibration performance of the classifiers.

In the context of the literature, this chapter offered a novel solution for addressing literature gap LG3 with a focus on the radar domain. Inspired by OMADA (Chap. 8), we address this literature gap by utilizing radar-specific information about the measure signal to estimate soft labels. This chapter is a third contribution from this thesis (in addition to Chap. 7 and 8) which aims at offering novel *real-time uncertainty calibration* (LG3). Collectively, the observations from these three chapters largely fill literature gap LG3 with general uncertainty solutions, as well as domain-specific solutions for radar classifiers.

Part IV.

Conclusion, outlook and appendix

Chapter 12.

Conclusion

It is widely known that radar sensors offer robust measurements of objects in the environment, but open questions remained about the ability of autonomous vehicles to classify the objects based on the measured signal. In this thesis we have aimed at answering this question (RQ1) by showing the competitive classification performance of deep learning models on a new challenging radar dataset which evaluates the generalization performance to unseen environments, driving patterns and object instances. We find that answering this question alone is not enough for the reliable operation of autonomous vehicles, which often encounter unknown situations where it is forced to extrapolate its learned knowledge, thus requires evaluation beyond generalization (e.g. robustness to spectra corruptions and outlier detection of unknown objects) (RQ2). Safely handling unknown situations requires highly-accurate predictive uncertainty estimates similar to those presented in this thesis in Part II (RQ3). For further generalization, robustness and uncertainty estimation performance, we also find that utilizing radar-specific knowledge to aid the data-driven learning algorithms (RQ4) is an essential step forward for further improvements for safe and robust perception using the radar sensor.

Next, in Sec. 12.1, we quantify the impact of this thesis for radar-based object classification, followed by a summary of the contributions (Sec. 12.2). We answer the research questions (RQs) posed in Chap. 2 in Sec. 12.3 and end with an outlook for future research directions (Sec. 12.4).

12.1. Quantitative impact of thesis

In order to quantify the impact of this thesis, we reproduce figures from the thesis for comparing the performance of the Baseline with the combination of the novel techniques presented in this thesis for radar classification (SpecDecay (Chap. 5) and P-smoothing (Chap. 11))). These figures put into context the contribution of this thesis for various situations which will often be encountered by automotive sensors in the real-world. We summarize the performance of these two radar classifiers under (1) realistic input changes (i.e. objects at increasing distances which makes the classification task harder) in Fig. 12.1, (2) synthetic input corruptions in Fig. 12.2, and (3) realistic inputs from unknown object classes in Fig. 12.3.

In addition to improving radar classification, the contributions of this thesis also show significant performance gains in the vision domain. For CIFAR 100 image classification, the combination of the uncertainty methods proposed in this thesis (OMADA (Chap. 8) and I-Max (Chap. 7)) improve classification accuracy of a WRN model from 69.26% to 74.45%, calibration ${}_{\text{top1}}$ECE from 9.38×10^{-2} to 0.97×10^{-2}, and outlier detection ${}_{\text{OOD}}$AUCfrom 73.87% to 86.12%.

Overall, we significantly improve the performance of object classifiers enabling both accurate and reliable classification using a robust sensor, able to operate under adverse weather and lighting conditions, as well as accurately quantify the predictive uncertainty associated with the object predictions.

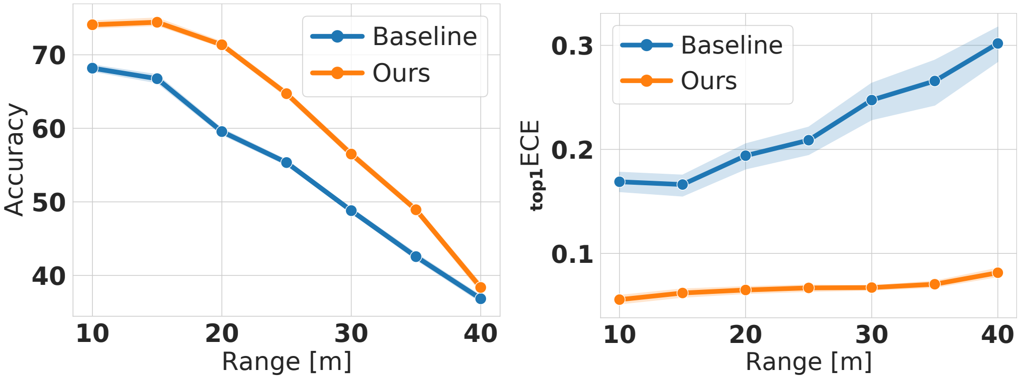


Figure 12.1.: Evaluation of (a) accuracy and (b) $_{\text{top1}}$ECEacross the object range for the Baseline (i.e. vanilla classifier training) and the best combination of methods proposed in this thesis (Ours). Overall, the combination of the methods proposed in this thesis (i.e. Ours) improve the accuracy from 52.40% to 59.20 and the $_{\text{top1}}$ECEfrom 0.227 to 0.033 on the Env2 test set. The accuracy remains higher for larger object distances when the signal strength greatly reduces, as well as reflects this loss of information through better calibrated predictions.

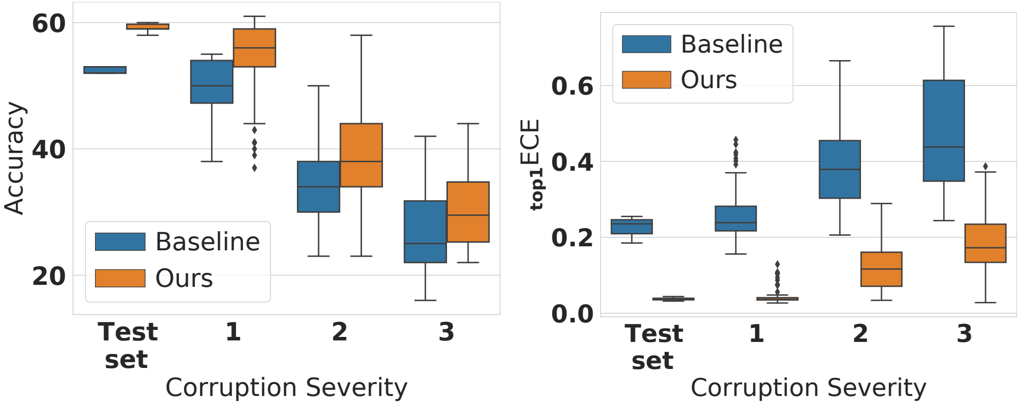


Figure 12.2.: Evaluation of (a) accuracy and (b) $_{\text{top1}}$ECEof the test set (i.e. no corruptions) and after input spectra corruptions with severities 1-3 for the Baseline (i.e. vanilla classifier training) and the best combination of methods proposed in this thesis (Ours). Even after severe input corruptions, which significantly reduce the classification accuracy, the proposed method consistently offers higher accuracies as well as significantly improves the uncertainty calibration of the predictions. The contributions of this thesis show significant robustness against incremental noise levels, introduced in the form of the seven corruptions studied in this thesis, through higher accuracy and calibration performance.

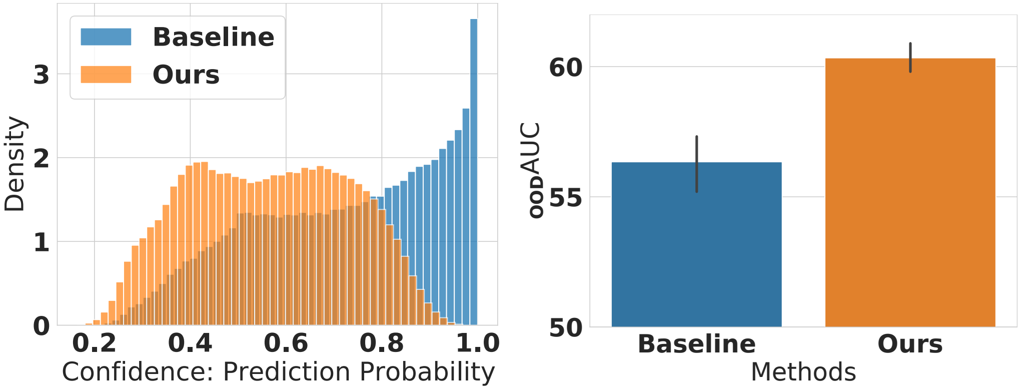


Figure 12.3.: (a) Confidence distribution for out-of-distribution (OOD) radar spectra for the Baseline (i.e. vanilla classifier training) and the best combination of methods proposed in this thesis (Ours). Additionally, the (b) outlier detection performance ($_{\mathrm{OOD}}$AUC) of these two radar classifiers. The over-confidence of deep learning classifiers is clearly visible in (a), which shows that the confidence distribution of the Baseline is severely skewed towards a 1.0, even though the classifier should not be recognizing any of the objects in the OOD dataset. We improve the $_{\mathrm{OOD}}$MMCfrom 0.715 to 0.550, which significantly reduces the confidences of predictions for OOD data. The proposed methods also improves the $_{\mathrm{OOD}}$AUCfrom 56.34% to 60.33%, allowing better detection of unknown objects. Given that classifiers cannot accurately classify all objects encountered in the real-world, these results show that this thesis contributes a better handling of such encounters by significantly lowering the confidences assigned to objects which the classifiers are not trained to classify.

12.2. Summary of key contributions

With the goal of effectuating safe and reliable automated driving, this research aimed at enhancing perception for scene understanding through object classification using radar sensors. Based on a quantitative and qualitative analysis, it can be concluded that radar-based object classification can offer an alternative robust source of information of the scene, and along with accurate uncertainty estimation, is an important factor to consider for automated driving systems. Using the methods proposed in this thesis, radar sensors can be a trustworthy source for accurate object recognition, especially for cases when other sensors fail to reliably perceive the environment, and along with uncertainty calibration techniques can also accurately quantify the uncertainty associated with these predictions. The benefits of improved uncertainty quantification also extent to other data modalities which are used for sensor perception in autonomous vehicles, and offer better assurances of the behaviour of the classifier in unknown situations when trustworthy predictions can not be made. For example, when a new unseen object is encountered, the classifiers should practice abstention instead of falsely and over-confidently assigning a class prediction. These results also set a strong foundation for downstream tasks which involve the fusion of information from multiple sensors, and allow decision-making systems to accurately estimate the reliability of each sensors predicted classification.

12.3. Answers to research questions

Throughout the thesis the main theme aimed at answering questions posed in Chap. 2, and in this section we aim to answer each of these questions in the context of the research done in this thesis.

RQ1 *How can automated driving be ameliorated for reliable and robust sensor perception?*
Reliable perception requires accurate measurements in all driving scenarios regardless of the weather, time of day, obstructions and sensor malfunctions. Given that robustness against all these factors cannot be achieved with a single sensor, it becomes important for automated driving systems to utilize an array of sensors which can collectively offer desired robustness. An important member of this array is the radar sensor and object classification through its measurements offer reliable perception in adverse weather conditions regardless of the amount of available sunlight. Through the radar spectra datasets created in Chap. 4, the study in Chap. 5 showed a significant improvement of classification generalization to novel viewing angles, object instances and different realistic environment setups. These results show that the radar spectra measurements offer an alternative source of reliable object classification and are particularly useful in scenarios where other sensor classifiers fail to offer reliable perception. In short, reliable and robust perception can be achieved through leveraging measurements from radar sensors.

RQ2 *Is optimizing the generalization performance of classification models enough for safety-critical applications?*
As much as good generalization performance is a necessity for solving classification tasks, it is by no means sufficient for the successful operation of autonomous systems. Given that reliable perception requires classification through multiple sensor modalities, generalization performance alone does not offer the tools for determining the trustworthiness of each prediction. This can only be achieved through uncertainty estimation which accurately quantifies the difficulty, and therefore the uncertainty, in determining a classification for the measurements. Current deep learning classifiers, albeit accurate and offering impressive generalization, were shown to be severely, and often wrongly, over-confident in their predictions in Part II. In Chap. 6 we described ways in which the severity of this notorious characteristic can be quantified, and in Chap. 7 and Chap. 8 we offered novel solutions to mitigate the poor uncertainty estimation of any deep learning classifier. Without such methods to accurately quantify the predictive uncertainty, the predictions cannot be trusted for further decision-making and such failures in detecting *wrong* over-confidence can often lead to accidents instead of safely practising abstention. In short, generalization performance is not the only criteria which should be used as a performance indicator, and other uncertainty estimation evaluations (e.g. those described in Chap. 6) are just as important.

RQ3 *How can classifiers learn meticulous real-time uncertainty estimation?*
Despite safety-critical application prioritizing reliable uncertainty quantification for efficient decision-making, deep learning classifier predictions remain as poor indicators of uncertainty. Many solutions in the current literature (Sec. 6.6), which aim to address this, either fail to be consistent in their performance gains (e.g. fail to generalize to classifiers with other model architectures) or do not offer real-time estimation which is vital for tasks such as automated driving. The former failure is specifically a problem after observations

from Chap. 8 show that the quality of uncertainty estimates are highly dependent on the model architecture and exact training process, thus requiring uncertainty estimates which do not over-fit to specific experimental settings (e.g. only a single model architecture). Using post-hoc calibration can offer a neat solution for improving uncertainty estimates with a real-time calibrator such as I-Max binning (Chap. 7), additionally offering a solution for already-trained and/or already-used-in-practice classification models without the need for re-training them again. Alternatively, modification of the training process also offers high-quality real-time uncertainty estimation with methods such as OMADA (Chap. 8) which are designed to implicitly optimize uncertainty calibration, in addition to generalization. In short, meticulous uncertainty estimates can be learned for deep learning classifiers using uncertainty calibration as an implicit optimization criteria through post-hoc calibration or considering calibration during the training process.

RQ4 *Can radar specific knowledge aid data-driven techniques?*
Even though the field of *deep learning for radar* is at a relatively early stage, researchers have been interested in the study of radar signals for much longer. Over the years, many observations and theories, explaining the behaviour of the radar sensor, have been published by experienced radar researchers. Instead of using this knowledge to facilitate the learning process, there seems to be a consensus of learning this knowledge directly from the data. Deep learning classifiers, which have the modelling capacity to potentially learn this directly from the data, are currently limited by the dataset size and diversity, as well as other optimization challenges such as over-fitting. In Chap. 5 we utilize radar domain knowledge, such as the range and azimuth, for over-coming some of these challenges for improved generalization performance. We also show in Chap. 11 how such information can also be used for learning radar classifiers which offer high-quality uncertainty estimation. In short, using data-driven learning algorithms which incorporate radar specific knowledge can greatly benefit the classifier performance.

12.4. Outlook

A number of future research directions can be followed to extend the research of this thesis. For furthering radar perception research, these include exploiting temporal features for classification, radar spectra generative modelling and novel evaluations of radar classification models. For uncertainty calibration, a couple of avenues can be followed to extend I-Max, for example, to other tasks.

Due to the sensitivity of radar sensors to minor viewpoint changes, the temporal transformation of the measured spectra of an object is still not entirely understood, even for the case of static objects where only the sensor is in motion. This is due to the complex interaction of radar signals with various parts of the object which depends on factors such as the object material, size, distance, surface roughness and reflectivity. Temporal features, which cannot be easily learned through domain knowledge due to the lack of understanding of this interaction process, can be found using data-driven algorithms (e.g. deep recurrent neural networks (RNNs)) which can learn from sequential temporal patterns and the relation of these patterns to past observations and predictions. In addition to identifying some known properties, such as signal strength increasing as the object is approached, deep learning algorithms can learn class specific spectra changes (e.g. approaching objects allow the radar spectra to measure reflected signals from different parts of the object which might be characteristic for classification) over time and exploit this knowledge for classification by incorporating observations and predictions from the past. For example, it could learn that larger objects are more susceptible to spectra changes, as they typically consist of more diverse parts which could all have different reflection properties. Based on the initial results presented in Chap. 5, this has the potential to significantly improve the generalization performance as classification can be based on significantly more information.

Addressing the shortage of data samples and diversity in radar datasets remains an open problem, and current generative modeling research for radar spectra is still at an early stage. Generative models have shown great success in reconstructing complex natural image scenes and offer a great foundation for further increasing and diversifying radar spectra datasets. Radar dataset measurement campaigns are typically expensive and unable to capture *all* objects from *all* viewpoints, and using available measurements to generate missing viewpoints of all objects has great consequences in furthering radar-based classification research to generalize to situations which can be expected in the real-world. This will be especially beneficial for boosting generalization performance to situations which classifier currently struggle with; for example objects at large distances or small objects. The results from Chap. 8 have shown that generative models, which are typically used for increasing and diversifying datasets, can also be used as a tool to address problems such as uncertainty calibration. A natural step for this diversification process involves using the manifold-based augmentation strategy, presented in Chap. 8 (OMADA), for generating new class-ambiguous spectra samples. In addition to generalization performance, such synthesis can aid in improving the quality of uncertainty estimation.

From the conclusions in Chap. 5 (SpecDecay) and Chap. 11 (P-smoothing), practitioners should follow research directions which aim to improve radar classification by incorporating radar domain knowledge to aid data-driven algorithms. The results of Chap. 10 highlight the need for uncertainty quantification techniques, such as OMADA (Chap. 8) and I-Max (Chap. 7), to tackle the over-confidence and mis-calibration characteristics of deep learning classifiers. As an first attempt, P-smoothing (Chap. 11), exploited the radar domain knowledge for implicitly optimizing uncertainty calibration, though future research directions can further capitalize on this to tackle other problems faced by radar spectra classification (e.g. using domain knowledge for explaining decisions made by deep learning classifiers). More domain-knowledge could also aid in learning radar-specific RNN architectures, as well as radar-specific generative modeling.

Unlike vision and most other domains, radar datasets are typically accompanied with more ground truth information than the classification label alone. These can be used for novel evaluations which can reduce the amount of realistic testing which is typically performed before deploying models in real-world systems. Similar to the evaluations in this thesis which leveraged the range information, as well as synthetic corruptions, new evaluations using this information can offer simulating challenging tests to thoroughly estimate the performance of the classifiers in the real-world. For example, in order to study the behaviour under sensor malfunctions, the spectra can be reproduced using only some of the radar receiving antennas. These would affect the angular resolution of the spectra images and simulate cases when the sensor does not operate properly.

