CentraleSupélec

IAEM

Epistemic Calibration for Bayesian Deep Learning: Principles, Issues and Solutions

THÈSE

*Présentée et soutenue publiquement le **7 juillet 2025** pour l'obtention du*

Doctorat de CentraleSupélec

NNT : 2025CSUP0002

Mention : Automatique, Traitement du Signal et des Images, Génie Informatique

Mohammed FELLAJI

Jury:

Rapporteurs :		
	Sébastien DESTERCKE	Directeur de recherche, CNRS – *(Président)*
	Willem WAEGEMAN	Professeur, Université de Gand
Examinatrices :		
	Marianne CLAUSEL	Professeure, Université de Lorraine
	Ines LYNCE	Professeure, IST, Université de Lisbonne
Directeur de thèse :		
	Miguel COUCEIRO	Professeur, IST, Université de Lisbonne
Co-directeur de thèse :		
	Frédéric PENNERATH	Maitre de conférence, CentraleSupélec
Membre Invité :		
	Brieuc CONAN-GUEZ	Maitre de conférence, Université de Lorraine

Loria
Laboratoire lorrain de recherche en informatique et ses applications

CNRS, INRIA, Université de Lorraine, CentraleSupélec

Abstract

Although most deep learning models provide probabilistic distributions as a predictive output, their evaluation often relies mainly on raw performance metrics (*e.g.* accuracy for classification) insensitive to the uncertainty expressed by these distributions. Yet, the inherent restrictions on the generalization ability of these models make them extremely unlikely to reach flawless performance on new data, hence advocating for the importance of examining the confidence of the predictions. In this regard, the field of *model calibration* has recently gained considerable attention in the deep learning community, with the aim of encouraging reliable predictions. Meanwhile, the development of models like *Bayesian neural networks*, *deep ensemble* or *evidential deep models* has made it possible to estimate the level of *epistemic uncertainty*, inherent to the learning process, in complement to the *aleatoric uncertainty* already estimated by standard models.

While the quality of predictive/aleatoric uncertainty can be measured by well-established calibration methods, the same cannot be said about epistemic uncertainty. Since the latter is considered the ideal score in a range of applications, it is therefore of utmost importance to explore its calibration properties, which has rarely been addressed in the literature. When attempting to define epistemic calibration, more challenges arise on how to formalize this calibration, assuming its existence. For instance, it may be worth considering whether it is feasible to study it similarly to model calibration, or at the very least, based on fundamental principles. Throughout this thesis, we have attempted to overcome these challenges by conducting work of both a theoretical and experimental nature in the specific context of deep classifiers.

After reviewing the state of the art to quantifying probabilistic uncertainty, especially in the field of deep models, and given the difficulty of quantitatively calibrating epistemic uncertainty, we first define formally two elementary principles that epistemic uncertainty should ideally satisfy to our view: *data-related* and *model-related* principles. Indeed, as epistemic uncertainty is associated with knowledge in the model, it should decrease with the amount of available data and increase with the expressivity/complexity of the model. Empirically, and on a variety of datasets, we show that commonly used Bayesian models or alternatives do not fully verify these fundamental principles. Therefore, we argue that these models lack epistemic calibration, and we refer to this phenomenon as the *epistemic uncertainty hole*.

Considering the critical role that the prior plays in shaping epistemic uncertainty, we investigate how much this failure of the tested models is due to an inadequate choice of prior. To this end, we introduce *Conflictual loss*, a loss function that favors diversity of the outputs thanks to the use of an *uninformative prior*. We then experimentally show that Conflictual loss leads to a better calibrated epistemic uncertainty and does not suffer from the epistemic uncertainty hole. Additionally, special inputs were investigated, which were either noisy samples or drawn from the test set, to understand the evolution of different sources of uncertainties. Furthermore, we analyze the specificities of the conflictual diversity in the parameters space, and highlight the differences with deep ensembles. Building on the findings of this analysis, a compact version of the model was formalized, further emphasizing the benefits of the uninformative prior. Finally, the models were evaluated on popular applications such as out-of-distribution (OOD) detection and Bayesian active learning.

Résumé

La plupart des modèles d'apprentissage profond produisent des distributions probabilistes pour leurs prédictions, mais leur évaluation se base souvent sur des métriques de performance brute (commme la précision de la classification), ignorant l'incertitude des prédictions. L'importance de mesurer la confiance des modèles est soulignée en raison de leurs limites de généralisation et de la nécessité de prévoir des performances fiables sur de nouvelles données. Le domaine de la calibration des modèles en apprentissage profond a ainsi émergé pour promouvoir des prédictions plus fiables. Entre-temps, le développement de modèles tels que les *réseaux neuronaux bayésiens*, les *ensembles profonds* ou les *modèles profonds "evidentiels"* a permis d'estimer le niveau d'*incertitude épistémique*, inhérent au processus d'apprentissage, en complément de l'*incertitude aléatoire* déjà estimée par les modèles standard.

Si la qualité de l'incertitude prédictive/aléatoire peut être mesurée par des méthodes de calibration bien établies, il n'en va pas de même pour l'incertitude épistémique. Cette dernière étant considérée comme la mesure idéale dans une série d'applications, il est donc important d'explorer ses propriétés de calibration, ce qui a rarement été abordé dans la littérature. Définir la calibration épistémique pose des défis quant à sa formalisation, sa comparaison avec la calibration de modèle, et son étude sur la base de principes fondamentaux. Cette thèse aborde ces questions par des approches théoriques et expérimentales appliquées aux classificateurs profonds dans le but d'explorer l'incertitude épistémique et ses implications dans diverses applications.

Après avoir étudié l'état de l'art sur la quantification de l'incertitude probabiliste, notamment dans les modèles profonds, et face à la difficulté de calibrer quantitativement l'incertitude épistémique, nous définissons formellement deux principes élémentaires que cette incertitude devrait idéalement respecter : les principes liés aux *données* et au *modèle*. En effet, l'incertitude épistémique étant liée au modèle, elle devrait diminuer avec la quantité de données disponibles et augmenter avec la complexité ou l'expressivité du modèle. Empiriquement, et sur divers jeux de données, nous montrons que les modèles bayésiens courants ou leurs alternatives ne respectent pas pleinement ces principes fondamentaux. Nous en concluons que ces modèles manquent de calibration épistémique, un phénomène que nous appelons le *trou d'incertitude épistémique*.

Étant donné l'importance du prior dans l'incertitude épistémique, nous analysons si l'échec des modèles provient d'un choix inadéquat du prior. Pour cela, nous introduisons la *Conflictual loss*, une fonction de loss qui favorise la diversité des sorties grâce à l'utilisation d'un *prior non informatif*. Nous montrons expérimentalement que cette loss améliore la calibration de l'incertitude épistémique et ne présente pas le *trou d'incertitude épistémique*. De plus, des entrées particulières ont été étudiées, comme des échantillons bruités ou issus du jeu de test, afin d'analyser l'évolution des différentes sources d'incertitudes. Nous analysons aussi les spécificités de la diversité conflictuelle dans l'espace des paramètres, en soulignant les différences avec les ensembles profonds. À partir de cette analyse, une version compacte du modèle est formalisée, mettant en valeur les apports du prior non informatif. Enfin, les modèles sont évalués sur des applications populaires telles que la détection hors distribution (OOD) et l'apprentissage actif bayésien.

Acknowledgements

This doctoral thesis, a journey of significant intellectual growth and personal development, would not have been possible without the invaluable support and guidance of many. I extend my deepest gratitude to all who have contributed to this endeavor, helping me navigate the complexities of research and the demands of doctoral study.

Firstly, I would like to express my sincere appreciation to the distinguished members of my jury for dedicating their invaluable time to meticulously review my work and for accepting to be a part of this important milestone. Your profound expertise and insightful perspectives were crucial for the rigorous evaluation of this dissertation. Your willingness to engage with my research has provided a final, critical layer of academic scrutiny that is indispensable to the doctoral process.

My profound gratitude goes to my exceptional supervisors, *Miguel*, *Frédéric*, and *Brieuc*. Your unwavering support, from the very first day you accepted me into the PhD program, through the challenging phases of experimentation and analysis, and right up to the final, meticulous stages of manuscript review, has been truly exceptional. Our numerous discussions, both formal and informal, spanning conceptual debates to granular problem-solving, have been instrumental in shaping my research direction, refining my ideas, and overcoming countless obstacles. Your continuous feedback and meticulous guidance throughout this demanding period have been invaluable; each suggestion, every pointed question, and all the constructive criticism contributed significantly to the clarity and robustness of my arguments. This document stands as a testament to your dedication, intellectual generosity, and the countless hours invested in refining my work and nurturing my growth.

I am also deeply thankful to my colleagues, whose presence made the research environment vibrant and supportive. To *Salim*, thank you for the consistently stimulating and insightful discussions we shared over the past years. I also extend my sincere thanks to *Burak*, *Marion*, *Elena* and to all the colleagues I had the chance to work alongside during my PhD.

Finally, and most importantly, I wish to express my heartfelt gratitude to my family, whose love and support formed the bedrock of this entire endeavor. To my parents, thank you for your endless encouragement, and for instilling in me the fundamental values of education, perseverance, and curiosity. To my dear wife, your extraordinary patience, profound understanding, unwavering belief, and unconditional love have been my greatest source of strength, especially during the most challenging and demanding times. Your sacrifices and steadfast support enabled my full dedication to this pursuit. And to my sisters and their families, your constant support, enthusiastic cheers, and the comforting sense of home and belonging you provide have been an indispensable anchor throughout this journey. Thank you all for wholeheartedly supporting my career choice, for celebrating every small victory, and for your enduring love, which made every effort worthwhile and every challenge surmountable.