The work in Part II can be extended in a number of ways. A natural next step is to directly extend the work to other classification settings, such as multi-label classification [Lydia & Francis, 2020] (i.e. samples can take on multiple ground truth classes) and semantic segmentation [Long et al., 2015] (i.e. pixel-level classification). As both settings still perform classification, both I-Max

(Chap. 7) and OMADA (Chap. 8) are readily applicable. For example, semantic segmentation would merely require considering each pixel as a stand-alone sample to apply I-Max calibration. Though, calibration for semantic segmentation could be greatly improved by adapting the I-Max algorithm to additionally incorporate neighboring pixel information for calibration rather than treating each pixel independently. One requirement before directly applying OMADA for semantic segmentation is learning the autoencoder-based generative model. Recent work on image generation from semantic pixel label maps [Park et al., 2019] can potentially help OMADA generate ambiguous augmentations and pixel level soft labels.

While the focus of this thesis was on classification, uncertainty calibration can also be extended to regression problems which output real, continuous predictions. One possibility is to extend the object detection work in Kuppers et al. [2020] using I-Max binning as a post-hoc uncertainty calibration step for object detection models. Using the benefits of calibrated object detection could also lead to simultaneous object detection and classification from the full radar field-of-view (FOV) which could avoid the region-of-interest (ROI) extraction step. I-Max binning can also be easily, and more directly, extended to other semantic segmentation and natural language processing tasks. Furthermore, one limitation of I-Max binning is that it cannot model class correlations as it calibrates each class independently. Another interesting direction would be to extend I-Max binning for modelling these class correlations through vector quantization.

This thesis tackled previously under-explored research directions (e.g. radar object classification and aiding data-driven algorithms with radar-specific knowledge), as well as an important and exciting research direction (uncertainty estimation for radar classification) never previously explored. The studies in this thesis also presented novel evaluation frameworks for the reliable performance evaluation of radar classifiers. It is my hope that this evaluation framework and the new research direction will set a precedent for further developments in this field.

Part V.

Appendix

Chapter 13.

Appendix: Multi-class uncertainty calibration via mutual information-based binning (I-Max)

This sections supplements the presentation of *Multi-Class Uncertainty Calibration via Mutual Information Maximization-based Binning* in Chap. 7 with the following:

A1: S vs. S_k as empirical approximations to $p(\lambda_k, y_k)$ for bin optimization in Sec. 7.2.1;

A2: Mathematical proof for Theorem 1 and algorithm details of I-Max in Sec. 7.2.2;

A3: Training details;

A4: Extend Tab. 7.1 in Sec. 7.8 for more datasets and models;

A5: Extend Tab. 7.3 in Sec. 7.10 for more scaling methods, datasets and models.

A1. S vs. S_k as empirical approximations to $p(\lambda_k, y_k)$ for bin optimization

In Sec. 7.2.1 of Chap. 7, we discussed the sample inefficiency issue when there are classes with small class priors. To tackle this, we proposed to merge the training sets $\{S_k\}$ across a selected set of classes (e.g., with similar class priors, belonging to the same class category or all classes) and use the merged S to train a single binning scheme for calibrating these classes, i.e., shared class-wise (sCW) instead of CW binning. Fig. 7.2-b) shows that after merging over the 1k ImageNet classes, the set S has sufficient numbers from both the positive $y = 1$ and negative $y = 0$ class under the one-vs-rest conversion. Tab. 7.1 showed the benefits of sCW over CW binnings. Tab. 7.4 showed that our proposal sCW is also beneficial to scaling methods which use one-vs-rest for multi-class calibration.

As pointed out in Sec. 7.2.1, both S and S_k are empirical approximations to the inaccessible ground truth $p(\lambda_k, y_k)$ for bin optimization. In Fig. A1, we empirically analyze their approximation errors. From the CIFAR10 test set, we take 5k samples to approximate per-class logit distribution $p(\lambda_k|y_k = 1)$ by means of histogram density estimation[1], and then use it as the baseline for comparison, i.e., BS_k in Fig. A1. The rest of the 5k samples in the CIFAR10 test set are reserved for constructing S_k and S. For each class, we respectively evaluate the square root of the Jensen-Shannon divergence (JSD) from the baseline BS_k to the empirical distribution of S or S_k attained at different numbers of samples.[2]

In general, Fig. A1 confirms that variance (due to not enough samples) outweights bias (due to training set merging). Nevertheless, sCW does not always have smaller JSDs than CW, for instance, the class 7 with the samples larger than 2k (the blue bar "sCW" is larger than the orange bar "CW"). So, for the class-7, the bias of merging logits starts outweighing the variance when the number of samples is more than 2k. Unfortunately, we don't have more samples to further evaluate JSDs, i.e., making the variance sufficiently small to reveal the bias impact. Another reason that we don't observe large JSDs of sCW for CIFAR10 is that the logit distributions of the 10 classes are similar. Therefore, the bias of sCW is small, making CIFAR10 a good use case of sCW. From CIFAR10 to CIFAR100 and ImageNet, there are more classes with even smaller class priors. Therefore, we expect that the sample inefficiency issue of S_k becomes more critical. It will be beneficial to exploit sCW for bin optimization as well as for other methods based on the one-vs-rest conversion for multi-class calibration.

A2. Proof of Theorem 1

In this section, we prove Theorem 1 in Sec. 7.2.2, discuss the connection to the information bottleneck (IB) [Tishby et al., 1999], analyze the convergence behavior of the iterative method derived in Sec. 7.2.2 and modify the k-means++ algorithm [Arthur & Vassilvitskii, 2007] for initialization. To assist the implementation of the iterative method, we further provide the pseudo code and perform complexity/memory cost analysis.

[1] Here, we focus on $p(\lambda_k|y_k = 1)$ as its empirical estimation suffers from small class priors, being much more challenging than $p(\lambda_k|y_k = 0)$ as illustrated in Fig. 7.2.

[2] Note, for JSD evaluation, the histogram estimator sets the bin number as the maximum of 'sturges' and 'fd' estimators, both of them optimize their bin setting towards the number of samples.

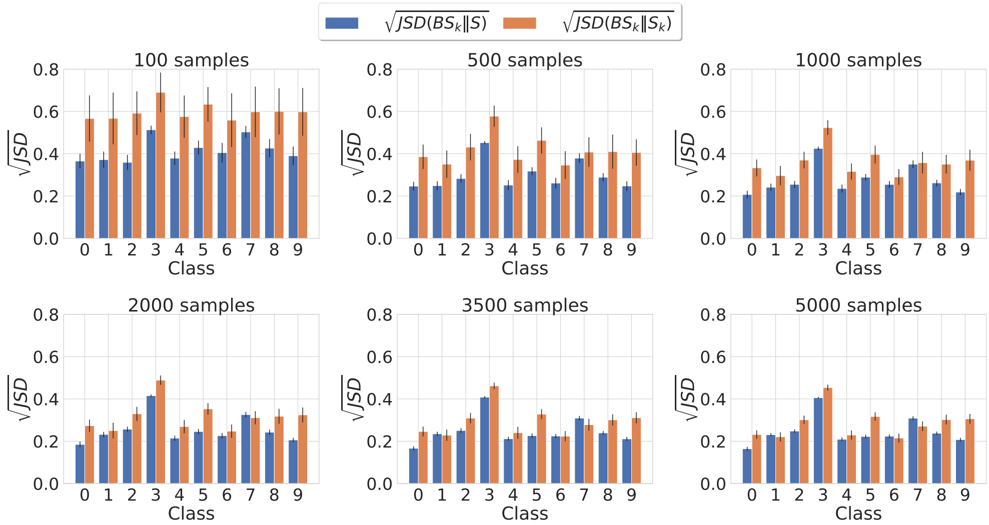


Figure A1.: Empirical approximation error of S vs. S_k, where Jensen-Shannon divergence (JSD) is used to measure the difference between the empirical distributions underlying the calibration sets for class-wise bin optimization. Overall, the merged set S is a more sample efficient choice over S_k.

Theorem 2. *The mutual information (MI) maximization problem given as follows:*

$$\{g_m^*\} = \arg \max_{Q:\{g_m\}} I(y; m = Q(\lambda)) \tag{13.1}$$

is equivalent to

$$\max_{Q:\{g_m\}} I(y; m = Q(\lambda)) \equiv \min_{\{g_m, \phi_m\}} \mathcal{L}(\{g_m, \phi_m\}) \tag{13.2}$$

where the loss $\mathcal{L}(\{g_m, \phi_m\})$ *is defined as*

$$\mathcal{L}(\{g_m, \phi_m\}) \triangleq \sum_{m=0}^{M-1} \int_{g_m}^{g_{m+1}} p(\lambda) \sum_{y' \in \{0,1\}} P(y = y' | \lambda) \log \frac{P(y = y')}{P_\sigma(y = y'; \phi_m)} \mathrm{d}\lambda \tag{13.3}$$

$$\textit{with} \quad P_\sigma(y; \phi_m) \triangleq \sigma\left[(2y - 1)\phi_m\right]. \tag{13.4}$$

As a set of real-valued auxiliary variables, $\{\phi_m\}$ *are introduced here to ease the optimization.*

***Proof*.** Before staring our proof, we note that the upper-case P indicates probability mass functions of discrete random variables, e.g., the label $y \in \{0,1\}$ and the bin interval index $m \in \{1, \ldots, M\}$, whereas the lower-case p is reserved for probability density functions of continuous random variables, e.g., the raw logit $\lambda \in \mathbb{R}$.

The key to prove the equivalence is to show the inequality

$$I(y; m = Q(\lambda)) \geq -\mathcal{L}(\{g_m,\phi_m\}), \tag{13.5}$$

and the equality is attainable by minimizing $\mathcal{L}$ over $\{\phi_m\}$.

By the definition of MI, we firstly expand $I(y; m = Q(\lambda))$ as

$$I(y; m = Q(\lambda)) = \sum_{m=0}^{M-1} \int\limits_{g_m}^{g_{m+1}} p(\lambda) \sum_{y' \in \{0,1\}} P(y = y'|\lambda) \log \frac{P(y = y'|m)}{P(y = y')} \mathrm{d}\lambda, \tag{13.6}$$

where the conditional distribution $P(y|m)$ is given as

$$P(y|m) = P(y|\lambda \in [g_m, g_{m+1})) = \frac{P(y) \int_{g_m}^{g_{m+1}} p(\lambda|y)\mathrm{d}\lambda}{\int_{g_m}^{g_{m+1}} p(\lambda)\mathrm{d}\lambda} = \frac{\int_{g_m}^{g_{m+1}} p(\lambda|y)P(y)\mathrm{d}\lambda}{\int_{g_m}^{g_{m+1}} p(\lambda)\mathrm{d}\lambda}. \tag{13.7}$$

From the above expression, we note that MI maximization effectively only accounts to the bin edges $\{g_m\}$. The bin representatives can be arbitrary as long as they can indicate the condition $\lambda \in [g_m, g_{m+1})$. So, the bin interval index m is sufficient to serve the role in conditioning the probability mass function of y, i.e., $P(y|m)$. After optimizing the bin edges, we have the freedom to set the bin representatives for the sake of post-hoc calibration.

Next, based on the MI expression, we compute its sum with $\mathcal{L}$

$$\begin{aligned} I(y; Q(\lambda)) + \mathcal{L}(\{g_m,\phi_m\}) &= \sum_{m=0}^{M-1} \int\limits_{g_m}^{g_{m+1}} p(\lambda) \sum_{y' \in \{0,1\}} P(y = y'|\lambda)\mathrm{d}\lambda \log \frac{P(y = y'|m)}{P_\sigma(y = y'; \phi_m)} \\ &\overset{(a)}{=} \sum_{m=0}^{M-1} P(m) \left[\sum_{y' \in \{0,1\}} P(y = y'|m) \log \frac{P(y = y'|m)}{P_\sigma(y = y'; \phi_m)} \right] \\ &\overset{(b)}{=} \sum_{m=0}^{M-1} P(m) \mathrm{KLD} \left[P(y = y'|m) \| P_\sigma(y = y'; \phi_m) \right] \\ &\overset{(c)}{\geq} 0. \end{aligned} \tag{13.8}$$

The equality (a) is based on

$$\int_{g_m}^{g_{m+1}} p(\lambda)P(y=y'|\lambda)\mathrm{d}\lambda = P(y=y',\lambda\in[g_m,g_{m+1})) = \underbrace{P(\lambda\in[g_m,g_{m+1}))}_{=P(m)}P(y=y'|m). \tag{13.9}$$

From the equality (a) to (b), it is simply because of identifying the term in $[\cdot]$ of the equality (a) as the Kullback-Leibler divergence (KLD) between two probability mass functions of y. As the probability mass function $P(m)$ and the KLD both are non-negative, we reach to the inequality at (c), where the equality holds if $P_\sigma(y;\phi_m) = P(y|m)$. By further noting that $\mathcal{L}$ is convex over $\{\phi_m\}$ and $P_\sigma(y;\phi_m) = P(y|m)$ minimizes the KLD and thus nulls out its gradient over $\{\phi_m\}$, we then reach to

$$I(y;Q(\lambda)) + \min_{\{\phi_m\}} \mathcal{L}(\{g_m,\phi_m\}) = 0. \tag{13.10}$$

The obtained equality then concludes our proof

$$\begin{aligned}\max_{\{g_m\}} I(y;Q(\lambda)) = \max_{\{g_m\}}\left[-\min_{\{\phi_m\}}\mathcal{L}(\{g_m,\phi_m\})\right] &= -\min_{\{g_m,\phi_m\}}\mathcal{L}(\{g_m,\phi_m\})\\ &\equiv \min_{\{g_m,\phi_m\}}\mathcal{L}(\{g_m,\phi_m\}).\end{aligned} \tag{13.11}$$

□

Lastly, we note that $\mathcal{L}(\{g_m,\phi_m\})$ can reduce to a NLL loss (as $P(y)$ in the log probability ratio is omittable), which is a common loss for calibrators. However, only through this equivalence proof and the MI maximization formulation, we can clearly identify the great importance of bin edges in preserving label information. So even though $\{g_m,\phi_m\}$ are jointly optimized in the equivalent problem, only $\{g_m\}$ play the determinant role in maximizing the MI.

A3. Derivation of the closed-form updates

In this section, we provide the derivation of the following closed-form updates (i.e. Eq. 7.4),

$$g_m = \log\left\{\frac{\log\left[\frac{1+e^{\phi_m}}{1+e^{\phi_{m-1}}}\right]}{\log\left[\frac{1+e^{-\phi_{m-1}}}{1+e^{-\phi_m}}\right]}\right\},\ \phi_m = \log\left\{\frac{\int_{g_m}^{g_{m+1}}\sigma(\lambda)p(\lambda)\mathrm{d}\lambda}{\int_{g_m}^{g_{m+1}}\sigma(-\lambda)p(\lambda)\mathrm{d}\lambda}\right\} \approx \log\left\{\frac{\sum_{\lambda_n\in\mathcal{S}_m}\sigma(\lambda_n)}{\sum_{\lambda_n\in\mathcal{S}_m}\sigma(-\lambda_n)}\right\}, \tag{13.12}$$

To minimize the loss $\mathcal{L}(\{g_i,\phi_i\})$ as given in (13.3),

$$\mathcal{L}(\{g_m,\phi_m\}) \triangleq \sum_{m=0}^{M-1} \int_{g_m}^{g_{m+1}} p(\lambda) \sum_{y'\in\{0,1\}} P(y=y'|\lambda) \log \frac{P(y=y')}{\sigma\left((2y'-1)\phi_m\right)} \mathrm{d}\lambda \tag{13.13}$$

$$\mathcal{L}(\{g_m,\phi_m\}) \triangleq \sum_{y'\in\{0,1\}} \sum_{m=0}^{M-1} \int_{g_m}^{g_{m+1}} p(\lambda) P(y=y'|\lambda) \log \left[P(y=y')(1+e^{(1-2y')\phi_m})\right] \mathrm{d}\lambda, \tag{13.14}$$

we first compute the derivative with respect to g_i as

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial g_i} = \sum_{y'\in\{0,1\}} & P(\lambda=g_m)P(y=y'|\lambda=g_i)\cdot\log\left[P(y=y')(1+e^{(1-2y)\phi_{m-1}})))\right] \\ & - P(\lambda=g_m)P(y=y'|\lambda=g_i)\cdot\log\left[P(y=y')(1+e^{(1-2y)\phi_m})))\right]\end{aligned} \tag{13.15}$$

$$\frac{\partial\mathcal{L}}{\partial g_i} = \sum_{y'\in\{0,1\}} P(\lambda=g_m)P(y=y'|\lambda=g_i)\cdot\log\left[\frac{1+e^{(1-2y')\phi_{m-1}}}{1+e^{(1-2y')\phi_m}}\right] \tag{13.16}$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial g_i} &= P(\lambda=g_m)P(y=0|\lambda=g_i)\cdot\log\left[\frac{1+e^{\phi_{m-1}}}{1+e^{\phi_m}}\right] \\ &\quad + P(\lambda=g_m)P(y=1|\lambda=g_i)\cdot\log\left[\frac{1+e^{-\phi_{m-1}}}{1+e^{-\phi_m}}\right],\end{aligned} \tag{13.17}$$

since $P(\lambda=g_m)$ is positive, the stationary point equation for g is,

$$P(y=1|\lambda=g_i)\cdot\log\left[\frac{1+e^{-\phi_{m-1}}}{1+e^{-\phi_m}}\right] = P(y=0|\lambda=g_i)\cdot\log\left[\frac{1+e^{\phi_m}}{1+e^{\phi_{m-1}}}\right] \tag{13.18}$$

$$\frac{P(y=1|\lambda=g_i)}{P(y=0|\lambda=g_i)} = \frac{\log\left[\frac{1+e^{\phi_m}}{1+e^{\phi_{m-1}}}\right]}{\log\left[\frac{1+e^{-\phi_{m-1}}}{1+e^{-\phi_m}}\right]} \tag{13.19}$$

We can further write this in terms of log likelihood ratios,

$$\log \frac{P(y=1|\lambda=g_i)}{P(y=0|\lambda=g_i)} = \log \left\{ \frac{\log\left[\frac{1+e^{\phi_m}}{1+e^{\phi_{m-1}}}\right]}{\log\left[\frac{1+e^{-\phi_{m-1}}}{1+e^{-\phi_m}}\right]} \right\} \tag{13.20}$$