CONTENTS

CHAPTER 1
INTRODUCTION

"I think it's much more interesting to live not knowing than to have answers which might be wrong."
Richard Feynman

1.1 Motivations

In a world where artificial intelligence (AI) becomes more important and present with each passing day, having reliable models is of utmost importance. This adoption of AI can range from simple tasks to critical applications, such as chatbots, voice assistants, healthcare, fraud detection, transportation, and military, to name a few. Regardless of the importance of the task, the chosen model should ideally yield correct predictions, and more critically be self-aware when unable to provide a reliable prediction. Therefore, it is preferable to consider both the predictions and their (un)certainties.

Let's take the simple example of large language models (LLMs) and ask the same question to a variety of models: *in one line and for each word, give the unique characters with their counts in the form of a dictionary: centralesupelec, strawberries, irregularities.* This example was inspired by the informally known problem *the strawberry problem*[1]. While this simple question does not require advanced level of intelligence, all the tested LLMs appear to fail[2] in counting the number of occurrences of unique characters, as illustrated in Table 1.1. We highlight that the goal of this experiment is to show the pitfalls of "blindly" relying on these models and this is not by any mean a comparison of these models.

Although these problems could be explained (with the tokenization process for example) and newer versions could potentially fix them, our goal here is to highlight the issue of overconfidence in machine learning, and how it could directly impact the user in real world applications. Namely, when the models are asked about their confidence in the results, all the chatbots indicate total confidence in the erroneous counts. Due to the large adoption currently of chatbots, they raise serious concerns regarding their outputs and overconfidence, suggesting a rather strict evaluation of confidence on outputs, or equivalently and more broadly a reliable and calibrated uncertainty measure.

[1]The problem is further shown in https://community.openai.com/t/incorrect-count-of-r-characters-in-the-word-strawberry/829618

[2]We only run the basic free publicly available version of each model on April 3, 2025.

	ChatGPT	DeepSeek	Le Chat	Gemini	Copilot	Qwen
centralesupelec	3c, 2r, 2a, 1l	2a	3c, 0a	1l, 3e	1l	1l
strawberries	2b, 3e	✓	2r	1e, 2r	2r	3s, 2r
irregularities	4i, 3e, 2l	3e, 2g, 2l	4i, 2r, 2l	2i, 1e, 1y	4r	4i, 2r, 2l

TABLE 1.1: Failures of chatbots when asked to count the number of occurrences of all unique characters in a word. We only report the incorrect characters: for example 3c refers to when the model wrongfully outputs 3 occurrences for the character "c".

Take, for example, a more crucial application, where an automated system makes real-world decisions, like self-driving cars, where the consequences could be deadly. The 2018 fatal Uber incident[3] is a good illustration of how overconfidence could affect self-driving cars. During an autonomous operation, a self-driving Uber car hit a pedestrian in Tempe, Arizona, that ended up losing her life. Although the car sensors detected the pedestrian, the model considered her a false positive thus it did not act to prevent the crash. This tragedy highlights how overconfidence can lead to catastrophic failures and show the importance of uncertainty in autonomous decision-making.

1.2 The importance of uncertainties

The previous examples are only a drop in the ocean when it comes to the importance of uncertainties in machine learning. The field of uncertainty quantification has gained significant attention in recent years, partially due to the increased use of machine learning models in different applications and/or in the decision-making loop. Throughout this thesis, we will primarily consider classification models trained through supervised learning and explore their uncertainties. Moreover, as these models are often described as black boxes due to the difficulty in understanding how predictions are made, quantifying uncertainties can provide a better understanding of their behavior.

Therefore, it is logical to ask the following questions: What is uncertainty in the specific context of machine learning? How can we measure the uncertainty associated with a prediction? Are there distinct sources of uncertainties? Is this distinction, if it exists, possible with all types of machine learning models? How to assess the reliability and the quality of a given uncertainty measure?

Usually when uncertainties in machine learning are discussed, it is common to distinguish two main sources of uncertainties: *aleatoric* and *epistemic*. In a nutshell, the former refers to the irreducible part of the uncertainty while the latter to the reducible part as the size of the training set increases. Although philosophical, this distinction is crucial as it has theoretical and practical implications on when to use which. For instance, aleatoric uncertainty is mainly related to the quality of the data and the separability of the classes. Hence, the unobserved features of the input that determines the output (or equivalently the latent variables) is a component of aleatoric uncertainty. On the other hand, epistemic uncertainty is associated with the quantity of the data and the imprecise knowledge about the model's parameters. Indeed, this source is often referred to as the model uncertainty.

Consider a system modeled by a Bernoulli process, representing the outcome of a coin flip. The observed result of each trial (heads or tails) is influenced by a multitude of interacting microphysical (latent) variables, such as aerodynamic forces, angular momentum, and surface interactions upon landing. Despite the deterministic nature of the underlying physical laws, the macroscopic outcome of each individual flip exhibits inherent stochasticity. This intrinsic variability in the outcome of each trial constitutes aleatoric uncertainty, which is considered irreducible due to its fundamental presence within the process itself.

[3] https://wired.com/story/ubers-fatal-self-driving-car-crash-saga-over-operator-avoids-prison

Now, let us assume that the true probability of the coin landing on heads, denoted by the parameter p_H, is unknown. This lack of precise knowledge regarding the value of p_H represents epistemic uncertainty. Epistemic uncertainty arises from a lack of information about the model parameters used to describe the system. Importantly, epistemic uncertainty is reducible. Through the acquisition of empirical data via repeated trials and subsequent statistical inference, it is possible to estimate the parameter p_H with increasing precision, thereby reducing the uncertainty associated with our knowledge of the coin's probabilistic behavior.

While measuring these uncertainties is crucial, it is insufficient to tackle in isolation the overconfidence or equivalently the underconfidence of a given model. To this end, they should be calibrated in order to offer reliable uncertainty information. Although restrictive, most classifiers are effectively trained with the goal of correctly classifying the inputs (*i.e.* through the use of argmax). By evaluating the model on the class level (*i.e.* as measured by accuracy), we are unable to take into account the confidence of the model. Fortunately, the evaluation on this class-wise level can be based on a solid ground-truth as it is easy to obtain labels associated to inputs.

Furthermore, it is common to consider the softmax-probabilities of machine learning classifiers to give a confidence score in addition to the predicted class. Thus, the model should ideally predict the correct class with the right level of confidence. Unfortunately, machine learning models are shown to be overconfident making the calibration of the softmax-probabilities a must in order to reliably represent its confidence. Achieving a calibrated model can be obtained using two main approaches: either by using post-hoc techniques, or by training the model with specific loss functions that regularize the model confidence. This specific calibration is related to aleatoric uncertainty, with the aleatoric ground-truth being also objectively available for the labels.

The calibration of epistemic uncertainty is more challenging, and it will be explored throughout this thesis. For instance, epistemic uncertainty can be measured for only a subset of models, such as credal approaches, Bayesian models, Deep Ensembles and Evidential Deep Learning. In addition, and perhaps more challenging, epistemic uncertainty depends heavily on the choice of the prior distribution (*i.e.* our assumed knowledge about the task, the data and the modeling). With this choice being subjective, epistemic calibration gets even more complicated.

Compared to aleatoric calibration, epistemic calibration is overlooked in the literature, and when explored, it is often attributed to how epistemic uncertainty was measured rather than its calibration. One way to motivate the discussion of epistemic calibration is to encourage a similar methodology as with class-wise and aleatoric levels: How to study epistemic calibration? Are the commonly used models epistemically calibrated? How to restore epistemic calibration of a trained model? Are there loss functions that regularize epistemic uncertainty as it is the case with aleatoric uncertainty?

1.3 Manuscript layout and key findings

In Chapter 2, we start by exploring the models under the Bayesian formalism allowing measuring epistemic uncertainty, namely, Bayesian models and Deep Ensembles (DE). Once uncertainties are defined and further detailed, we discuss how to measure them and focus on the framework of information theory, for its mathematical soundness. In addition, Evidential Deep Learning are also discussed with their particularities and how uncertainties are measured for this family of models.

Given the dependency of epistemic calibration on the prior distribution, we elaborate in Chapter 3 on the large choice of this distribution, often referred to in the literature as the *no-free lunch theorem* for the prior distribution. We then expend the discussion on the notion of aleatoric calibration, formally known as probabilistic model calibration. Finally, we start highlighting the challenges of epistemic uncertainty compared to aleatoric uncertainty.

Now with all the components clearly and formally defined, we tackle in Chapter 4 the methodology of evaluating epistemic calibration. We introduce and formalize two fundamental principles, *data-related* and *model-related* principles and elaborate on their importance. Once our experimental setups are detailed, we show that commonly used Bayesian models lack epistemic calibration. In order to understand this pitfall, we looked at these experiments from different angles and find that these models fail to faithfully represent epistemic uncertainty.

To restore the epistemic calibration of DE, we propose in Chapter 5 *Conflictual loss*, based on the exploratory analysis in Chapter 4. Intuitively, this loss aims at making each model in the ensemble specialized in a single class. When tested under the same experimental protocol of previously tested models, we show that *Conflictual DE*, a DE trained with Conflictual loss, is epistemically calibrated under the two fundamental principles. As we further analyze the trained Conflictual DE in the weight space, we notice that it encourages diversity especially in the last layer. We then present *Conflictual Model* as a compact version of Conflictual DE that is also epistemically calibrated.

Finally, we discuss in Chapter 6 two important applications: out-of-distribution (OOD) detection and active learning. Arguably, epistemic uncertainty is the ideal measure to detect OOD samples, and to explore the input space and to find the ideal samples within the unlabeled pool for labeling. A unique and innovative practical approach will be employed to put this theoretical assumption to the test.