In the case when λ is a log-likelihood ratio, the LHS of Eq. 13.20 is equal to g_m. This yields the following closed form,

$$g_m = \log \left\{ \frac{\log\left[\frac{1+e^{\phi_m}}{1+e^{\phi_{m-1}}}\right]}{\log\left[\frac{1+e^{-\phi_{m-1}}}{1+e^{-\phi_m}}\right]} \right\} \tag{13.21}$$

Next, we compute the derivative with respect to ϕ_i as

$$\frac{\partial \mathcal{L}}{\partial \phi_i} = \sum_{y' \in \{0,1\}} \int\limits_{g_m}^{g_{m+1}} p(\lambda) P(y=y'|\lambda) \frac{(1-2y')e^{(1-2y')\phi_m}}{1+e^{(1-2y')\phi_m}} \mathrm{d}\lambda \tag{13.22}$$

$$\frac{\partial \mathcal{L}}{\partial \phi_i} = \int\limits_{g_m}^{g_{m+1}} p(\lambda) P(y=0|\lambda) \mathrm{d}\lambda \cdot \frac{e^{\phi_m}}{1+e^{\phi_m}} - \int\limits_{g_m}^{g_{m+1}} p(\lambda) P(y=1|\lambda) \mathrm{d}\lambda \cdot \frac{e^{-\phi_m}}{1+e^{-\phi_m}} \tag{13.23}$$

The stationary point equation for ϕ is,

$$\frac{\int_{g_m}^{g_{m+1}} p(\lambda) P(y=1|\lambda) \mathrm{d}\lambda}{\int_{g_m}^{g_{m+1}} p(\lambda) P(y=0|\lambda) \mathrm{d}\lambda} = \frac{\frac{e^{\phi_m}}{1+e^{\phi_m}}}{\frac{e^{-\phi_m}}{1+e^{-\phi_m}}} \tag{13.24}$$

$$= \frac{e^{2\phi_m}(1+e^{-\phi_m})}{1+e^{\phi_m}} \tag{13.25}$$

$$= \frac{e^{\phi_m}(1+e^{\phi_m})}{1+e^{\phi_m}} \tag{13.26}$$

$$\log \frac{\int_{g_m}^{g_{m+1}} p(\lambda) P(y=1|\lambda) \mathrm{d}\lambda}{\int_{g_m}^{g_{m+1}} p(\lambda) P(y=0|\lambda) \mathrm{d}\lambda} = \log e^{\phi_m} \tag{13.27}$$

$$= \phi_m \tag{13.28}$$

When the conditional distribution $P(y|\lambda)$ takes the sigmoid model, i.e., $P(y|\lambda) \approx \sigma[(2y-1)\lambda]$, the stationary points of $\mathcal{L}$ gradients over ϕ_m, has a closed-form expression,

$$\phi_m = \log \left\{ \frac{\int_{g_m}^{g_{m+1}} \sigma(\lambda) p(\lambda) \mathrm{d}\lambda}{\int_{g_m}^{g_{m+1}} \sigma(-\lambda) p(\lambda) \mathrm{d}\lambda} \right\} \approx \log \left\{ \frac{\sum_{\lambda_n \in \mathcal{S}_m} \sigma(\lambda_n)}{\sum_{\lambda_n \in \mathcal{S}_m} \sigma(-\lambda_n)} \right\}, \tag{13.29}$$

A4. Training details

A4.1. Pre-trained classification networks

We evaluate post-hoc calibration methods on four benchmark datasets, i.e., ImageNet [Deng et al., 2009], CIFAR-100 [Krizhevsky & Hinton, 2009], CIFAR-10 [Krizhevsky & Hinton, 2009] and SVHN [Netzer et al., 2011], and across three modern DNNs for each dataset, i.e., InceptionResNetV2 [Szegedy et al., 2017], DenseNet161 [Huang et al., 2017] and ResNet152 [He et al., 2016] for ImageNet, and Wide ResNet (WRN) [Zagoruyko & Komodakis, 2016] for the two CIFAR datasets and SVHN. Additionally, we train DenseNet-BC ($L = 190$, $k = 40$) [Huang et al., 2017] and ResNext8x64 [Xie et al., 2017] for the two CIFAR datasets.

The ImageNet and CIFAR models are publicly available pre-trained networks and details are reported at the respective websites, i.e., ImageNet classifiers: https://github.com/Cadene/pretrained-models.pytorch and CIFAR classifiers: https://github.com/bearpaw/pytorch-classification.

A4.2. Training scaling methods

The hyper-parameters were decided based on the original respective scaling methods publications with some exceptions. We found that the following parameters were the best for all the scaling methods. All scaling methods use the Adam optimizer with batch size 256 for CIFAR and 4096 for ImageNet. The learning rate was set to 10^{-3} for temperature scaling [Guo et al., 2017] and Platt scaling [Platt, 1999], 10^{-5} for vector scaling [Guo et al., 2017] and 10^{-5} for matrix scaling [Guo et al., 2017]. Matrix scaling was further regularized as suggested by [Kull et al., 2019] with a L_2 loss on the bias vector and the off-diagonal elements of the weighting matrix. BBQ [Naeini et al., 2015], isotonic regression [Zadrozny & Elkan, 2002] and Beta [Kull et al., 2017] hyper-parameters were taken directly from [Wenger et al., 2020].

A4.3. Training I-Max binning

The I-Max bin optimization started from k-means++ initialization, which uses JSD instead of Euclidean metric as the distance measure, see Sec. 7.5. Then, we iteratively and alternatively updated $\{g_m\}$ and $\{\phi_m\}$ according to (7.4) until 200 iterations. With the attained bin edges $\{g_m\}$, we set the bin representatives $\{r_m\}$ based on the empirical frequency of class-1. If a scaling method is combined with binning, an alternative setting for $\{r_m\}$ is to take the averaged prediction probabilities based on the scaled logits of the samples per bin, e.g., in Tab. 7.3

in Sec. 7.10. Note that, for CW binning in 7.1, the number of samples from the minority class is too few, i.e., $25\text{k}/1\text{k} = 25$. We only have about $25/15 \approx 2$ samples per bin (assuming 15 bins), which are too few to use empirical frequency estimates. Alternatively, we set $\{r_m\}$ based on the raw prediction probabilities. For ImageNet and CIFAR 10/100, which have test sets with uniform class priors, the used sCW setting is to share one binning scheme among all classes. Alternatively, for the imbalanced multi-class SVHN setting, we share binning among classes with similar class priors, and thus use the following class (i.e. digit) groupings: $\{0 - 1\}, \{2 - 4\}, \{5 - 9\}$.

A5. Extend Tab. 1 for additional datasets and models.

Tab. 7.1 in Sec. 7.8 of Chap. 7 is replicated across datasets and models, where the basic setting remains the same. Specifically, three different ImageNet models can be found in Tab. A1, Tab. A2 and Tab. A3. Three models for CIFAR100 can be found in Tab. A4, Tab. A5 and Tab. A6. Similarly, CIFAR10 models can be found in Tab. A7, Tab. A8 and Tab. A9. The accuracy degradation of Eq. Mass reduces as the dataset has less number of classes, e.g., CIFAR10. This is a result of a higher class prior, where the one-vs-rest conversion becomes less critical for CIFAR10 than ImageNet. Nevertheless, its accuracy losses are still much larger than the other binning schemes, i.e., Eq. Size and I-Max binning. Therefore, its calibration performance is not considered for comparison. Overall, the observations of Tab. A1- A9 are similar to Tab. 7.1, showing the stable performance gains of I-Max binning across datasets and models.

A6. Extend Tab. 2 for additional scaling methods, datasets and models

Tab. 7.3 in Sec. 7.10 of Chap. 7 is replicated across datasets and models, and include more scaling methods for comparison. The three binning methods all use the shared CW strategy, therefore 1k calibration samples are sufficient. The basic setting remains the same as Tab. 7.3. Three different ImageNet models can be found in Tab. A10, Tab. A11 and Tab. A12. Three models for CIFAR100 can be found in Tab. A13, Tab. A14 and Tab. A15. Similarly, CIFAR10 models can be found in Tab. A16, Tab. A17 and Tab. A18.

Being analogous to Tab. 7.3, we observe that in most cases matrix scaling performs the best at the accuracy, but fail to provide satisfactory calibration performance measured by ECEs, Brier scores and NLLs. Among the scaling methods, GP [Wenger et al., 2020] is the top performing one. Among the binning

Table A1.: Tab. 1 Extension: ImageNet - InceptionResNetV2

Binn.	sCW(?)	size	Acc_{top1} ↑	Acc_{top5} ↑	$_{\text{CW}}\text{ECE}_{\frac{1}{K}}$ ↓	$_{\text{top1}}\text{ECE}$ ↓	NLL
Baseline	✗	-	**80.33** ± 0.15	**95.10** ± 0.15	0.0486 ± 0.0003	0.0357 ± 0.0009	0.8406 ± 0.0095
Eq. Mass	✗	25k	7.78 ± 0.15	27.92 ± 0.71	0.0016 ± 0.0001	0.0606 ± 0.0013	3.5960 ± 0.0137
Eq. Mass	✓	1k	5.02 ± 0.13	26.75 ± 0.37	0.0022 ± 0.0001	0.0353 ± 0.0012	3.5272 ± 0.0142
Eq. Size	✗	25k	78.52 ± 0.15	89.06 ± 0.13	0.1344 ± 0.0005	0.0547 ± 0.0017	1.5159 ± 0.0136
Eq. Size	✓	1k	80.14 ± 0.23	88.99 ± 0.12	0.1525 ± 0.0023	0.0279 ± 0.0043	1.2671 ± 0.0130
I-Max	✗	25k	80.27 ± 0.17	95.01 ± 0.19	0.0342 ± 0.0006	0.0329 ± 0.0010	0.8499 ± 0.0105
I-Max	✓	1k	80.20 ± 0.18	94.86 ± 0.17	**0.0302** ± 0.0041	**0.0200** ± 0.0033	**0.7860** ± 0.0208

Table A2.: Tab. 1 Extension: ImageNet - DenseNet

Binn.	sCW(?)	size	Acc_{top1} ↑	Acc_{top5} ↑	$_{\text{CW}}\text{ECE}_{\frac{1}{K}}$ ↓	$_{\text{top1}}\text{ECE}$ ↓	NLL
Baseline	✗	-	**77.21** ± 0.12	**93.51** ± 0.14	0.0502 ± 0.0006	0.0571 ± 0.0014	0.9418 ± 0.0120
Eq. Mass	✗	25k	18.48 ± 0.19	45.12 ± 0.26	0.0017 ± 0.0000	0.1657 ± 0.0020	2.9437 ± 0.0162
Eq. Mass	✓	1k	17.21 ± 0.47	45.69 ± 1.22	0.0054 ± 0.0004	0.1572 ± 0.0047	2.9683 ± 0.0561
Eq. Size	✗	25k	74.34 ± 0.28	88.27 ± 0.11	0.1272 ± 0.0011	0.0660 ± 0.0018	1.6699 ± 0.0165
Eq. Size	✓	1k	77.06 ± 0.28	88.22 ± 0.10	0.1519 ± 0.0016	0.0230 ± 0.0050	1.3948 ± 0.0105
I-Max	✗	25k	77.07 ± 0.13	93.40 ± 0.17	0.0334 ± 0.0004	0.0577 ± 0.0008	0.9492 ± 0.0130
I-Max	✓	1k	77.13 ± 0.14	93.34 ± 0.17	**0.0263** ± 0.0119	**0.0201** ± 0.0088	**0.9229** ± 0.0103

Table A3.: Tab. 1 Extension: ImageNet - ResNet152

Binn.	sCW(?)	size	Acc_{top1} ↑	Acc_{top5} ↑	$_{\text{CW}}\text{ECE}_{\frac{1}{K}}$ ↓	$_{\text{top1}}\text{ECE}$ ↓	NLL
Baseline	✗	-	**78.33** ± 0.17	**94.00** ± 0.14	0.0500 ± 0.0004	0.0512 ± 0.0018	0.8760 ± 0.0133
Eq. Mass	✗	25k	17.45 ± 0.10	44.87 ± 0.37	0.0017 ± 0.0000	0.1555 ± 0.0010	2.9526 ± 0.0168
Eq. Mass	✓	1k	16.25 ± 0.54	45.53 ± 0.81	0.0064 ± 0.0004	0.1476 ± 0.0054	2.9471 ± 0.0556
Eq. Size	✗	25k	75.50 ± 0.28	88.85 ± 0.19	0.1223 ± 0.0008	0.0604 ± 0.0017	1.6012 ± 0.0252
Eq. Size	✓	1k	78.24 ± 0.16	88.81 ± 0.19	0.1480 ± 0.0015	0.0286 ± 0.0053	1.3308 ± 0.0178
I-Max	✗	25k	78.24 ± 0.16	93.91 ± 0.17	0.0334 ± 0.0005	0.0521 ± 0.0015	0.8842 ± 0.0135
I-Max	✓	1k	78.19 ± 0.21	93.82 ± 0.17	**0.0295** ± 0.0030	**0.0196** ± 0.0049	**0.8638** ± 0.0135

schemes, our proposal of I-Max binning outperforms Eq. Mass and Eq. Size at accuracies, ECEs, NLLs and Brier scores. The combination of I-Max binning with GP excels at the ECE performance. Note that, among all methods, Eq. Mass binning and VUC [Kumar et al., 2019] suffer from severe accuracy degradation after multi-class calibration. The reason behind Eq. Mass binning was discussed in Sec. 7.2.2 of Chap. 7. As VUC [Kumar et al., 2019] combined Eq. Mass binning with Platt scaling, its accuracy loss is due to the same reason. Given their poor accuracy, they are not in the scope of calibration performance comparison.

We also observe that GP performs better at NLL/Brier than the I-Max variants. GP is trained by directly optimizing the NLL as its loss. As a non-parametric Bayesian method, GP has larger model expressive capacity than binning. While achieving better NLL/Brier, it costs significantly more computational complexity and memory. In contrast, I-Max only relies on logic comparisons at test time.

Table A4.: Tab. 1 Extension: CIFAR100 - WRN

Binn.	sCW(?)	size	$\mathrm{Acc}_{\mathrm{top1}} \uparrow$	$_{\mathrm{CW}}\mathrm{ECE}_{\frac{1}{K}} \downarrow$	$_{\mathrm{top1}}\mathrm{ECE} \downarrow$	NLL
Baseline	✗	-	**81.35** ± 0.13	0.1113 ± 0.0010	0.0748 ± 0.0018	0.7816 ± 0.0076
Eq. Mass	✗	5k	60.78 ± 0.62	0.0129 ± 0.0010	0.4538 ± 0.0074	1.1084 ± 0.0117
Eq. Mass	✓	1k	62.04 ± 0.53	0.0252 ± 0.0032	0.4744 ± 0.0049	1.1789 ± 0.0308
Eq. Size	✗	5k	80.39 ± 0.36	0.1143 ± 0.0013	0.0783 ± 0.0032	1.0772 ± 0.0184
Eq. Size	✓	1k	81.12 ± 0.15	0.1229 ± 0.0030	0.0273 ± 0.0055	1.0165 ± 0.0105
I-Max	✗	5k	81.22 ± 0.12	0.0692 ± 0.0020	0.0751 ± 0.0024	0.7878 ± 0.0090
I-Max	✓	1k	81.30 ± 0.22	**0.0518** ± 0.0036	**0.0231** ± 0.0067	**0.7593** ± 0.0085

Table A5.: Tab. 1 Extension: CIFAR100 - ResNeXt8x64

Binn.	sCW(?)	size	$\mathrm{Acc}_{\mathrm{top1}} \uparrow$	$_{\mathrm{CW}}\mathrm{ECE}_{\frac{1}{K}} \downarrow$	$_{\mathrm{top1}}\mathrm{ECE} \downarrow$	NLL
Baseline	✗	-	81.93 ± 0.08	0.0979 ± 0.0015	0.0590 ± 0.0028	0.7271 ± 0.0026
Eq. Mass	✗	5k	63.02 ± 0.54	0.0131 ± 0.0012	0.4764 ± 0.0057	1.0535 ± 0.0191
Eq. Mass	✓	1k	64.48 ± 0.64	0.0265 ± 0.0011	0.4980 ± 0.0070	1.1232 ± 0.0277
Eq. Size	✗	5k	80.81 ± 0.26	0.1070 ± 0.0008	0.0700 ± 0.0030	1.0178 ± 0.0066
Eq. Size	✓	1k	81.99 ± 0.21	0.1195 ± 0.0013	0.0230 ± 0.0033	0.9556 ± 0.0071
I-Max	✗	5k	**81.99** ± 0.08	0.0601 ± 0.0027	0.0627 ± 0.0034	0.7318 ± 0.0026
I-Max	✓	1k	81.96 ± 0.14	**0.0549** ± 0.0081	**0.0205** ± 0.0074	**0.7127** ± 0.0040

Table A6.: Tab. 1 Extension: CIFAR100 - DenseNet

Binn.	sCW(?)	size	$\mathrm{Acc}_{\mathrm{top1}} \uparrow$	$_{\mathrm{CW}}\mathrm{ECE}_{\frac{1}{K}} \downarrow$	$_{\mathrm{top1}}\mathrm{ECE} \downarrow$	NLL
Baseline	✗	-	**82.36** ± 0.26	0.1223 ± 0.0008	0.0762 ± 0.0015	0.7542 ± 0.0143
Eq. Mass	✗	5k	57.23 ± 0.50	0.0117 ± 0.0011	0.4173 ± 0.0051	1.1819 ± 0.0228
Eq. Mass	✓	1k	58.11 ± 0.21	0.0233 ± 0.0005	0.4339 ± 0.0024	1.2049 ± 0.0405
Eq. Size	✗	5k	81.35 ± 0.23	0.1108 ± 0.0017	0.0763 ± 0.0029	1.0207 ± 0.0183
Eq. Size	✓	1k	82.22 ± 0.30	0.1192 ± 0.0024	0.0219 ± 0.0021	0.9482 ± 0.0137
I-Max	✗	5k	82.35 ± 0.26	0.0740 ± 0.0007	0.0772 ± 0.0010	0.7618 ± 0.0145
I-Max	✓	1k	82.32 ± 0.22	**0.0546** ± 0.0122	**0.0189** ± 0.0071	**0.7022** ± 0.0124

Among the binning schemes, I-Max w. GP achieves the best NLL/Brier across the datasets and models. It is noted that I-Max w. GP remains to be a binning scheme. So, the combination does not change the model capacity of I-Max. GP is only exploited during training to improve the optimization on I-Max's bin representatives. Besides the low complexity benefit, I-Max w. GP as a binning scheme does not suffer from the ECE underestimation issue of scaling methods such as GP.