Contributions. This thesis is based on and extends the ideas and concepts explored in our prior research, as published in the following papers:

- Fellaji and Pennerath (2023): Fellaji, M. and Pennerath, F. (2023). The Epistemic Uncertainty Hole: An issue of Bayesian Neural Networks. In *Conférence Sur l'Apprentissage Automatique (Affiliated to PFIA)*, Strasbourg, France
- Fellaji et al. (2024): Fellaji, M., Pennerath, F., Conan-Guez, B., and Couceiro, M. (2024). On the Calibration of Epistemic Uncertainty: Principles, Paradoxes and Conflictual Loss. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 160–176, Vilnius, Lithuania. Springer

CHAPTER 2
UNCERTAINTIES

"Information is the resolution of uncertainty meaning."
Claude Shannon

In the recent years, the uncertainty of machine learning models has gained increasing interest among researchers and machine learning practitioners for several reasons, one of which is the overconfidence of such models. For instance, measuring uncertainties is of great importance for critical applications or for those depending on a reliable estimation of confidence. More precisely, different sources of uncertainties exist with each associated to a particular task. In this chapter, we start by highlighting the different types of models that will be explored throughout this thesis, especially models that are considered Bayesian neural networks. Moreover, we will discuss different ways of computing uncertainties, mainly on classification tasks. Finally, we will shed the light on how to model a second-order probability distribution for standard models.

2.1 Bayesian Neural Networks

In standard deep learning, the optimization problem consists of learning a point estimate of the model parameters. In contrast, by training Bayesian neural networks (BNNs), we are no longer interested in a point estimate but rather in an adequate probability distribution over the model parameters, allowing for uncertainty estimation and more robust predictions. As a result, Bayesian neural networks could be seen as a probabilistic interpretation of standard neural networks. Although this interpretation is easy to formulate, it is difficult to perform inference in Bayesian neural networks (Gal, 2016). In this section, we focus on Bayesian neural networks by laying their foundations and illustrating some of the most common Bayesian models.

2.1.1 Bayesian Estimation

As uncertainties rely on the notion of measuring confidence, it is important to view the model from the lens of probabilistic modelling instead of the deterministic framework. While in the latter the optimization process consists of learning a point estimate of the model parameters (such as maximum-likelihood $\hat{\theta}_{\text{MLE}}$, or maximum-a-posteriori $\hat{\theta}_{\text{MAP}}$), the former opts to learn, from the training data $\mathcal{D}$, a probability distribution over the parameters rather than a point estimate. This is possible, in part, thanks to the formulation under the Bayes theorem (Theorem 2.1).

Theorem 2.1: Bayes theorem

Let's A and B be two events, with $\mathrm{P}(A)$ and $\mathrm{P}(B)$ being their respective probabilities such that $\mathrm{P}(B) \neq 0$, then:

$$\mathrm{P}(A \mid B) = \frac{\mathrm{P}(B \mid A)\mathrm{P}(A)}{\mathrm{P}(B)} \tag{2.1}$$

By applying Theorem 2.1 to the random variable $\boldsymbol{\Theta}$ representing the model parameters $\boldsymbol{\theta}$ and the available training data $\mathcal{D}$, we get for a given model:

$$\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{\mathrm{p}(\mathcal{D} \mid \boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})}{\mathrm{p}(\mathcal{D})} \tag{2.2}$$

Importantly, we emphasize that all the probabilities in Equation 2.2 are conditioned on a given model $\mathcal{M}$, characterized by the parameters and the parametric function (*i.e.* the choice of the architecture). For simplicity and unless mentioned otherwise, the dependency on the model will be omitted.

Equation 2.2 allows us to compute the *posterior distribution* $\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$ based on the *likelihood* $\mathrm{p}(\mathcal{D} \mid \boldsymbol{\theta})$, the *prior* $\mathrm{p}(\boldsymbol{\theta})$, and the *model-evidence* $\mathrm{p}(\mathcal{D})$. The likelihood describes to what extent the model fits the training dataset. The prior distribution is chosen beforehand, and it encodes our prior knowledge about the model and the distribution of the parameters. An in-depth examination of the priors will be presented later in Chapter 3. Finally, the model evidence measures on average the probability of observing the data given a randomly selected model from $\mathrm{p}(\boldsymbol{\theta})$.

From a practical perspective, the main challenge in Bayesian deep learning (BDL) is computing the posterior $\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$, which is intractable in most cases. Therefore, alternative approximation techniques are typically employed to estimate its value more efficiently (more on this in Section 2.1.3). Once the posterior distribution is computed, we would like to infer the output probability for a new datapoint x. This can be done by averaging the outputs $\mathrm{p}(y \mid x, \boldsymbol{\theta})$ produced by the model parameterized with $\boldsymbol{\theta}$:

$$\mathrm{p}(y \mid x, \mathcal{D}) = \mathbb{E}_{\boldsymbol{\theta} \sim \mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})}[\mathrm{p}(y \mid x, \boldsymbol{\theta})] \tag{2.3}$$

The inference in Equation 2.3 is often referred to in the literature as the *predictive posterior* or the *Bayesian model averaging* (BMA): $\mathrm{p}(y \mid x, \mathcal{D})$ is computed by marginalizing over the posterior $\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$ instead of relying on one single sample from the posterior distribution to predict the output y for the input x.

In our work, we mainly focus on the likelihood and prior components since the model evidence can be seen as a normalization coefficient. Furthermore, we will consider the special case when the dataset is generated *i.i.d.* (independent and identically distributed), for which the likelihood can be computed as the product of the probabilities associated to each input-output tuple. Under the *i.i.d.* assumption, the posterior distribution can be further simplified:

$$\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D}) \propto \mathrm{p}(\mathcal{D} \mid \boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}) = \mathrm{p}(\boldsymbol{\theta}) \times \prod_{(x,y) \in \mathcal{D}} \mathrm{p}(y \mid x, \boldsymbol{\theta}) \tag{2.4}$$

Deterministic models. Linking the Bayesian framework to the deterministic formulation can be done, thanks to Equation 2.3, by considering the posterior distribution as a Dirac distribution

centered around a point estimator $\hat{\theta}_{\mathcal{D}}$ learned from the data $\mathcal{D}$:

$$\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D}) = \delta(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\mathcal{D}}) \implies \mathrm{p}(y \mid x, \mathcal{D}) = \mathrm{p}(y \mid x, \hat{\boldsymbol{\theta}}_{\mathcal{D}})$$

This is the case for common point estimators such as the *MLE* and the *MAP* estimators:

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} \in \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left(\mathrm{p}(\mathcal{D} \mid \boldsymbol{\theta})\right) \quad ; \quad \hat{\boldsymbol{\theta}}_{\text{MAP}} \in \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left(\mathrm{p}(\mathcal{D} \mid \boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})\right)$$

Intuitively, Figure 2.1 illustrates a neural network trained as a deterministic model and under the Bayesian formalism. In the former, the learning process consists of finding the best point estimates by minimizing a loss function, whereas in the latter each connection in the network is characterized by a probability distribution (posterior) computed from the data.

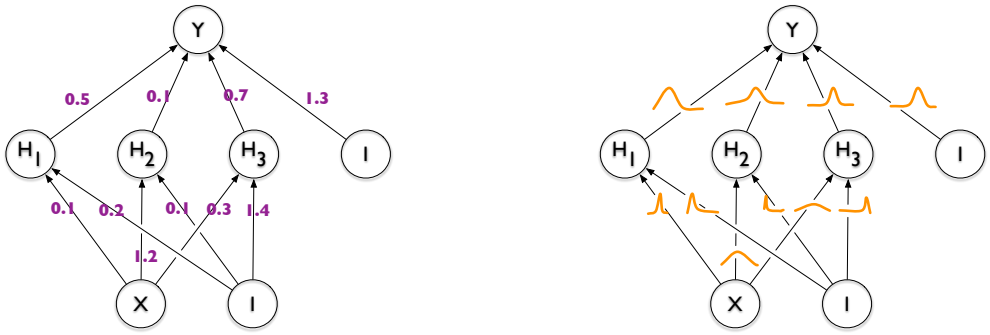


FIGURE 2.1: Visualization of a standard point estimator (left) and model with distribution over the weights (right). Credits to Blundell et al. (2015).

2.1.2 Bayesian Model selection and the Occam's Razor principle

The model evidence $\mathrm{p}(\mathcal{D})$ is also referred to in the literature as marginal likelihood since it can be computed by integrating over the prior distribution:

$$\mathrm{p}(\mathcal{D}) = \mathbb{E}_{\boldsymbol{\theta} \sim \mathrm{p}(\boldsymbol{\theta})}[\mathrm{p}(\mathcal{D} \mid \boldsymbol{\theta})] \tag{2.5}$$

MacKay (1991) argues that the model evidence could be used for model selection highlighting the existence of a correlation between the evidence and the generalization error. However, within the same paper, the author questions the generalization of selecting models based only on the model evidence, since for example this correlation is poorer when weight decay penalty is employed. Furthermore, this correlation was further analyzed in Lotfi et al. (2022) where the authors show that model evidence can be inversely correlated with generalization.

The choice of the model architecture is not a trivial task, especially for deep learning models. Occam's razor is the principle suggesting that the simplest explanation is often the correct one. In the context of machine learning, simple models are often preferable to complex ones as they could generalize better to new data and are less prone to overfitting (MacKay, 1991; Neal, 1996). If the model is optimized using the maximum likelihood estimator, the use of a less complex model is perhaps more justified (Neal, 1996), as the risk of overfitting decreases while better satisfying the Occam's razor principle.

In contrast, Neal (1996) claims that with Bayesian models, one should not limit the complexity solely depending on the available training set, and worry about overfitting. This is explained by

the assertion that these models do not overfit in theory: once the choice of the prior and the model architecture result in good convergence and has good performance with many datapoints, the model should be correct with fewer datapoints as well. Therefore, for complex tasks, the Bayesian approach is to select the largest model, regardless of the size of the training set, with the only constraint being the computing resources.

2.1.3 Variational Bayes

When computing the BMA, one indispensable challenge is the marginalization over the posterior distribution, that is in most cases intractable and hard to compute analytically. To this end, it is practical to compute an estimate of the expectation, for instance, a *Monte Carlo estimate* (Proposition 2.1).