We further note that as a cross entropy measure between two distributions, the NLL would be an ideal metric for calibration evaluation. However, *empirical* NLL and Brier favor high accuracy and high confident classifiers, as each sample only having one hard label essentially implies the maximum confidence on a single class. For this reason, during training, the empirical NLL loss will keep pushing the prediction probability to one even after reaching 100% training set accuracy.

Table A7.: Tab. 1 Extension: CIFAR10 - WRN

Binn.	sCW(?)	size	$\text{Acc}_{\text{top1}} \uparrow$	$_{\text{CW}}\text{ECE}_{\frac{1}{K}} \downarrow$	$_{\text{top1}}\text{ECE} \downarrow$	NLL
Baseline	✗	-	**96.12** ± 0.14	0.0457 ± 0.0011	0.0288 ± 0.0007	0.1682 ± 0.0062
Eq. Mass	✗	5k	91.06 ± 0.54	0.0180 ± 0.0045	0.0794 ± 0.0066	0.2066 ± 0.0091
Eq. Mass	✓	1k	91.24 ± 0.27	0.0212 ± 0.0009	0.0836 ± 0.0091	0.2252 ± 0.0220
Eq. Size	✗	5k	96.04 ± 0.14	0.0344 ± 0.0008	0.0290 ± 0.0013	0.2231 ± 0.0074
Eq. Size	✓	1k	96.04 ± 0.15	0.0278 ± 0.0021	0.0105 ± 0.0028	0.2744 ± 0.0812
I-Max	✗	5k	96.10 ± 0.14	0.0329 ± 0.0011	0.0276 ± 0.0007	0.1704 ± 0.0067
I-Max	✓	1k	96.06 ± 0.13	**0.0304** ± 0.0012	**0.0113** ± 0.0039	**0.1595** ± 0.0604

Table A8.: Tab. 1 Extension: CIFAR10 - ResNext8x64

Binn.	sCW(?)	size	$\text{Acc}_{\text{top1}} \uparrow$	$_{\text{CW}}\text{ECE}_{\frac{1}{K}} \downarrow$	$_{\text{top1}}\text{ECE} \downarrow$	NLL
Baseline	✗	-	**96.30** ± 0.18	0.0485 ± 0.0014	0.0201 ± 0.0021	**0.1247** ± 0.0058
Eq. Mass	✗	5k	89.40 ± 0.55	0.0168 ± 0.0037	0.0589 ± 0.0052	0.2011 ± 0.0085
Eq. Mass	✓	1k	89.85 ± 0.61	0.0269 ± 0.0051	0.0676 ± 0.0127	0.2208 ± 0.0172
Eq. Size	✗	5k	96.30 ± 0.20	0.0274 ± 0.0013	0.0174 ± 0.0013	0.1613 ± 0.0101
Eq. Size	✓	1k	96.17 ± 0.24	0.0288 ± 0.0039	0.0114 ± 0.0025	0.2495 ± 0.0571
I-Max	✗	5k	96.26 ± 0.20	**0.0240** ± 0.0020	0.0167 ± 0.0014	0.1264 ± 0.0066
I-Max	✓	1k	96.22 ± 0.21	0.0254 ± 0.0030	**0.0104** ± 0.0025	0.1397 ± 0.0276

Table A9.: Tab. 1 Extension Dataset: CIFAR10 - DenseNet

Binn.	sCW(?)	size	$\text{Acc}_{\text{top1}} \uparrow$	$_{\text{CW}}\text{ECE}_{\frac{1}{K}} \downarrow$	$_{\text{top1}}\text{ECE} \downarrow$	NLL
Baseline	✗	-	96.65 ± 0.09	0.0404 ± 0.001	0.0253 ± 0.0009	0.1564 ± 0.0075
Eq. Mass	✓	1k	88.80 ± 0.47	0.0233 ± 0.0024	0.0637 ± 0.0023	0.2694 ± 0.0274
Eq. Mass	✗	5k	89.51 ± 0.36	0.0137 ± 0.0039	0.0657 ± 0.0041	0.2283 ± 0.0101
Eq. Size	✓	1k	96.64 ± 0.22	0.0262 ± 0.0035	0.0101 ± 0.0035	0.2465 ± 0.0543
Eq. Size	✗	5k	96.74 ± 0.07	0.0301 ± 0.0012	0.0242 ± 0.0013	0.1912 ± 0.0075
I-Max	✓	1k	96.59 ± 0.32	0.0261 ± 0.0025	0.0098 ± 0.0027	0.1208 ± 0.0044
I-Max	✗	5k	96.71 ± 0.09	0.0284 ± 0.0013	0.0233 ± 0.0009	0.1608 ± 0.0086

As a result, the trained classifier showed poor calibration performance at test time [Guo et al., 2017]. In contrast to NLL/Brier, empirical ECEs use hard labels differently. The ground truth correctness associated to the prediction confidence p is estimated by averaging over the hard labels of the samples receiving the prediction probability p or close to p. Due to averaging, the empirical ground truth correctness is usually not a hard label. Lastly, we use a small example to show the difference between the NLL/Brier and ECE: for N predictions, all assigned a confidence of 1.0 and containing M mistakes, the calibrated confidence is $M/N < 1$. Unlike ECE, the NLL/Brier loss is only non-zero only for the M wrong predictions, despite all N predictions being miscalibrated. This example shows that NLL/Brier penalize miscalibration far less than ECE.

Table A10.: Tab. 2 Extension: ImageNet - InceptionResnetV2

Calibrator	Acc$_{\text{top1}}$ ↑	Acc$_{\text{top5}}$ ↑	$_{\text{CW}}$ECE$_{\frac{1}{K}}$ ↓	$_{\text{top1}}$ECE ↓	NLL	Brier
Baseline	80.33 ± 0.15	95.10 ± 0.15	0.0486 ± 0.0003	0.0357 ± 0.0009	0.8406 ± 0.0095	0.1115 ± 0.0007
	25k Calibration Samples					
BBQ	53.89 ± 0.30	88.63 ± 0.22	0.0287 ± 0.0009	0.2689 ± 0.0033	1.7104 ± 0.0370	0.3273 ± 0.0016
Beta	80.47 ± 0.14	94.84 ± 0.15	0.0706 ± 0.0003	0.0346 ± 0.0022	0.9038 ± 0.0270	0.1174 ± 0.0010
Isotonic Reg.	80.08 ± 0.19	93.46 ± 0.20	0.0644 ± 0.0014	0.0468 ± 0.0020	1.8375 ± 0.0587	0.1203 ± 0.0012
Platt	80.48 ± 0.14	95.18 ± 0.12	0.0597 ± 0.0007	0.0775 ± 0.0015	0.8083 ± 0.0106	0.1205 ± 0.0010
Vec Scal.	80.53 ± 0.19	95.18 ± 0.16	0.0494 ± 0.0002	0.0300 ± 0.0010	0.8269 ± 0.0097	0.1106 ± 0.0007
Mtx Scal.	**80.78** ± 0.18	**95.38** ± 0.15	0.0508 ± 0.0003	0.0282 ± 0.0014	0.8042 ± 0.0100	0.1090 ± 0.0006
BWS	80.33 ± 0.16	95.10 ± 0.16	0.0561 ± 0.0008	0.044 ± 0.0019	0.8273 ± 0.0105	0.1129 ± 0.0009
ETS-MnM	80.33 ± 0.16	95.10 ± 0.16	0.0479 ± 0.0004	0.0358 ± 0.0009	0.8426 ± 0.0097	0.1115 ± 0.0008
	1k Calibration Samples					
TS	80.33 ± 0.16	95.10 ± 0.16	0.0559 ± 0.0015	0.0439 ± 0.0022	0.8293 ± 0.0107	0.1134 ± 0.0010
GP	80.33 ± 0.15	95.11 ± 0.15	0.0485 ± 0.0035	0.0186 ± 0.0034	**0.7556** ± 0.0118	**0.1069** ± 0.0007
Eq. Mass	5.02 ± 0.13	26.75 ± 0.37	0.0022 ± 0.0001	0.0353 ± 0.0012	3.5272 ± 0.0142	0.0489 ± 0.0012
Eq. Size	80.14 ± 0.23	88.99 ± 0.12	0.1525 ± 0.0023	0.0279 ± 0.0043	1.2671 ± 0.0130	0.1115 ± 0.0011
I-Max	80.20 ± 0.18	94.86 ± 0.17	0.0302 ± 0.0041	0.0200 ± 0.0033	0.7860 ± 0.0208	0.1116 ± 0.0008
Eq. Mass w. TS	5.02 ± 0.13	26.87 ± 0.43	0.0023 ± 0.0001	0.0357 ± 0.0012	3.5454 ± 0.0222	0.0490 ± 0.0012
Eq. Mass w. GP	5.02 ± 0.13	26.87 ± 0.43	0.0022 ± 0.0001	0.0353 ± 0.0012	3.4778 ± 0.0217	0.0489 ± 0.0012
Eq. Size w. TS	80.26 ± 0.18	88.99 ± 0.12	0.1470 ± 0.0007	0.0391 ± 0.0038	1.2721 ± 0.0116	0.1136 ± 0.0012
Eq. Size w. GP	80.26 ± 0.18	88.99 ± 0.12	0.1508 ± 0.0021	0.0140 ± 0.0056	1.2661 ± 0.0121	0.1105 ± 0.0008
I-Max w. TS	80.20 ± 0.18	94.87 ± 0.19	0.0354 ± 0.0124	0.0402 ± 0.0019	0.8339 ± 0.0108	0.1142 ± 0.0009
I-Max w. GP	80.20 ± 0.18	94.87 ± 0.19	**0.0300** ± 0.0041	**0.0121** ± 0.0048	0.7787 ± 0.0102	0.1111 ± 0.0006

Table A11.: Tab. 2 Extension: ImageNet - DenseNet

Calibrator	Acc$_{\text{top1}}$ ↑	Acc$_{\text{top5}}$ ↑	$_{\text{CW}}$ECE$_{\frac{1}{K}}$ ↓	$_{\text{top1}}$ECE ↓	NLL	Brier
Baseline	77.21 ± 0.12	93.51 ± 0.14	0.0502 ± 0.0006	0.0571 ± 0.0014	0.9418 ± 0.0120	0.1228 ± 0.0009
	25k Calibration Samples					
BBQ	54.69 ± 0.42	86.55 ± 0.19	0.0274 ± 0.0007	0.2819 ± 0.0050	1.9805 ± 0.0500	0.3355 ± 0.0026
Beta	77.35 ± 0.22	93.34 ± 0.17	0.0494 ± 0.0008	0.0253 ± 0.0022	0.9768 ± 0.0254	0.1209 ± 0.0010
Isotonic Reg.	76.81 ± 0.24	91.98 ± 0.17	0.0577 ± 0.0003	0.0490 ± 0.0021	1.9819 ± 0.0634	0.1281 ± 0.0012
Platt	77.43 ± 0.21	93.64 ± 0.15	0.0448 ± 0.0010	0.0906 ± 0.0022	0.9168 ± 0.0139	0.1297 ± 0.0012
Vec Scal.	77.44 ± 0.20	93.62 ± 0.17	0.0492 ± 0.0006	0.0516 ± 0.0018	0.9276 ± 0.0134	0.1208 ± 0.0011
Mtx Scal.	**77.56** ± 0.11	**93.81** ± 0.15	0.0498 ± 0.0006	0.0491 ± 0.0015	0.9159 ± 0.0158	0.1202 ± 0.0016
BWS	77.21 ± 0.12	93.51 ± 0.14	0.0395 ± 0.0007	0.0301 ± 0.0012	0.9106 ± 0.0116	0.1197 ± 0.0008
ETS-MnM	77.21 ± 0.12	93.51 ± 0.14	0.0357 ± 0.0008	0.0234 ± 0.0011	0.9188 ± 0.0103	0.1194 ± 0.0006
	1k Calibration Samples					
TS	77.21 ± 0.12	93.51 ± 0.15	0.0375 ± 0.0007	0.0300 ± 0.0019	0.9116 ± 0.0110	0.1197 ± 0.0008
GP	77.22 ± 0.12	93.51 ± 0.13	0.0394 ± 0.0037	0.0268 ± 0.0035	**0.8914** ± 0.0120	**0.1188** ± 0.0005
Eq. Mass	17.21 ± 0.47	45.69 ± 1.22	0.0054 ± 0.0004	0.1572 ± 0.0047	2.9683 ± 0.0561	0.1671 ± 0.0046
Eq. Size	77.06 ± 0.28	88.22 ± 0.10	0.1519 ± 0.0016	0.0230 ± 0.0050	1.3948 ± 0.0105	0.1206 ± 0.0013
I-Max	77.13 ± 0.14	93.34 ± 0.17	0.0263 ± 0.0119	0.0201 ± 0.0088	0.9229 ± 0.0103	0.1201 ± 0.0010
Eq. Mass w. TS	17.21 ± 0.47	45.73 ± 1.07	0.0054 ± 0.0004	0.1571 ± 0.0047	2.9104 ± 0.0482	0.1671 ± 0.0046
Eq. Mass w. GP	17.21 ± 0.47	45.71 ± 1.08	0.0054 ± 0.0004	0.1571 ± 0.0047	2.9090 ± 0.0485	0.1671 ± 0.0046
Eq. Size w. TS	77.19 ± 0.12	88.22 ± 0.10	0.1464 ± 0.0005	0.0241 ± 0.0032	1.3928 ± 0.0106	0.1201 ± 0.0008
Eq. Size w. GP	77.19 ± 0.12	88.22 ± 0.10	0.1527 ± 0.0007	0.0215 ± 0.0037	1.3944 ± 0.0094	0.1200 ± 0.0005
I-Max w. TS	77.13 ± 0.14	93.34 ± 0.17	0.0320 ± 0.0026	0.0245 ± 0.0024	0.9242 ± 0.0117	0.1201 ± 0.0007
I-Max w. GP	77.13 ± 0.14	93.34 ± 0.17	**0.0258** ± 0.0100	**0.0204** ± 0.0021	0.9200 ± 0.0124	0.1201 ± 0.0005

Table A12.: Tab. 2 Extension: ImageNet - ResNet152

Calibrator	Acc$_{\text{top1}}$ ↑	Acc$_{\text{top5}}$ ↑	${}_{\text{CW}}\text{ECE}_{\frac{1}{K}}$ ↓	${}_{\text{top1}}\text{ECE}$ ↓	NLL	Brier
Baseline	78.33 ± 0.17	94.00 ± 0.14	0.05 ± 0.0004	0.0512 ± 0.0018	0.8760 ± 0.0133	0.1174 ± 0.0013
	25k Calibration Samples					
BBQ	55.04 ± 0.26	87.15 ± 0.21	0.0278 ± 0.0004	0.2840 ± 0.0028	1.8490 ± 0.0474	0.3361 ± 0.0014
Beta	78.44 ± 0.16	93.71 ± 0.20	0.0507 ± 0.0012	0.0264 ± 0.0010	0.9365 ± 0.0249	0.1174 ± 0.0013
Isotonic Reg	77.97 ± 0.07	92.33 ± 0.32	0.0590 ± 0.0016	0.0486 ± 0.0027	1.9437 ± 0.1020	0.1248 ± 0.0015
Platt	78.56 ± 0.15	94.06 ± 0.19	0.0458 ± 0.0009	0.0852 ± 0.0021	0.8557 ± 0.0159	0.1246 ± 0.0015
Vec Scal.	**78.61** ± 0.21	94.12 ± 0.18	0.0490 ± 0.0003	0.0469 ± 0.0017	0.8625 ± 0.0143	0.1159 ± 0.0012
Mtx Scal.	78.54 ± 0.23	**94.14** ± 0.22	0.0496 ± 0.0004	0.0443 ± 0.0026	0.8583 ± 0.0180	0.1160 ± 0.0016
BWS	78.33 ± 0.18	94.00 ± 0.15	0.0402 ± 0.0005	0.0277 ± 0.0019	0.8488 ± 0.0127	0.1147 ± 0.0012
ETS-MnM	78.33 ± 0.18	94.00 ± 0.15	0.0366 ± 0.0007	0.0198 ± 0.0006	0.8609 ± 0.0117	0.1145 ± 0.0011
	1k Calibration Samples					
TS	78.33 ± 0.18	94.00 ± 0.15	0.0378 ± 0.0007	0.0285 ± 0.0023	0.8505 ± 0.0126	0.1147 ± 0.0012
GP	78.33 ± 0.17	94.00 ± 0.14	0.0403 ± 0.0021	0.0202 ± 0.0030	**0.8366**± 0.0118	**0.1138** ± 0.0012
Eq. Mass	16.25 ± 0.54	45.53 ± 0.81	0.0064 ± 0.0004	0.1476 ± 0.0054	2.9471 ± 0.0556	0.1579 ± 0.0052
Eq. Size	78.24 ± 0.16	88.81 ± 0.19	0.1480 ± 0.0015	0.0286 ± 0.0053	1.3308 ± 0.0178	0.1167 ± 0.0011
I-Max	78.19 ± 0.21	93.82 ± 0.17	0.0295 ± 0.0030	0.0196 ± 0.0049	0.8638 ± 0.0135	0.1157 ± 0.0012
Eq. Mass w. TS	16.25 ± 0.54	45.54 ± 0.71	0.0064 ± 0.0004	0.1476 ± 0.0054	2.9024 ± 0.0401	0.1579 ± 0.0052
Eq. Mass w. GP	16.25 ± 0.54	45.52 ± 0.74	0.0064 ± 0.0004	0.1475 ± 0.0054	2.9021 ± 0.040	0.1579 ± 0.0052
Eq. Size w. TS	78.27 ± 0.17	88.81 ± 0.19	0.1428 ± 0.0007	0.0225 ± 0.0022	1.3286 ± 0.0171	0.1153 ± 0.0013
Eq. Size w. GP	78.27 ± 0.17	88.81 ± 0.19	0.1475 ± 0.0016	0.0138 ± 0.0049	1.330 ± 0.0171	0.1150 ± 0.0012
I-Max w. TS	78.19 ± 0.21	93.82 ± 0.17	**0.0281** ± 0.0029	0.0219 ± 0.0016	0.8637 ± 0.0125	0.1152 ± 0.0015
I-Max w. GP	78.19 ± 0.21	93.82 ± 0.17	0.0296 ± 0.0029	**0.0144** ± 0.0050	0.8602 ± 0.0127	0.1150 ± 0.0014