Proposition 2.1: (informal) Monte Carlo Simulation

Let X be a random variable with p its probability density function, and g some function. Given T *i.i.d.* datapoints sampled from p, $\{x_1, \dots, x_T\}$, we have thanks to the law of large numbers:

$$\hat{I} = \frac{1}{T}\sum_{i=1}^{T} g(x_i) \xrightarrow[T\to+\infty]{} \mathbb{E}[g(X)]$$

$\hat{I}$ is an unbiased estimator of $\mathbb{E}[g(X)]$.

Therefore, the BMA (Equation 2.11) can be approximated with by using a Monte Carlo estimate with T samples:

$$\mathrm{p}(y \mid x, \mathcal{D}) \approx \frac{1}{T}\sum_{i=1}^{T} \mathrm{p}(y \mid x, \boldsymbol{\theta}_i) \quad ; \forall i \in \{1, \dots, T\} \;\; \boldsymbol{\theta}_i \sim \mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D}) \tag{2.6}$$

Variational inference. Although the previous estimate alleviates the need to compute the integral of the BMA (Equation 2.3), intractable in most cases, the remaining challenge is how to compute, let alone sample from the posterior distribution $\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$. To this end, an efficient approach is to approximate the posterior distribution for Bayesian deep learning models. While different approximation techniques exist in the literature, our focus will be on *variational inference*. Applying variational inference to Bayesian inference consists of approximating the posterior distribution with a simpler distribution $q_{\boldsymbol{\omega}}(\boldsymbol{\theta})$, known as a *variational distribution*, which is parameterized with the *variational parameters* $\boldsymbol{\omega}$ such that it minimizes the Kullback-Leibler (KL) divergence:

$$\hat{\boldsymbol{\omega}} = \underset{\boldsymbol{\omega}}{\operatorname{argmin}} \left(\mathrm{D}_{\mathrm{KL}}(q_{\boldsymbol{\omega}}(\boldsymbol{\theta}) \,\|\, \mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D}))\right) \tag{2.7}$$

The minimization of the KL divergence (Equation 2.7) is equivalent to maximizing the evidence lower bound (ELBO) (Saul et al., 1996; Neal and Hinton, 1998), formally defined as:

$$\begin{aligned}
\mathrm{ELBO}(q_{\boldsymbol{\omega}}) &= \log(\mathrm{p}(\mathcal{D})) - \mathrm{D}_{\mathrm{KL}}(q_{\boldsymbol{\omega}}(\boldsymbol{\theta}) \,\|\, \mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})) && (2.8)\\
&= \mathbb{E}_{\boldsymbol{\theta}\sim q_{\boldsymbol{\omega}}(\boldsymbol{\theta})}[\log(\mathrm{p}(\mathcal{D} \mid \boldsymbol{\theta}))] - \mathrm{D}_{\mathrm{KL}}(q_{\boldsymbol{\omega}}(\boldsymbol{\theta}) \,\|\, \mathrm{p}(\boldsymbol{\theta})) && (2.9)\\
&= \mathbb{E}_{\boldsymbol{\theta}\sim q_{\boldsymbol{\omega}}(\boldsymbol{\theta})}[\log(\mathrm{p}(\mathcal{D} \mid \boldsymbol{\theta})) + \log(\mathrm{p}(\boldsymbol{\theta})) - \log(q_{\boldsymbol{\omega}}(\boldsymbol{\theta}))] && (2.10)
\end{aligned}$$

Given that we aim at finding ω that satisfies Equation 2.7, the log-*evidence* could be ignored in Equation 2.8:

$$\underset{\omega}{\operatorname{argmax}} \left(\text{ELBO}(q_\omega)\right) = \underset{\omega}{\operatorname{argmin}} \left(\mathrm{D}_{\mathrm{KL}}(q_\omega(\boldsymbol{\theta}) \,\|\, \mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D}))\right)$$

By shifting the problem from computing the posterior distribution to a variational inference setting, the marginalization in the learning process (Equation 2.4) is replaced with an optimization step and thus dealing with derivatives instead of integrals, which is easier in practice (Jordan et al., 1998; Gal, 2016).

When merging both the approximation resulting from variational inference and Monte Carlo estimate, BMA can be computed in practice as follows:

$$\mathrm{p}(y \mid x, \mathcal{D}) \approx \mathbb{E}_{\theta \sim q_{\hat{\omega}}(\boldsymbol{\theta})}[\mathrm{p}(y \mid x, \theta)] \tag{2.11}$$

$$\approx \frac{1}{T} \sum_{i=1}^{T} \mathrm{p}(y \mid x, \boldsymbol{\theta}_i) \quad ; \forall i \in \{1, \ldots, T\} \;\; \boldsymbol{\theta}_i \sim q_{\hat{\omega}}(\boldsymbol{\theta}) \tag{2.12}$$

This is another advantage to using variational inference, since it is easier to sample from the variational distribution, making the application of a Monte Carlo estimate more straightforward for the variational BMA (Equation 2.11).