Table A13.: Tab. 2 Extension: CIFAR100 - WRN

Calibrator	Acc$_{\text{top1}}$ ↑	${}_{\text{CW}}\text{ECE}_{\frac{1}{K}}$ ↓	${}_{\text{top1}}\text{ECE}$ ↓	NLL	Brier
Baseline	81.35 ± 0.13	0.1113 ± 0.0010	0.0748 ± 0.0018	0.7816 ± 0.0076	0.1082 ± 0.0021
	5k Calibration Samples				
BBQ	80.44 ± 0.19	0.0576 ± 0.0018	0.0672 ± 0.0044	1.7976 ± 0.0443	0.1297 ± 0.0019
Beta	81.44 ± 0.17	0.0952 ± 0.0006	0.0379 ± 0.0027	0.7624 ± 0.0148	0.1018 ± 0.0016
Isotonic Reg.	81.25 ± 0.27	0.0597 ± 0.0029	0.0487 ± 0.0040	1.4015 ± 0.0748	0.1059 ± 0.0013
Platt	81.35 ± 0.12	0.0827 ± 0.0014	0.0585 ± 0.0038	0.7491 ± 0.0073	0.1026 ± 0.0017
Vec Scal.	81.35 ± 0.21	0.1063 ± 0.0013	0.0687 ± 0.0029	0.7619 ± 0.0064	0.1055 ± 0.0017
Mtx Scal.	**81.44** ± 0.20	0.1085 ± 0.0008	0.0692 ± 0.0033	0.7531 ± 0.0078	0.1059 ± 0.0019
BWS	81.35 ± 0.14	0.1069 ± 0.0009	0.0451 ± 0.0028	0.737 ± 0.0057	0.1037 ± 0.0017
ETS-MnM	81.35 ± 0.14	0.0976 ± 0.0019	0.0451 ± 0.0027	0.7695 ± 0.0052	0.1027 ± 0.0020
	1k Calibration Samples				
TS	81.35 ± 0.14	0.0911 ± 0.0036	0.0511 ± 0.0059	0.7527 ± 0.0074	0.1036 ± 0.0025
GP	81.34 ± 0.12	0.1074 ± 0.0043	0.0358 ± 0.0039	**0.6943** ± 0.0025	**0.0996** ± 0.0019
Eq. Mass	62.04 ± 0.53	0.0252 ± 0.0032	0.4744 ± 0.0049	1.1789 ± 0.0308	0.4606 ± 0.0034
Eq. Size	81.12 ± 0.15	0.1229 ± 0.0030	0.0273 ± 0.0055	1.0165 ± 0.0105	0.1039 ± 0.0017
I-Max	81.30 ± 0.22	0.0518 ± 0.0036	0.0231 ± 0.0067	0.7593 ± 0.0085	0.1016 ± 0.0018
Eq. Mass w. TS	62.04 ± 0.53	0.0253 ± 0.0034	0.4764 ± 0.0052	1.0990 ± 0.0184	0.4624 ± 0.0037
Eq. Mass w. GP	62.04 ± 0.53	0.0252 ± 0.0032	0.4749 ± 0.0051	1.1110 ± 0.0226	0.4610 ± 0.0036
Eq. Size w. TS	81.31 ± 0.15	0.1197 ± 0.0029	0.0362 ± 0.0065	1.0106 ± 0.0113	0.1038 ± 0.0026
Eq. Size w. GP	81.31 ± 0.15	0.1205 ± 0.0025	0.0189 ± 0.0054	1.0161 ± 0.0115	0.1032 ± 0.0020
I-Max w. TS	81.34 ± 0.20	**0.051** ± 0.0035	0.0365 ± 0.0067	0.7716 ± 0.0066	0.1025 ± 0.0021
I-Max w. GP	81.34 ± 0.20	0.0559 ± 0.0089	**0.0179** ± 0.0046	0.7609 ± 0.0080	0.1014 ± 0.0014

Table A14.: Tab. 2 Extension: CIFAR100 - ResNeXt

Calibrator	Acc_{top1} ↑	$_{\text{CW}}\text{ECE}_{\frac{1}{K}}$ ↓	$_{\text{top1}}\text{ECE}$ ↓	NLL	Brier
Baseline	81.93 ± 0.08	0.0979 ± 0.0015	0.0590 ± 0.0028	0.7271 ± 0.0026	0.0984 ± 0.0022
	5k Calibration Samples				
BBQ	81.06 ± 0.30	0.0564 ± 0.0013	0.0608 ± 0.0058	1.6878 ± 0.0546	0.1176 ± 0.0022
Beta	82.19 ± 0.31	0.0918 ± 0.0020	0.0368 ± 0.0047	0.7095 ± 0.0074	0.0947 ± 0.0024
Isotonic Reg.	81.89 ± 0.19	0.0619 ± 0.0023	0.0503 ± 0.0036	1.3015 ± 0.0656	0.0995 ± 0.0018
Platt	82.28 ± 0.21	0.0790 ± 0.0025	0.0534 ± 0.0047	0.7050 ± 0.0045	0.0961 ± 0.0026
Vec Scal.	82.24 ± 0.27	0.0963 ± 0.0013	0.0572 ± 0.0037	0.7129 ± 0.0053	0.0973 ± 0.0021
Mtx Scal.	**82.38** ± 0.17	0.0970 ± 0.0014	0.0578 ± 0.0040	0.7042 ± 0.0046	0.0973 ± 0.0023
BWS	81.93 ± 0.08	0.1045 ± 0.0015	0.0448 ± 0.0044	0.6897 ± 0.0031	0.0969 ± 0.0017
ETS-MnM	81.93 ± 0.08	0.0932 ± 0.0020	0.0460 ± 0.001	0.7284 ± 0.0029	0.0963 ± 0.0022
	1k Calibration Samples				
TS	81.93 ± 0.08	0.0864 ± 0.0036	0.0525 ± 0.0057	0.7163 ± 0.0037	0.0975 ± 0.0020
GP	81.93 ± 0.09	0.1025 ± 0.0037	0.0345 ± 0.0038	**0.6456** ± 0.0071	**0.0927** ± 0.0019
Eq. Mass	64.48 ± 0.64	0.0265 ± 0.0011	0.4980 ± 0.0070	1.1232 ± 0.0277	0.4770 ± 0.0051
Eq. Size	81.99 ± 0.21	0.1195 ± 0.0013	0.0230 ± 0.0033	0.9556 ± 0.0071	0.0974 ± 0.0014
I-Max	81.96 ± 0.14	0.0549 ± 0.0081	0.0205 ± 0.0074	0.7127 ± 0.0040	0.0959 ± 0.0018
Eq. Mass w. TS	64.48 ± 0.64	0.0262 ± 0.0013	0.5003 ± 0.0066	1.0468 ± 0.0228	0.4793 ± 0.0048
Eq. Mass w. GP	64.48 ± 0.64	0.0264 ± 0.0012	0.4986 ± 0.0066	1.0555 ± 0.0227	0.4776 ± 0.0048
Eq. Size w. TS	81.94 ± 0.09	0.1179 ± 0.0015	0.0343 ± 0.0029	0.9498 ± 0.0058	0.0968 ± 0.0022
Eq. Size w. GP	81.94 ± 0.09	0.1177 ± 0.0009	0.0151 ± 0.0029	0.9561 ± 0.0056	0.0959 ± 0.0018
I-Max w. TS	81.96 ± 0.14	**0.053** ± 0.0073	0.0333 ± 0.0023	0.7286 ± 0.0029	0.0964 ± 0.0019
I-Max w. GP	81.96 ± 0.14	0.0532 ± 0.0077	**0.0121** ± 0.0026	0.7111 ± 0.0024	0.0950 ± 0.0017

Table A15.: Tab. 2 Extension Dataset: CIFAR100 - DenseNet

Calibrator	Acc_{top1} ↑	$_{\text{CW}}\text{ECE}_{\frac{1}{K}}$ ↓	$_{\text{top1}}\text{ECE}$ ↓	NLL	Brier
Baseline	82.36 ± 0.26	0.1223 ± 0.0008	0.0762 ± 0.0015	0.7542 ± 0.0143	0.1041 ± 0.0008
	5k Calibration Samples				
BBQ	81.56 ± 0.22	0.0567 ± 0.0020	0.0635 ± 0.0052	1.5876 ± 0.0914	0.1216 ± 0.0026
Beta	82.39 ± 0.28	0.0953 ± 0.0013	0.0364 ± 0.0034	0.6935 ± 0.0185	0.0966 ± 0.0008
Isotonic Reg.	82.05 ± 0.26	0.0591 ± 0.0016	0.0506 ± 0.0025	1.3030 ± 0.1107	0.1019 ± 0.0014
Platt	82.34 ± 0.28	0.0866 ± 0.0012	0.0491 ± 0.0012	0.6835 ± 0.0138	0.0969 ± 0.0015
Vec Scal.	82.38 ± 0.32	0.1195 ± 0.0005	0.0711 ± 0.0015	0.7362 ± 0.0173	0.1028 ± 0.0015
Mtx Scal.	**82.53** ± 0.19	0.1214 ± 0.0006	0.0733 ± 0.0013	0.7360 ± 0.0153	0.1025 ± 0.0015
BWS	82.36 ± 0.27	0.1028 ± 0.0013	0.0445 ± 0.0021	0.682 ± 0.0125	0.0975 ± 0.0008
ETS-MnM	82.36 ± 0.27	0.1007 ± 0.0016	0.0387 ± 0.0012	0.6986 ± 0.0111	0.0969 ± 0.0008
	1k Calibration Samples				
TS	82.36 ± 0.27	0.0938 ± 0.0017	0.0447 ± 0.0023	0.6851 ± 0.0115	0.0976 ± 0.0008
GP	82.35 ± 0.27	0.1021 ± 0.0032	0.0338 ± 0.0011	**0.6536** ± 0.0120	**0.0943** ± 0.0007
Eq. Mass	58.11 ± 0.21	0.0233 ± 0.0005	0.4339 ± 0.0024	1.2049 ± 0.0405	0.4317 ± 0.0017
Eq. Size	82.22 ± 0.30	0.1192 ± 0.0024	0.0219 ± 0.0021	0.9482 ± 0.0137	0.0997 ± 0.0014
I-Max	82.32 ± 0.22	0.0546 ± 0.0122	0.0189 ± 0.0071	0.7022 ± 0.0124	0.0967 ± 0.0019
Eq. Mass w. TS	58.11 ± 0.21	0.0233 ± 0.0006	0.4347 ± 0.0024	1.1483 ± 0.0102	0.4324 ± 0.0017
Eq. Mass w. GP	58.11 ± 0.21	0.0233 ± 0.0005	0.4342 ± 0.0024	1.1508 ± 0.0099	0.4319 ± 0.0018
Eq. Size w. TS	82.40 ± 0.24	0.1134 ± 0.0014	0.0245 ± 0.0025	0.9427 ± 0.0137	0.0986 ± 0.0013
Eq. Size w. GP	82.40 ± 0.24	0.1166 ± 0.0021	0.0126 ± 0.0012	0.9455 ± 0.0142	0.0985 ± 0.0013
I-Max w. TS	82.36 ± 0.21	**0.048** ± 0.0090	0.0237 ± 0.0009	0.7040 ± 0.0104	0.0967 ± 0.0010
I-Max w. GP	82.36 ± 0.21	0.0535 ± 0.0121	**0.0114** ± 0.0025	0.6988 ± 0.0104	0.0964 ± 0.0010

Table A16.: Tab. 2 Extension: CIFAR10 - WRN

Calibrator	Acc_{top1} ↑	${}_{\text{CW}}\text{ECE}_{\frac{1}{K}}$ ↓	${}_{\text{top1}}\text{ECE}$ ↓	NLL	Brier
Baseline	96.12 ± 0.14	0.0457 ± 0.0011	0.0288 ± 0.0007	0.1682 ± 0.0062	0.0307 ± 0.0008
	5k Calibration Samples				
BBQ	95.98 ± 0.15	0.0290 ± 0.0047	0.0198 ± 0.0044	0.2054 ± 0.0156	0.0314 ± 0.0005
Beta	96.31 ± 0.06	0.0504 ± 0.0015	0.0208 ± 0.0023	0.1335 ± 0.0039	0.0271 ± 0.0007
Isotonic Reg.	96.20 ± 0.12	0.0241 ± 0.0021	0.0138 ± 0.0017	0.1764 ± 0.0241	0.0273 ± 0.0005
Platt	96.24 ± 0.09	0.0489 ± 0.0011	0.0177 ± 0.0015	0.1359 ± 0.0039	0.0270 ± 0.0006
Vec Scal.	**96.27** ± 0.11	0.0449 ± 0.0008	0.0229 ± 0.0008	0.1437 ± 0.0050	0.0286 ± 0.0007
Mtx Scal.	96.20 ± 0.10	0.0444 ± 0.0005	0.0277 ± 0.0007	0.1625 ± 0.0062	0.0302 ± 0.0008
BWS	96.12 ± 0.14	0.0467 ± 0.0012	0.0195 ± 0.0014	0.1395 ± 0.0077	0.0279 ± 0.0007
ETS-MnM	96.12 ± 0.14	0.0647 ± 0.0014	0.0329 ± 0.0012	0.1478 ± 0.0038	0.0270 ± 0.0006
	1k Calibration Samples				
TS	96.12 ± 0.14	0.0486 ± 0.0024	0.0205 ± 0.0009	0.1385 ± 0.0048	0.0278 ± 0.0007
GP	96.10 ± 0.13	0.0549 ± 0.0021	0.0146 ± 0.0022	**0.1281** ± 0.0055	0.0269 ± 0.0009
Eq. Mass	91.24 ± 0.27	0.0212 ± 0.0009	0.0836 ± 0.0091	0.2252 ± 0.0220	0.0858 ± 0.0055
Eq. Size	96.04 ± 0.15	0.0278 ± 0.0021	0.0105 ± 0.0028	0.2744 ± 0.0812	0.0305 ± 0.0015
I-Max	96.06 ± 0.13	0.0304 ± 0.0012	0.0113 ± 0.0039	0.1595 ± 0.0604	0.0274 ± 0.0013
Eq. Mass w. TS	91.24 ± 0.27	0.0219 ± 0.0005	0.0837 ± 0.0092	0.1944 ± 0.0093	0.0853 ± 0.0054
Eq. Mass w. GP	91.24 ± 0.27	0.0212 ± 0.0008	0.0821 ± 0.0088	0.1918 ± 0.0091	0.0851 ± 0.0054
Eq. Size w. TS	96.13 ± 0.12	0.0286 ± 0.0018	0.0125 ± 0.0024	0.1940 ± 0.0063	0.0296 ± 0.0009
Eq. Size w. GP	96.13 ± 0.11	**0.0266** ± 0.0016	**0.0066** ± 0.0028	0.1917 ± 0.0058	0.0292 ± 0.0009
I-Max w. TS	96.14 ± 0.13	0.0293 ± 0.0010	0.0163 ± 0.0012	0.1417 ± 0.0047	0.0280 ± 0.0008
I-Max w. GP	96.14 ± 0.13	0.0276 ± 0.0011	0.0074 ± 0.0035	0.1331 ± 0.0042	**0.0268** ± 0.0008

Table A17.: Tab. 2 Extension: CIFAR10 - ResNeXt

Calibrator	Acc_{top1} ↑	${}_{\text{CW}}\text{ECE}_{\frac{1}{K}}$ ↓	${}_{\text{top1}}\text{ECE}$ ↓	NLL	Brier
Baseline	96.30 ± 0.18	0.0485 ± 0.0014	0.0201 ± 0.0021	0.1247 ± 0.0058	0.0266 ± 0.0013
	5k Calibration Samples				
BBQ	96.18 ± 0.12	0.0256 ± 0.0027	0.0166 ± 0.0020	0.1951 ± 0.0134	0.0286 ± 0.0004
Beta	96.31 ± 0.22	0.0517 ± 0.0011	0.0148 ± 0.0016	0.1163 ± 0.0040	0.0256 ± 0.0011
Isotonic Reg.	96.35 ± 0.20	0.0241 ± 0.0016	0.0129 ± 0.0008	0.1686 ± 0.0099	0.0264 ± 0.0011
Platt	96.34 ± 0.19	0.0511 ± 0.0008	0.0143 ± 0.0017	0.1159 ± 0.0042	0.0256 ± 0.0011
Vec Scal.	**96.37** ± 0.19	0.0495 ± 0.0017	0.0161 ± 0.0017	0.1189 ± 0.0053	0.0258 ± 0.0013
Mtx Scal.	96.34 ± 0.21	0.0492 ± 0.0020	0.0187 ± 0.0020	0.1225 ± 0.0060	0.0263 ± 0.0014
BWS	96.3 ± 0.19	0.0514 ± 0.0013	0.015 ± 0.0008	0.1199 ± 0.0048	0.0257 ± 0.0012
ETS-MnM	96.3 ± 0.19	0.0547 ± 0.0013	0.0159 ± 0.0027	0.1193 ± 0.0043	0.0257 ± 0.0011
	1k Calibration Samples				
TS	96.30 ± 0.19	0.0524 ± 0.0028	0.0150 ± 0.0009	0.1182 ± 0.0051	0.0257 ± 0.0012
GP	96.31 ± 0.17	0.0529 ± 0.0017	0.0125 ± 0.0021	**0.1176** ± 0.0051	**0.0258** ± 0.0011
Eq. Mass	89.85 ± 0.61	0.0269 ± 0.0051	0.0676 ± 0.0127	0.2208 ± 0.0172	0.0841 ± 0.0042
Eq. Size	96.17 ± 0.24	0.0288 ± 0.0039	0.0114 ± 0.0025	0.2495 ± 0.0571	0.0277 ± 0.0008
I-Max	96.22 ± 0.21	0.0254 ± 0.0030	0.0104 ± 0.0025	0.1397 ± 0.0276	0.0265 ± 0.0012
Eq. Mass w. TS	89.85 ± 0.61	0.0269 ± 0.0054	0.0676 ± 0.0128	0.1966 ± 0.0104	0.0844 ± 0.0043
Eq. Mass w. GP	89.85 ± 0.61	0.0266 ± 0.0049	0.0669 ± 0.0126	0.1962 ± 0.0106	0.0841 ± 0.0043
Eq. Size w. TS	96.29 ± 0.18	0.0270 ± 0.0022	0.0062 ± 0.0024	0.1574 ± 0.0091	0.0264 ± 0.0013
Eq. Size w. GP	96.29 ± 0.18	0.0271 ± 0.0020	0.0063 ± 0.0030	0.1576 ± 0.0093	0.0264 ± 0.0012
I-Max w. TS	96.28 ± 0.19	0.0224 ± 0.0016	0.0053 ± 0.0024	0.1208 ± 0.0058	0.0259 ± 0.0012
I-Max w. GP	96.28 ± 0.19	**0.0223** ± 0.0018	**0.0052** ± 0.0029	0.1206 ± 0.0061	0.0259 ± 0.0012

Table A18.: Tab. 2 Extension: CIFAR10 - DenseNet

Calibrator	Acc_{top1} ↑	$_{\text{CW}}\text{ECE}_{\frac{1}{K}}$ ↓	$_{\text{top1}}\text{ECE}$ ↓	NLL	Brier
Baseline	96.65 ± 0.09	0.0404 ± 0.0010	0.0253 ± 0.0009	0.1564 ± 0.0075	0.0259 ± 0.0007
	5k Calibration Samples				
BBQ	96.75 ± 0.19	0.0245 ± 0.0030	0.0170 ± 0.0022	0.1806 ± 0.0105	0.0279 ± 0.0010
Beta	96.81 ± 0.10	0.0468 ± 0.0003	0.0154 ± 0.0013	0.1151 ± 0.0042	0.0234 ± 0.0007
Isotonic Reg.	96.84 ± 0.08	0.0236 ± 0.0022	0.0140 ± 0.0024	0.1501 ± 0.0137	0.0241 ± 0.0007
Platt	96.82 ± 0.11	0.0459 ± 0.0007	0.0141 ± 0.0010	0.1154 ± 0.0040	0.0233 ± 0.0007
Vec Scal.	**96.84** ± 0.14	0.0413 ± 0.0014	0.0223 ± 0.0010	0.1373 ± 0.0077	0.0249 ± 0.0007
Mtx Scal.	96.73 ± 0.09	0.0402 ± 0.0017	0.0245 ± 0.0008	0.1531 ± 0.0081	0.0257 ± 0.0007
BWS	96.65 ± 0.10	0.0423 ± 0.0010	0.0188 ± 0.0016	0.1239 ± 0.0065	0.0239 ± 0.0006
ETS-MnM	96.65 ± 0.10	0.0527 ± 0.0012	0.0212 ± 0.0012	0.1196 ± 0.0038	0.0230 ± 0.0007
	1k Calibration Samples				
TS	96.65 ± 0.10	0.0425 ± 0.0005	0.0169 ± 0.0010	0.1186 ± 0.0051	0.0237 ± 0.0006
GP	96.66 ± 0.09	0.0490 ± 0.0022	0.0135 ± 0.0025	**0.1143** ± 0.0048	**0.0228** ± 0.0007
Eq. Mass	88.80 ± 0.47	0.0233 ± 0.0024	0.0637 ± 0.0023	0.2694 ± 0.0274	0.0881 ± 0.0033
Eq. Size	96.64 ± 0.22	0.0262 ± 0.0035	0.0101 ± 0.0035	0.2465 ± 0.0543	0.0256 ± 0.0003
I-Max	96.59 ± 0.32	0.0261 ± 0.0025	0.0098 ± 0.0027	0.1208 ± 0.0044	0.0239 ± 0.0005
Eq. Mass w. TS	88.80 ± 0.47	0.0234 ± 0.0026	0.0626 ± 0.0023	0.2102 ± 0.0051	0.0877 ± 0.0030
Eq. Mass w. GP	88.80 ± 0.47	0.0233 ± 0.0026	0.0634 ± 0.0025	0.2098 ± 0.0053	0.0880 ± 0.0030
Eq. Size w. TS	96.75 ± 0.10	0.0250 ± 0.0011	0.0133 ± 0.0014	0.1657 ± 0.0056	0.0249 ± 0.0007
Eq. Size w. GP	96.77 ± 0.10	0.0242 ± 0.0022	0.0050 ± 0.0012	0.1612 ± 0.0048	0.0245 ± 0.0005
I-Max w. TS	96.81 ± 0.15	0.0229 ± 0.0016	0.0125 ± 0.0017	0.1224 ± 0.0056	0.0239 ± 0.0007
I-Max w. GP	96.81 ± 0.15	**0.0218** ± 0.0012	**0.0048** ± 0.0009	0.1173 ± 0.0054	0.0231 ± 0.0005

Chapter 14.

Appendix: On-manifold adversarial data augmentation improves uncertainty calibration (OMADA)

B1. Visualizing other OMADA attack paths

Fig. B1 depicts more examples of attack paths, with different start and end targets, produced by the presented method. The OMADA attack path examples include paths where the target is set to another class (e.g. blue path with target “2"), as well as paths where the target is a decision boundary (e.g. green path with target between “1" and “8" and red path with target between “0" and “2"). The decision boundary between two classes can be reached by setting the target vector (y_i^o) in Eq. 8.1 to 0.5 for the two classes and 0 elsewhere. It can be seen that the images produced by the decision boundary paths produce confusing samples which reflect features from the neighboring clusters. Furthermore, this confusion is reflected in the soft-label.

B2. Input mixup example

In Fig. B1, the Input Mixup projection path is visualized in magenta. This path is produced by projecting the linearly interpolated images produced by Input Mixup into the latent space using the encoder. Even though Mixup mainly produces unrealistic images (Fig. 8.1), when it does produce realistic samples from another class, the soft label would not reflect the presence of this class. For example, in Fig. B1, Input Mixup produces an interpolated image between the classes “5" and “2" which looks similar to an “8". It can be seen that Input Mixup assigns zero probability for class “8", whereas using our encoder the images get mapped to the “8" cluster, which means a soft label produced by our method would reflect the presence of the class “8" .

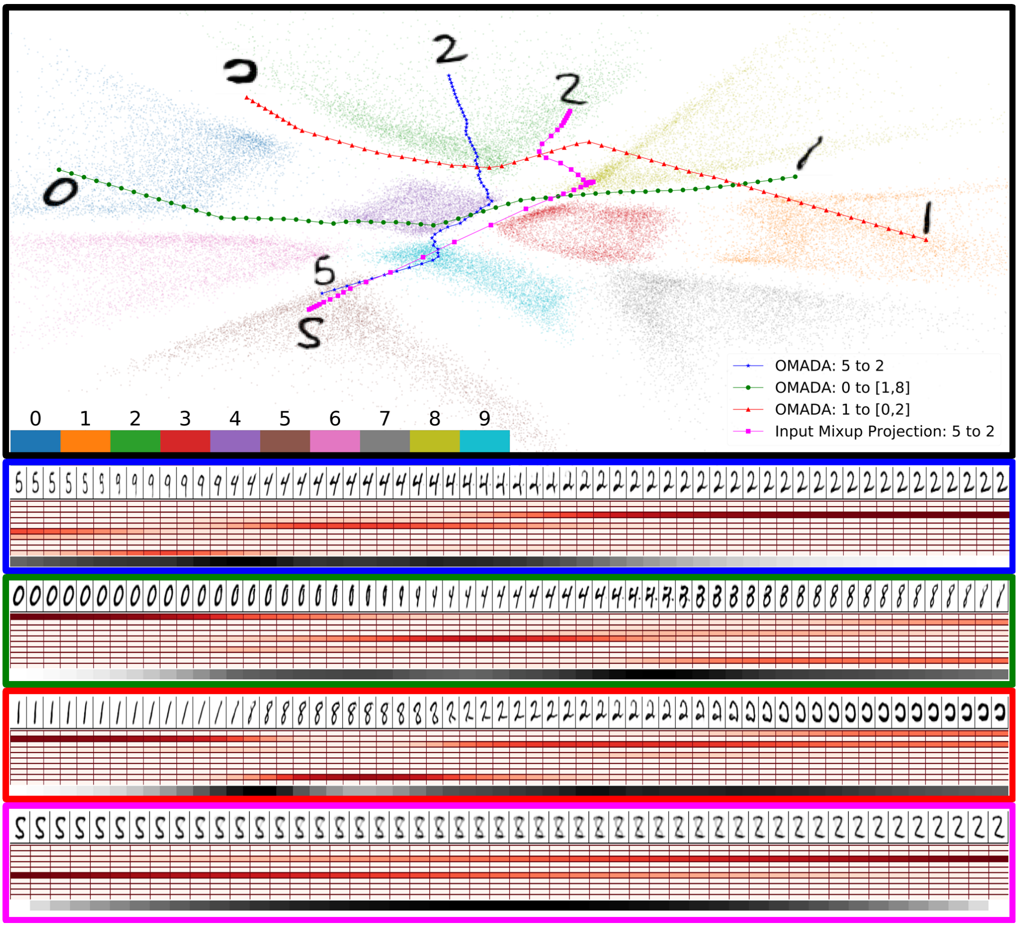


Figure B1.: Visualization of an MNIST encoder-decoder latent space with multiple trajectories traversing through the latent space. The paths depict 3 On-manifold adversarial attack paths, as well as 1 Input Mixup projection into the same latent space. Below the latent space we visualize the decoded image path for OMADA (top 3 blocks) and the Input-Mixup images (bottom block) along with their corresponding soft labels (10 rows below images, red intensity corresponds to likelihood for classes 0 to 9), and the class entropy (bottom row, black shows high entropy). The green and red paths are generated when setting the target as a soft-label between two classes (targeting specifically the decision boundaries). For example, the green path starts at cluster “0" and optimizes Eq. 8.1 with the target (y_i^o) set to a soft label with 0.5 for classes 1 and 8, and 0 elsewhere. As a result, this produces perturbations which direct the path to the decision boundary between the classes 1 and 8. The magenta path shows the projection of Input Mixup images between samples “5" and “2". It can be seen the OMADA paths produce mostly confusing samples at the decision boundaries, and that the soft labels reflect this confusion, whereas Input Mixup produces images which resemble an “8" (seen in the image path below as well as the magenta projection path going to the “8" cluster first before heading towards the target cluster ‘2"); Mixup’s soft label does not reflect this in its soft-label (soft-label is zero at class “8").

B3. Experiment hyperparameters

This section will present detailed information regarding the training process.

B3.1. Model and training hyper-parameters

All optimizer training hyper-parameters for the training of the image-space classifiers can be found in Table B1. These parameters are kept unchanged across the three datasets and all methods, as well as across all 5 repetitions (where only the random seed was changed). We use the default training hyper-parameters of the BigBi-GAN model. For the latent-space model we use a simple 4-layer neural network, each with 1024 units and ReLU non-linear activations. We use a latent space dimension of 128 for all latent-space classifiers.

B3.2. OMADA training hyper-parameters

Each OMADA-trained network uses a balanced 50% of real training samples (with hard one-hot labels) and 50% of the On-manifold adversarial samples in each *batch*. In order to enable direct comparison to alternative methods in the literature, we ensure that for each epoch, the total numbers of gradient updates performed are the same with the balanced number of samples from both datasets. Therefore, at the end of each epoch, 50% of the real training samples are not seen and instead replaced by On-manifold adversarial samples. It should be noted that the 50% real samples seen during each epoch vary across epochs. In order to speed up the training process, we create an offline On-manifold adversarial dataset, and sample from this dataset to fill up each batch during training.
For all networks, 1K random training samples are withheld to create a validation set. The validation set accuracy is used for early stopping, and all experiments (unless stated otherwise) report the results from the checkpoint with the highest validation accuracy during training. Furthermore, the validation set is used to find the best temperature to produce the temperature scaling results.

B3.3. Details about literature methods

This sub-section reports the hyper-parameters of the alternative methods in the literature in more detail.

1. **Base**: base network trained using only real samples with hard labels and no data augmentation.

Model	Num Params.	Weight Decay	Epochs	LR Scheduler	Milestones	LR Decay	Batch
DenseNet-100-12	$796,162$	1×10^{-4}	300	Multi-step	[150,255]	0.1	64
WRN-28-10	$36,479,194$	5×10^{-4}	200	Multi-step	[60,120,160]	0.1	128
VGG-16	$33,646,666$	1×10^{-4}	160	Multi-step	[80,120]	0.1	128
ResNeXt-29	$34,426,698$	5×10^{-4}	300	Multi-step	[150,255]	0.1	128

Table B1.: Training hyper-parameters of image-space classifiers. We use SGD with a base learning rate of 0.1 and momentum of 0.9 for all trainings.

2. **Mixup**: mixup training [Zhang et al., 2018] with $\alpha = 0.1$ [Thulasidasan et al., 2019]. Augments the training dataset by linearly interpolating between both images and labels within a mini batch.

3. **Manifold Mixup**: extends mixup training by taking linear interpolations of hidden layers in the network and hard labels. We use $\alpha = 2.0$ [Verma et al., 2019].

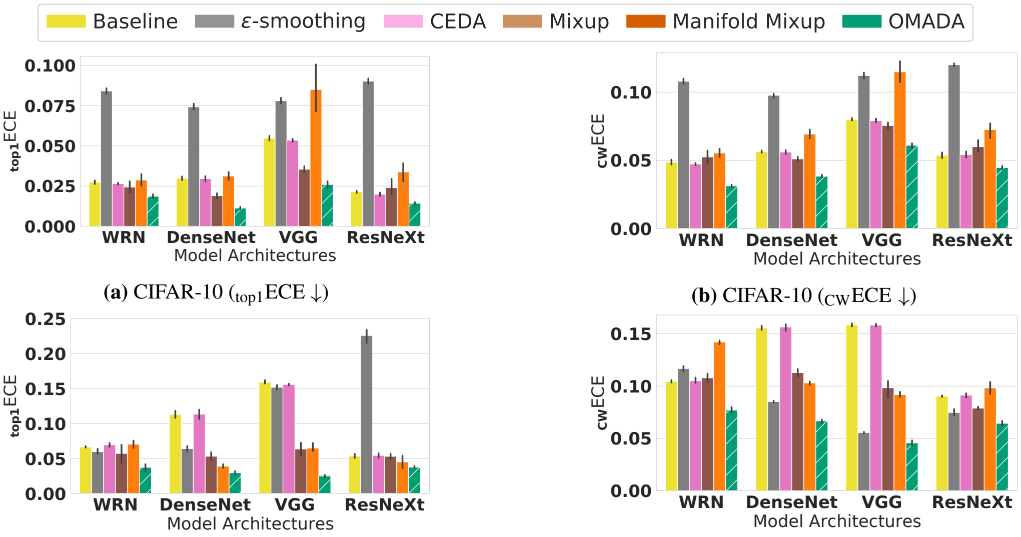
4. **ϵ-smoothing**: smooths the labels with $\epsilon = 0.1$ (found to be best in [Thulasidasan et al., 2019]) by taking a linear combination of the $(1-\epsilon) \times$ hard-label and $\epsilon \times$ uniform-class-label.

5. **CEDA**: confidence enhancing data augmentation (CEDA) is a training scheme that enforces uniform confidences on out-of-distribution noise. These out-of-distribution images are included into the training by replacing half of the batch of real samples with 25% permuted pixel images and 25% uniform random noise images. For each of these augmented images, a Gaussian filter with standard deviation $\sigma \in [1.0, 2.5]$ [Hein et al., 2019] is applied on the images, to have more low-frequency structure in the noise. The label for each of these images is the uniform class label.

B4. Additional results

B4.1. Standard augmentation baseline

We visualize the calibration performance using a standard augmentation baseline in Fig. B2 in the same setting as Sec. 8.6, except that we now include standard augmentations into all training procedures. The standard augmentations include data augmentation on the training samples (random crop with padding 4) and horizontal flips (flip probability 0.50). The results are consistent with Fig. 8.3 and show that OMADA can easily be combined with alternative generalization techniques, while maintaining the best calibration performance compared to all other methods. Interestingly, comparing the observations from the two different baselines, it can be seen that even though standard augmentation improved the accuracy of all methods, the calibration performance has not always stayed the same. Some methods have significantly higher ECEs, for example, CEDA and Mixup on CIFAR-100 for DenseNet and Manifold Mixup on CIFAR-100 for WRN. This shows that standard augmentation on its own strongly influences the calibration quality of the networks, thus making it harder to isolate the effect of each calibration method. Nonetheless, OMADA did not compromise its original calibration gains when using standard augmentation for improving accuracy. Standard augmentations like crops and random flips are specific to images, therefore the comparison of calibration gains in both the base and the augmented setup is important to highlight the strengths of each technique for potential future applications in non-image domains (e.g. text or point clouds). In summary, for in-distribution samples, OMADA results in well-calibrated, accurate classifiers across all diverse network architectures and datasets, especially in comparison to competing label smoothing/data augmentation approaches.



(a) CIFAR-10 ($_{\text{top1}}$ECE ↓)

(b) CIFAR-10 ($_{\text{CW}}$ECE ↓)

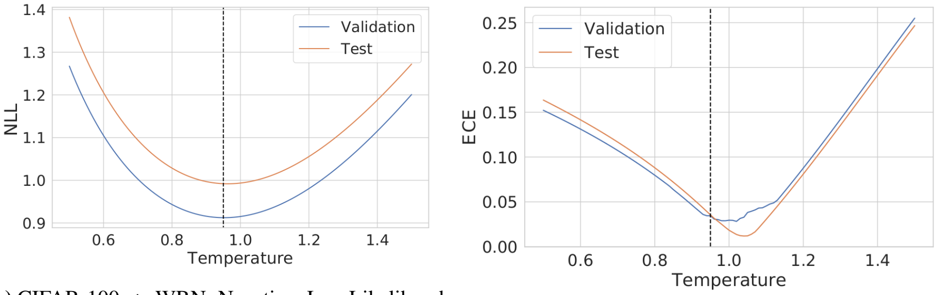
(c) CIFAR-100 ($_{\text{top1}}$ECE ↓)

(d) CIFAR-100 ($_{\text{CW}}$ECE ↓)

Figure B2.: $_{\text{top1}}$ECEand $_{\text{CW}}$ECEcalibration performance of label-smoothing methods using standard augmentations (horizontal flips and random crops) during training on in-distribution test dataset (a-b) CIFAR-10 and (c-d) CIFAR-100. Similar to previous observations, OMADA results in a lower ECE than all other calibration methods. This shows that OMADA can be combined with orthogonal generalization techniques to improve accuracy and at the same time yields better calibration. Hatched bars indicate the best-performing method per model architecture. Error bars are ± 1 std. dev. over 5 runs.

B4.2. When does temperature scaling help?

Temperature scaling is a simple method for improving network calibration. Interestingly, we observe that temperature scaling does not always improve performance; for classifiers which are already fairly well calibrated, the ECE becomes worse after applying temperature scaling. This suggests that the negative log likelihood (NLL) optimized by temperature scaling does not always correlate with a lower ECE. We show this phenomenon for WRN on CIFAR-100, where the optimized temperature increased the calibration error (ECE). Fig. B3 shows the NLL and ECE when performing a grid-search across temperatures. It can be seen that the best temperature (T=0.952) based on the validation NLL (vertical dashed black line) does not minimize the ECE on the test nor the validation set.



(a) CIFAR-100 + WRN Negative Log-Likelihood (NLL)

(b) CIFAR-100 + WRN $_{\text{top1}}$ECE

Figure B3.: The figure depicts the (a) NLL and (b) $_{\text{top1}}$ECEvalues when performing a grid-search for finding the best temperature T on the validation and test sets for CIFAR-100 on WRN. The vertical dashed black line shows the chosen temperature (T=0.952) based on the lowest NLL on the validation set (i.e the optimized temperature). It can be seen that for both validation and test sets, the optimized temperatures do not minimize the $_{\text{top1}}$ECE. In this case, in can be seen that not applying temperature scaling (T=1) would give a lower $_{\text{top1}}$ECE.

Bibliography

Abdulatif, S., Wei, Q., Aziz, F., Kleiner, B., & Schneider, U. (2018). Micro-doppler based human-robot classification using ensemble and deep learning approaches. *IEEE Radar Conference*, (pp. 1043–1048).

Angelov, A., Robertson, A., Murray-Smith, R., & Fioranelli, F. (2018). Practical classification of different moving targets using automotive radar and deep neural networks. *IET Radar, Sonar & Navigation*, *12*(10), 1082–1089.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., & Lacoste-Julien, S. (2017). A closer look at memorization in deep networks.

Arthur, D., & Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. In *Proceedings of the ACM-SIAM symposium on Discrete algorithms*, (pp. 1027–1035). Philadelphia, PA, USA.

Ashukha, A., Lyzhov, A., Molchanov, D., & Vetrov, D. (2020). Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *arXiv*.

Bagloee, S. A., Tavana, M., Asadi, M., & Oliver, T. (2016). Autonomous vehicles: challenges, opportunities, and future implications for transportation policies. *Journal of modern transportation*, *24*(4), 284–303.

Bartsch, A., Fitzek, F., & Rasshofer, R. (2012). Pedestrian recognition using automotive radar sensors. *Advances in Radio Science*, *10*(B. 2), 45–55.

Bhattacharya, A., & Vaughan, R. (2020). Deep learning radar design for breathing and fall detection. *IEEE Sensors Journal*, *20*(9), 5072–5085.

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural networks. In *International Conference on Machine Learning (ICML)*, (pp. 1613–1622). Lille, France.

Brach, K., Sick, B., & Dürr, O. (2020). Single shot mc dropout approximation.

Brooks, D. A., Schwander, O., Barbaresco, F., Schneider, J.-Y., & Cord, M. (2018). Temporal deep learning for drone micro-doppler classification. In *2018 19th International Radar Symposium (IRS)*, (pp. 1–10). IEEE.

Bystrov, A., Hoare, E., Tran, T.-Y., Clarke, N., Gashinova, M., & Cherniakov, M. (2016). Road surface classification using automotive ultrasonic sensor. *Procedia Engineering*, *168*, 19–22.

Chapelle, O., Weston, J., Bottou, L., & Vapnik, V. (2001). Vicinal risk minimization. *Neural Information Processing Systems (NeurIPs)*.

Chen, V. C. (2008). Doppler signatures of radar backscattering from objects with micro-motions. *IET Signal Processing*, *2*(3), 291–300.

Chierchia, G., Cozzolino, D., Poggi, G., & Verdoliva, L. (2017). Sar image despeckling through convolutional neural networks.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The Cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 3213–3223).

Danzer, A., Griebel, T., Bach, M., & Dietmayer, K. (2019). 2d car detection in radar data with pointnets. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, (pp. 61–66). IEEE.

de la Riva, M., & Mettes, P. S. M. (2019). Bayesian 3d convnets for action recognition from few examples. In *IEEE International Conference on Computer Vision Workshops*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. of the IEEE/CVF Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Desai, S., & Durrett, G. (2020). Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (pp. 295–302). Online: Association for Computational Linguistics.

Ding, J., Chen, B., Liu, H., & Huang, M. (2016a). Convolutional neural network with data augmentation for SAR target recognition. *IEEE Geoscience and Remote Sensing Letters*, *13*(3), 364–368.

Ding, J., Chen, B., Liu, H., & Huang, M. (2016b). Convolutional neural network with data augmentation for sar target recognition. *IEEE Geoscience and Remote Sensing Letters*, *13*, 364–368.

Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., & Darrell, T. (2016). Long-term recurrent convolutional networks for visual recognition and description.

Donahue, J., & Simonyan, K. (2019). Large scale adversarial representation learning. *Neural Information Processing Systems (NeurIPs).*

Dubé, R., Hahn, M., Schutz, M., Dickmann, J., & Gingras, D. (2014). Detection of parked vehicles from a radar based occupancy grid. In *IEEE Intelligent Vehicles Symposium Proceedings*, (pp. 1415–1420).

Engelhardt, N., Pérez, R., & Rao, Q. (2019). Occupancy grids generation using deep radar network for autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, (pp. 2866–2871).

Feng, Z., Zhang, S., Kunert, M., & Wiesbeck, W. (2019). Point cloud segmentation with a high-resolution automotive radar. In *AmE 2019-Automotive meets Electronics; 10th GMM-Symposium*, (pp. 1–5). VDE.

Filos, A., Farquhar, S., Gomez, A. N., Rudner, T. G. J., Kenton, Z., Smith, L., Alizadeh, M., de Kroon, A., & Gal, Y. (2019). A systematic comparison of bayesian deep learning robustness in diabetic retinopathy tasks.

Foong, A. Y. K., Burt, D. R., Li, Y., & Turner, R. E. (2019). On the Expressiveness of Approximate Inference in Bayesian Neural Networks. *arXiv e-prints*, (p. arXiv:1909.00719).

Fortunato, M., Blundell, C., & Vinyals, O. (2017). Bayesian recurrent neural networks. *arXiv preprint arXiv:1704.02798.*

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML).*

Gast, J., & Roth, S. (2018). Lightweight probabilistic deep networks.

Goan, E., & Fookes, C. (2020). Bayesian neural networks: An introduction and survey. *Lecture Notes in Mathematics*, (pp. 45–87).

Godard, C., Aodha, O. M., Firman, M., & Brostow, G. (2019). Digging into self-supervised monocular depth estimation.

Gong, M., Zhao, J., Liu, J., Miao, Q., & Jiao, L. (2016). Change detection in synthetic aperture radar images based on deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *27*, 125–138.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR).*

Grondahl, T., Pajola, L., Juuti, M., Conti, M., & Asokan, N. (2018). All you need is "love": Evading hate speech detection. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, AISec 2018, (pp. 2–12). New York, NY, USA: Association for Computing Machinery.

Grosch, T. (1995). Radar sensors for automotive collision warning and avoidance. *Proc. , SPIE*, *2463*.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, (pp. 1321–1330). Sydney, Australia.

Gupta, K., Rahimi, A., Ajanthan, T., Mensink, T., Sminchisescu, C., & Hartley, R. (2020). Calibration of neural networks using splines.

Gurbuz, S., & Amin, M. (2019). Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring. *IEEE Signal Processing Magazine*, *36*, 16–28.

Gustafsson, F. K., Danelljan, M., & Schoen, T. B. (2020). Evaluating scalable bayesian deep learning methods for robust computer vision.

Harper, R., & Southern, J. (2020). A bayesian deep learning framework for end-to-end prediction of emotion from heartbeat. *IEEE Transactions on Affective Computing*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 770–778).

Hein, M., Andriushchenko, M., & Bitterwolf, J. (2019). Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hendrycks, D., & Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations (ICLR)*.

Hendrycks, D., & Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., & Lakshminarayanan, B. (2020). Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations (ICLR)*.

Heuel, S., & Rohling, H. (2010). Pedestrian recognition based on 24 ghz radar sensors. In *11-th INTERNATIONAL RADAR SYMPOSIUM*, (pp. 1–6).

Heuel, S., & Rohling, H. (2011). Two-stage pedestrian classification in automotive radar systems. In *2011 12th International Radar Symposium (IRS)*, (pp. 477–484).

Heuel, S., & Rohling, H. (2012). Pedestrian classification in automotive radar systems. In *2012 13th International Radar Symposium*, (pp. 39–44).

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–1780.

Holstein, T., Dodig-Crnkovic, G., & Pelliccione, P. (2018). Ethical and social aspects of self-driving cars.

Huang, G., Liu, Z., v. d. Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, P.-Y., Hsu, W.-T., Chiu, C.-Y., Wu, T.-F., & Sun, M. (2018). Efficient uncertainty estimation for semantic segmentation in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, (pp. 520–535).

Huizing, A., Heiligers, M., Dekker, B., de Wit, J., Cifola, L., & Harmanny, R. (2019). Deep learning for classification of mini-uavs using micro-doppler spectrograms in cognitive radar. *IEEE Aerospace and Electronic Systems Magazine*, *34*(11), 46–56.

Hutson, M., et al. (2017). Even artificial intelligence can acquire biases against race and gender. *Science*, *10*.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, (pp. 448–456).

Iqbal, H., Sajjad, M. B., Mueller, M., & Waldschmidt, C. (2015). Sar imaging in an automotive scenario. In *2015 IEEE 15th Mediterranean Microwave Symposium (MMS)*, (pp. 1–4).

Jain, S., Liu, G., Mueller, J., & Gifford, D. (2020). Maximizing overall diversity for improved uncertainty estimates in deep ensembles. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 4264–4271.

Ji, B., Jung, H., Yoon, J., Kim, K., & y. Shin (2019). Bin-wise temperature scaling (BTS): Improvement in confidence calibration performance through simple scaling techniques. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, (pp. 4190–4196).

Jiang, Z., Xu, F. F., Araki, J., & Neubig, G. (2020). How can we know what language models know? *Transactions of the Association for Computational Linguistics*, *8*, 423–438.

Jochumsen, L. W., Nielsen, E., Ostergaard, J., Jensen, S. H., & Pedersen, M. Ø. (2015). Using position uncertainty in recursive automatic target classification of radar tracks. In *2015 IEEE Radar Conference (RadarCon)*, (pp. 0168–0173). IEEE.

Jochumsen, L. W., Pedersen, M. O., Hansen, K., Jensen, S. H., & Ostergaard, J. (2014). Recursive bayesian classification of surveillance radar tracks based on kinematic with temporal dynamics and static features. In *2014 International Radar Conference*, (pp. 1–6).

Jokanovic, B., & Amin, M. (2018). Fall detection using deep learning in range-doppler radars. *IEEE Transactions on Aerospace and Electronic Systems*, *54*(1), 180–189.

Kan, T., xin, G., xiaowei, L., & zhongshan, L. (2020). Implementation of real-time automotive sar imaging. In *2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, (pp. 1–4).

Keat, C. T. M., Pradalier, C., & Laugier, C. (2005). Vehicle detection and car park mapping using laser scanner. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 2054–2060).

Kendall, A., & Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *Neural Information Processing Systems (NeurIPs)*.

Khalid, H.-U.-R., Pollin, S., Rykunov, M., Bourdoux, A., & Sahli, H. (2019). Convolutional long short-term memory networks for Doppler-radar based target classification. In *IEEE Radar Conference*.

Khan, R. H., & Power, D. (1995). Aircraft detection and tracking with high frequency radar. In *Proceedings International Radar Conference*, (pp. 44–48).

Kim, B., Kim, H., Kim, K., Kim, S., & Kim, J. (2019). Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 9012–9020).

Kim, Y., & Moon, T. (2016). Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, *13*(1), 8–12.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*, *abs/1412.6980*.

Kingma, D. P., Salimans, T., & Welling, M. (2015). Variational dropout and the local reparameterization trick. In *Neural Information Processing Systems (NeurIPs)*, (pp. 2575–2583).

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*.

Kraus, F., Scheiner, N., Ritter, W., & Dietmayer, K. (2020). Using machine learning to detect ghost images in automotive radar. *arXiv preprint arXiv:2007.05280*.

Kristiadi, A., Hein, M., & Hennig, P. (2020). Being Bayesian, even just a bit, fixes overconfidence in ReLU networks. *arXiv preprint arXiv:2002.10118*.

Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. .

Kull, M., Filho, T. S., & Flach, P. (2017). Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *aistats*, (pp. 623–631). Fort Lauderdale, FL, USA.

Kull, M., Perello Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., & Flach, P. (2019). Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Neural Information Processing Systems (NeurIPs)*, (pp. 12295–12305).

Kumar, A., Liang, P. S., & Ma, T. (2019). Verified uncertainty calibration. In *Neural Information Processing Systems (NeurIPs)*, (pp. 3787–3798).

Kumar, A., Sarawagi, S., & Jain, U. (2018). Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning (ICML)*, (pp. 2805–2814).

Kuppers, F., Kronenberger, J., Shantia, A., & Haselhoff, A. (2020). Multivariate confidence calibration for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, (pp. 326–327).

Kurakin, A., Goodfellow, I. J., & Bengio, S. (2017). Adversarial machine learning at scale. *International Conference on Learning Representations (ICLR)*.

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

Lang, P., Fu, X., Martorella, M., Dong, J., Qin, R., Meng, X., & Xie, M. (2020). A comprehensive survey of machine learning applied to radar signal processing.

Lang, Y., Hou, C., Yang, Y., Huang, D., & He, Y. (2017). Convolutional neural network for human micro-doppler classification. *European Microwave Conference*.

Larrazabal, A. J., Nieto, N., Peterson, V., Milone, D. H., & Ferrante, E. (2020). Gender imbalance in medical imaging datasets produces biased classifiers for computer-aided diagnosis. *Proceedings of the National Academy of Sciences*, *117*(23), 12592–12594.

Le, H. T., Phung, S. L., Bouzerdoum, A., & Tivive, F. H. C. (2018). Human motion classification with micro-doppler radar and bayesian-optimized convolutional neural networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (pp. 2961–2965). IEEE.

Lecun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time-series. In *The Handbook of Brain Theory and Neural Networks*. MIT Press.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

Lee, K., Lee, H., Lee, K., & Shin, J. (2018). Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations (ICLR).*

Lee, S. (2020). Deep learning on radar centric 3d object detection. *arXiv preprint arXiv:2003.00851*.

Li, T., Fan, L., Zhao, M., Liu, Y., & Katabi, D. (2019a). Making the invisible visible: Action recognition through walls and occlusions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).*

Li, X., He, Y., & Jing, X. (2019b). A survey of deep learning-based human activity recognition in radar. *Remote Sensing*, *11*(9), 1068.

Lin, T., Chen, X., Tang, X., He, L., He, S., & Hu, Q. (2020). *Radar Spectral Maps Classification Based on Deep Learning*, (pp. 29–33). New York, NY, USA: Association for Computing Machinery.

Liu, G., Zhou, M., Wang, L., Wang, H., & Guo, X. (2017). A blind spot detection and warning system based on millimeter wave radar for driver assistance. *Optik*, *135*, 353 – 365.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., & Berg, A. (2016). SSD: Single shot multibox detector. In *European Conference on Computer Vision*, (pp. 21–37).

Liu, X., Li, Y., Wu, C., & Hsieh, C.-J. (2019). Adv-BNN: Improved adversarial defense through robust bayesian neural network. *International Conference on Learning Representations, ICLR*.

Lombacher, J., Hahn, M., Dickmann, J., & Wöhler, C. (2015). Detection of arbitrarily rotated parked cars based on radar sensors. In *2015 16th International Radar Symposium (IRS)*, (pp. 180–185).

Lombacher, J., Hahn, M., Dickmann, J., & Wöhler, C. (2016). Potential of radar for static object classification using deep learning methods. In *IEEE Int. Conference on Microwaves for Intelligent Mobility*, (pp. 1–4).

Lombacher, J., Laudt, K., Hahn, M., Dickmann, J., & Wöhler, C. (2017). Semantic radar grids. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, (pp. 1170–1175).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 3431–3440).

Louizos, C., & Welling, M. (2017). Multiplicative normalizing flows for variational Bayesian neural networks. In *International Conference on Machine Learning (ICML)*, (pp. 2218–2227). Sydney, Australia.

Lydia, A. A., & Francis, F. S. (2020). Multi-label classification using deep convolutional neural network. In *2020 International Conference on Innovative Trends in Information Technology (ICITIIT)*, (pp. 1–6).

Maitre, J., Bouchard, K., & Gaboury, S. (2020). Fall detection with uwb radars and cnn-lstm architecture. *IEEE Journal of Biomedical and Health Informatics*, (pp. 1–1).

Major, B., Fontijne, D., Ansari, A., Teja Sukhavasi, R., Gowaikar, R., Hamilton, M., Lee, S., Grzechnik, S., & Subramanian, S. (2019). Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, (pp. 0–0).

Malinin, A., & Gales, M. (2018). Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, (pp. 7047–7058).

Malinin, A., & Gales, M. (2019). Reverse KL-divergence training of prior networks: Improved uncertainty and adversarial robustness. In *Advances in Neural Information Processing Systems*, (pp. 14547–14558).

Malinin, A., Mlodozeniec, B., & Gales, M. (2020). Ensemble distribution distillation. *International Conference on Learning Representations, ICLR*.

Maronas, J., & Paredes, R. (2020). Improving calibration in mixup-trained deep neural networks through confidence-based loss functions.

Meinke, A., & Hein, M. (2020). Towards neural networks that provably know when they don't know. *International Conference on Learning Representations, ICLR*.

Meronen, L., Irwanto, C., & Solin, A. (2020). Stationary activations for uncertainty calibration in deep learning. *arXiv preprint arXiv:2010.09494*.

Metzen, J. H., Genewein, T., Fischer, V., & Bischoff, B. (2017). On detecting adversarial perturbations. *International Conference on Learning Representations, ICLR*.

Milios, D., Camoriano, R., Michiardi, P., Rosasco, L., & Filippone, M. (2018). Dirichlet-based gaussian processes for large-scale calibrated classification. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.) *Neural Information Processing Systems (NeurIPs)*, (pp. 6005–6015).

Miok, K., Nguyen-Doan, D., Zaharie, D., & Robnik-Šikonja, M. (2019). Generating data using monte carlo dropout. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, (pp. 509–515). IEEE.

Molchanov, P., Vinel, A., Astola, J., & Egiazarian, K. (2013). Radar frequency band invariant pedestrian classification. In *2013 14th International Radar Symposium (IRS)*, vol. 2, (pp. 740–745).

Mostajabi, M., Ming Wang, C., Ranjan, D., & Hsyu, G. (2020). High-resolution radar dataset for semi-supervised learning of dynamic objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, (pp. 100–101).

Mueller, R., Kornblith, S., & Hinton, G. E. (2019). When does label smoothing help? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alche'-Buc, E. Fox, & R. Garnett (Eds.) *Advances in Neural Information Processing Systems*, vol. 32, (pp. 4694–4703). Curran Associates, Inc.

Mukherjee, S., Zimmer, A., Kottayil, N. K., Sun, X., Ghuman, P., & Cheng, I. (2018). Cnn-based insar denoising and coherence metric. *2018 IEEE SENSORS*.

Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P. H. S., & Dokania, P. K. (2020). Calibrating deep neural networks using focal loss. *Neural Information Processing Systems (NeurIPs)*.

Mukhtar, A., Xia, L., & Tang, T. B. (2015). Vehicle detection techniques for collision avoidance systems: A review. *IEEE Trans. Intelligent Transportation Systems*, *16*(5), 2318–2338.

Mullachery, V., Khera, A., & Husain, A. (2018). Bayesian neural networks. *arXiv preprint arXiv:1801.07710*.

Mure-Dubois, J., Vincent, F., & Bonacci, D. (2011). Sonar and radar sar processing for parking lot detection. In *2011 12th International Radar Symposium (IRS)*, (pp. 471–476).

Naeini, M. P., Cooper, G. F., & Hauskrecht, M. (2015). Obtaining well calibrated probabilities using Bayesian binning. In *Proc. of Conference on Artificial Intelligence (AAAI)*, (pp. 2901–2907).

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Nguyen, A. M., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., & Tran, D. (2019). Measuring calibration in deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Oezcan, M. B., Guerbuez, S. Z., Persico, A. R., Clemente, C., & Soraghan, J. (2016). Performance analysis of co-located and distributed mimo radar for micro-doppler classification. In *2016 European Radar Conference (EuRAD)*, (pp. 85–88).

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., & Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*.

Palffy, A., Dong, J., Kooij, J. F., & Gavrila, D. M. (2020). Cnn based road user detection using the 3d radar cube. *IEEE Robotics and Automation Letters*, *5*(2), 1263–1270.

Pang, G., Shen, C., Cao, L., & Hengel, A. v. d. (2020). Deep learning for anomaly detection: A review. *arXiv preprint arXiv:2007.02500*.

Pang, Y., & Liu, Y. (2020). Probabilistic aircraft trajectory prediction considering weather uncertainties using dropout as bayesian approximate variational inference. In *AIAA Scitech 2020 Forum*, (p. 1413).

Park, T., Liu, M.-Y., Wang, T.-C., & Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Patel, K., Beluch, W., Rambach, K., Cozma, A.-E., Pfeiffer, M., & Yang, B. (2021a). Investigation of uncertainty of deep learning-based object classification on radar spectra. In *IEEE Radar Conference (RadarConf)*.

Patel, K., Beluch, W., Yang, B., Pfeiffer, M., & Zhang, D. (2021b). Multi-class uncertainty calibration via mutual information maximization-based binning. In *International Conference on Learning Representations (ICLR)*.

Patel, K., Beluch, W., Zhang, D., Pfeiffer, M., & Yang, B. (2020). On-manifold adversarial data augmentation improves uncertainty calibration. In *2020 26th International Conference on Pattern Recognition (ICPR)*. IEEE Computer Society.

Patel, K., Rambach, K., Visentin, T., Rusev, D., Pfeiffer, M., & Yang, B. (2019). Deep learning-based object classification on automotive radar spectra. In *IEEE Radar Conference (RadarConf)*.

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., & Hinton, G. (2017). Regularizing neural networks by penalizing confident output distributions. *International Conference on Learning Representations (ICLR)*.

Pérez, R., Schubert, F., Rasshofer, R., & Biebl, E. (2018). Single-frame vulnerable road users classification with a 77 ghz fmcw radar sensor and a convolutional neural network. In *2018 19th International Radar Symposium (IRS)*, (pp. 1–10). IEEE.

Peterson, J., Battleday, R., Griffiths, T. L., & Russakovsky, O. (2019). Human uncertainty makes classification more robust. In *IEEE International Conference on Computer Vision (ICCV).*

Platt, J. (1999). Probabilities for SV machines. In *Advances in Large Margin Classifiers*, (pp. 61–74).

Postels, J., Ferroni, F., Coskun, H., Navab, N., & Tombari, F. (2019). Sampling-free epistemic uncertainty estimation using approximated variance propagation. In *IEEE International Conference on Computer Vision (ICCV).*

Prophet, R., Hoffmann, M., Vossiek, M., Sturm, C., Ossowska, A., Malik, W., & Lübbert, U. (2018). Pedestrian classification with a 79 GHz automotive radar sensor. In *International Radar Symposium (IRS)*, (pp. 1–6).

Prophet, R., Martinez, J., Michel, J. F., Ebelt, R., Weber, I., & Vossiek, M. (2019). Instantaneous ghost detection identification in automotive scenarios. In *2019 IEEE Radar Conference (RadarConf)*, (pp. 1–6).

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space.

Razavi, A., & van den Oord an Oriol Vinyals, A. (2019). Generating diverse high-fidelity images with VQ-VAE-2. *International Conference on Learning Representations (ICLR).*

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, (pp. 91–99).

Reynolds, D. (2009). *Gaussian Mixture Models*, (pp. 659–663). Boston, MA: Springer US.

Ritter, H., Botev, A., & Barber, D. (2018). A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, vol. 6. International Conference on Representation Learning.

Roelofs, R., Cain, N., Shlens, J., & Mozer, M. C. (2020). Mitigating bias in calibration error estimation.

Rohling, H. (2011). Ordered Statistic CFAR technique - an overview. In *International Radar Symposium*, (pp. 631–638).

Rohling, H., Heuel, S., & Ritter, H. (2010). Pedestrian detection procedure integrated into an 24 ghz automotive radar. In *IEEE Radar Conference*, (pp. 1229–1232).

Roodaki, P. M., Taghian, F., Bashirzadeh, S., & Jalaali, M. (2011). A survey of millimeter-wave technologies. In *2011 International Conference on Electrical and Control Engineering*, (pp. 5726–5728).

Rosca, M., Lakshminarayanan, B., Warde-Farley, D., & Mohamed, S. (2017). Variational approaches for auto-encoding generative adversarial networks. *arxiv preprint arxiv:1706.0498*.

Roy, A. G., Conjeti, S., Navab, N., Wachinger, C., Initiative, A. D. N., et al. (2019). Bayesian quicknat: model uncertainty in deep whole-brain segmentation for structure-wise quality control. *NeuroImage*, *195*, 11–22.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *IEEE International Conference on Computer Vision (ICCV)*.

Ryou, S., Jeong, S.-G., & Perona, P. (2019). Anchor loss: Modulating loss scale based on prediction difficulty. In *IEEE International Conference on Computer Vision (ICCV)*.

Saltikoff, E., Friedrich, K., Soderholm, J., Lengfeld, K., Nelson, B., Becker, A., Hollmann, R., Urban, B., Heistermann, M., & Tassone, C. (01 Sep. 2019). An overview of using weather radar for climatological studies: Successes, challenges, and potential. *Bulletin of the American Meteorological Society*, *100*(9), 1739 – 1752.

Scheiner, N., Appenrodt, N., Dickmann, J., & Sick, B. (2019). Radar-based road user classification and novelty detection with recurrent neural network ensembles. *2019 IEEE Intelligent Vehicles Symposium (IV)*.

Schubert, E., Kunert, M., Menzel, W., Fortuny-Guasch, J., & Chareau, J. (2013). Human rcs measurements and dummy requirements for the assessment of radar based active pedestrian safety systems. In *2013 14th International Radar Symposium (IRS)*, vol. 2, (pp. 752–757).

Schubert, E., Meinl, F., Kunert, M., & Menzel, W. (2015). Clustering of high resolution automotive radar detections and subsequent feature extraction for classification of road users. In *International Radar Symposium*, (pp. 174–179).

Schubert, E., Meinl, F., Kunert, M., & Menzel, W. (2015). Clustering of high resolution automotive radar detections and subsequent feature extraction for classification of road users. In *International Radar Symposium (IRS)*, (pp. 174–179).

Schumann, O., Hahn, M., Dickmann, J., & Wöhler, C. (2018). Semantic segmentation on radar point clouds. In *International Conference on Information Fusion*, (pp. 2179–2186).

Schuster, M., Blaich, M., & Reuter, J. (2014). Collision avoidance for vessels using a low-cost radar sensor. *IFAC Proceedings Volumes*, *47*(3), 9673 – 9678. 19th IFAC World Congress.

Sensoy, M., Kaplan, L., & Kandemir, M. (2018). Evidential deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing Systems*, (pp. 3179–3189).

Shafaei, A., Schmidt, M., & Little, J. J. (2019). A less biased evaluation of out-of-distribution sample detectors. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, (p. 3). BMVA Press.

Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). Fast yolo: A fast you only look once system for real-time embedded object detection in video. *arXiv preprint arXiv:1709.05943*.

Sheeny, M., Wallace, A., & Wang, S. (2019). 300 ghz radar object recognition based on deep neural networks and transfer learning.

Sheeny, M., Wallace, A., & Wang, S. (2020). Radio: Parameterized generative radar data augmentation for small datasets. *Applied Sciences*, *10*(11), 3861.

Shi, X., & Wang, C. (2010). An automatic parking system based on laser radar. *Mechatronics*, *16*(3), 72–74.

Shizhe Zang, M. A. K., Ming Ding (2019). The impact of adverse weather conditions on autonomous vehicles: Examining how rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE Vehicular Technology Magazine*, (pp. 828–841).

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*, 484–503.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*.

Singh, S. (2015). Critical reasons for crashes investigated in the national motor vehicle crash causation survey. *National Motor Vehicle Crash Causation*.

Skolnik, M. (2008). *Radar Handbook*. New York: McGraw-Hill, 3rd ed.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Stutz, D., Hein, M., & Schiele, B. (2019). Disentangling adversarial robustness and generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Su, J., Vargas, D. V., & Sakurai, K. (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, *23*(5), 828–841.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Conference on Artificial Intelligence*, (pp. 4278–4284). San Francisco, U.S.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tahmoush, D. (2015). Review of micro-doppler signatures. *IET Radar, Sonar & Navigation*, *9*(9), 1140–1146.

Thulasidasan, S., Chennupati, G., Bilmes, J., Bhattacharya, T., & Michalak, S. (2019). On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Neural Information Processing Systems (NeurIPs)*.

Tishby, N., Pereira, F. C., & Bialek, W. (1999). The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, (pp. 368–377).

Tommasi, T., Patricia, N., Caputo, B., & Tuytelaars, T. (2017). A deeper look at dataset bias. In *Domain adaptation in computer vision applications*, (pp. 37–55). Springer.

Torralba, A., & Efros, A. A. (2011). Unbiased look at dataset bias. In *CVPR 2011*, (pp. 1521–1528). IEEE.

Ulrich, M., Glaeser, C., & Timm, F. (2021). Deepreflecs: Deep learning for automotive object classification with radar reflections. In *IEEE Radar Conference*.

Vaicenavicius, J., Widmann, D., Andersson, C., Lindsten, F., Roll, J., & Schön, T. (2019). Evaluating model calibration in classification. In *aistats*, (pp. 3459–3467). Okinawa, Japan.

Vapnik, V. (1992). Principles of risk minimization for learning theory. In *Neural Information Processing Systems (NeurIPs).*

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., & Bengio, Y. (2019). Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning (ICML).*

Visentin, T. (2019). *Polarimetric Radar for Automotive Applications*. Ph.D. thesis, Karlsruher Institut für Technologie (KIT).

Visentin, T., Hasch, J., & Zwick, T. (2017). Calibration of a fully polarimetric 8x8 MIMO FMCW radar system at 77 GHz. In *11th European Conference on Antennas and Propagation*, (pp. 2530–2534).

Visentin, T., Hasch, J., & Zwick, T. (2018). Analysis of multipath and DOA detection using a fully polarimetric automotive radar. *International Journal of Microwave and Wireless Technologies*, *10*(5-6), 570–577.

Wang, J., Zheng, T., Lei, P., & Bai, X. (2018). Ground target classification in noisy sar images using convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *11*(11), 4180–4192.

Wang, P., Zhang, H., & Patel, V. M. (2017). SAR image despeckling using a convolutional neural network. *IEEE Signal Processing Letters*, *24*(12), 1763–1767.

Wang, W., Song, Y., Zhang, J., & Deng, H. (2014). Automatic parking of vehicles: A review of literatures. *International Journal of Automotive Technology*, *15*, 967–978.

Wang, Y. (2017). Classification and tracking of moving objects for automotive radar. *IEEE International Conference on Computational Electromagnetics (ICCEM).*

Wang, Y., Jiang, Z., Li, Y., Hwang, J.-N., Xing, G., & Liu, H. (2021). Rodnet: A real-time radar object detection network cross-supervised by camera-radar fused object 3d localization. *IEEE Journal of Selected Topics in Signal Processing*, *15*(4), 954–967.

Weng, S., Li, J., Chen, Y., & Wang, C. (2016). Road traffic sign detection and classification from mobile lidar point clouds. In *SPIE*, vol. 9901, (p. 99010A).

Wenger, J., Kjellström, H., & Triebel, R. (2020). Non-parametric calibration for classification. In *aistats*.

Widmann, D., Lindsten, F., & Zachariah, D. (2019). Calibration tests in multi-class classification: A unifying framework. In *NeurIPs*, (pp. 12257–12267).

Wu, A., Nowozin, S., Meeds, E., Turner, R. E., Hernández-Lobato, J. M., & Gaunt, A. L. (2019). Deterministic variational inference for robust bayesian neural networks. *International Conference on Learning Representations, ICLR*.

Wu, H., & Zwick, T. (2009). Automotive sar for parking lot detection. In *2009 German microwave conference*, (pp. 1–8). IEEE.

Xie, S., Girshick, R. B., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yeo, H.-S., Flamich, G., Schrempf, P., Harris-Birtill, D., & Quigley, A. (2016). Radarcat: Radar categorization for input & interaction. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, (pp. 833–841). New York, NY, USA: Association for Computing Machinery.

Yonatan, G., & Ran, E.-Y. (2017). Selective classification for deep neural networks. In *Neural Information Processing Systems (NeurIPs)*.

Yoneda, K., Suganuma, N., Yanase, R., & Aldibaja, M. (2019a). Automated driving recognition technologies for adverse weather conditions. *IATSS Research*, *43*(4), 253 – 262.

Yoneda, K., Suganuma, N., Yanase, R., & Aldibaja, M. (2019b). Automated driving recognition technologies for adverse weather conditions. *IATSS Research*, *43*(4), 253 – 262.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *IEEE International Conference on Computer Vision (ICCV)*.

Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *International Conference on Machine Learning (ICML)*, (pp. 609–616).

Zadrozny, B., & Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *SIGKDD Conference on Knowledge Discovery and Data Mining*, (pp. 694–699).

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*.

Zaied, S., Toumi, A., & Khenchaf, A. (2018). Target classification using convolutional deep learning and auto-encoder models. In *2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, (pp. 1–6).

Zhang, H., Cissé, M., Dauphin, Y., & Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. *International Conference on Learning Representations (ICLR).*

Zhang, J., Kailkhura, B., & Han, T. (2020). Mix-n-Match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International Conference on Machine Learning (ICML)*. Vienna, Austria.

Zhang, J., Tao, J., & Shi, Z. (2017). Doppler-radar based hand gesture recognition system using convolutional neural networks. In *International Conference in Communications, Signal Processing, and Systems*, (pp. 1096–1113). Springer.

Zhang, J., & Zong, C. (2015). Deep neural networks in machine translation: An overview. *IEEE Intelligent Systems*, *30*, 16–25.

Zhang, R., & Cao, S. (2019). Real-time human motion behavior detection via cnn using mmwave radar. *IEEE Sensors Letters*, *3*(2), 1–4.

Zhao, J., Guo, W., Liu, B., Cui, S., Zhang, Z., & Wenxian, Y. (2017a). Convolutional neural network-based sar image classification with noisy labels. *Journal of Radars*, *6*, 514–523.

Zhao, M., Li, T., Alsheikh, M. A., Tian, Y., Zhao, H., Torralba, A., & Katabi, D. (2018a). Through-wall human pose estimation using radio signals. In *IEEE Conference on Computer Vision and Pattern Recognition.*

Zhao, M., Tian, Y., Zhao, H., Alsheikh, M., Li, T., Hristov, R., Kabelac, Z., Katabi, D., & Torralba, A. (2018b). RF-based 3D skeletons. In *ACM Special Interest Group on Data Communication*, (pp. 267–281).

Zhao, M., Yue, S., Katabi, D., Jaakkola, T. S., & Bianchi, M. T. (2017b). Learning sleep stages from radio signals: A conditional adversarial architecture. In *International Conference on Machine Learning*, (pp. 4100–4109).

Zhao, M., et al. (2017c). *Emotion recognition using wireless signals*. Ph.D. thesis, Massachusetts Institute of Technology.

Zhao, Z., Song, Y., Cui, F., Zhu, J., Song, C., Xu, Z., & Ding, K. (2020). Point cloud features-based kernel svm for human-vehicle classification in millimeter wave radar. *IEEE Access*, *PP*, 1–1.

Zou, J., & Schiebinger, L. (2018). AI can be sexist and racist - it's time to make it fair. *Nature Publishing Group.*