Bayes by Backprop (Blundell et al., 2015) applies successfully variational inference to deep learning models over a Gaussian variational posterior $q_\omega(\boldsymbol{\theta})$ with a scale mixture of two Gaussian distributions for the prior $\mathrm{p}(\boldsymbol{\theta})$. The number of learnable parameters only doubles while the model is probabilistic, and we get an approximate distribution over the weights of the model instead of a point estimate. In Blundell et al. (2015), the ELBO (Equation 2.10) is approximated with its Monte Carlo estimate (Equation 2.13). By doing so, it is possible to optimize for ω with gradient descent. For instance, the reported results are with a diagonal Gaussian distribution (*i.e.* assuming independence of the parameters) as the variational posterior distribution for which, thanks to the re-parameterization trick (Kingma and Welling, 2013) and the chain rule, the gradient over the variational parameters ω could be easily computed, much like computing the gradient in standard deep learning models.

$$\text{ELBO}(q_\omega) \approx \frac{1}{T} \sum_{i=1}^{T} \Big(\log(\mathrm{p}(\mathcal{D} \mid \theta_i)) + \log(\mathrm{p}(\theta_i)) - \log(q_\omega(\theta_i)) \Big) \tag{2.13}$$

> **Remark** *(Posterior).* Unless mentioned otherwise, even in the case when variational inference is used, the variational posterior distribution $q_{\hat{\omega}}(\boldsymbol{\theta})$ will be referred to with the posterior distribution notation $\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$.

MC-Dropout. Stochastic regularization techniques (SRTs) are commonly used to regularize models by adjusting its outputs in order to reduce the risk overfitting for a given input in a stochastic way: two forward passes of the model on the same input results in two different outputs. One popular SRT is Dropout (Srivastava et al., 2014) that masks randomly a portion of the vector (hidden units for example) with a fixed probability. Gal (2016) demonstrates that optimizing a neural network that has Dropout layers can be seen as performing variational inference with a Bernoulli variational distribution, and hence this model can be considered as a BNN: a neural network trained with dropout can be considered a Bayesian neural network, inheriting all the properties associated with it. Consequently, we will explore in our work Monte Carlo Dropout (*MC-Dropout*) (Gal and Ghahramani, 2016) which refers to a model containing Dropout layers with

a combination of Monte Carlo estimate during inference time (Equation 2.12), the weights θ_i are the submodels resulting from applying different stochastic masks.

2.1.4 Gaussian processes

A powerful and flexible family of probabilistic models that has gained widespread use in various fields, especially in machine learning, is *Gaussian process*, formally defined in Definition 2.1. Perhaps one of the most influential books in the topic is the work of (Rasmussen and Williams, 2006).

Definition 2.1: Gaussian process

A family of random variables $(Z_t)_{t\in\mathbb{R}^+}$ is said to be a *Gaussian process* if, for any finite set of indices $(t_1, \ldots, t_k) \in \mathbb{R}^+$ with $k \in \mathbb{N}^+$, the joint distribution of the random variables $(Z_{t_1}, \ldots, Z_{t_k})$ is a multivariate Gaussian distribution.

Since multivariate normal distribution is closed under linear transformation, conditioning and marginalization, GPs are completely defined by their mean and covariance functions. For the notation, we say that a process $f(x)$ follows a GP defined by its mean function $m(x)$ and its covariance function $k(x, x')$:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')).$$

What makes GPs interesting is their capacity to provide an estimation of the uncertainties associated with the predictions through the covariance matrix.

It is common to define the covariance functions using kernels. The latter are generally positive definite functions that take two arguments and return a real value ($\mathbb{R}$). The choice of the kernel function is directly related to our prior beliefs about the subset of functions that describe the data and the similarity of a pair of inputs based on some distance measure.

Perhaps choosing a prior that encodes our knowledge in the case of BNNs is quite hard since the prior is defined in the weights (parameters) space, thus setting a prior for each layer of the model is not straightforward. On the other hand, the prior of the GPs is in the functions space, making its choice rather easier and interpretable. In fact, our assumptions about the adapted set of functions for a given task is translated by the choice of the kernel, and they enable a significant amount of flexibility.

Similarly to BMA for BNNs (Equation 2.3), Bayes theorem could be applied in the functions space for GPs providing an alternative view of the problem from the functions space:

$$\mathrm{p}(y \mid x, \mathcal{D}) = \mathbb{E}_{f\sim\mathrm{p}(f\mid\mathcal{D})}[\mathrm{p}(y \mid x, f)] \tag{2.14}$$

As aforementioned, $\mathrm{p}(y \mid x, \mathcal{D})$ is a multivariate normal distribution. However, some of the limitation of GPs are the time and space complexities: $\mathcal{O}(|\mathcal{D}|^3)$ and $\mathcal{O}(|\mathcal{D}|^2)$ respectively. This is mainly associated with the inversion and the storage of the $|\mathcal{D}| \times |\mathcal{D}|$ covariance matrix of the observations. Although some work tries to reduce these complexities and drawbacks (Snelson and Ghahramani, 2005; Wilson and Nickisch, 2015), they do not entirely tackle the limitations of GPs for large scale datasets.

Remark *(Bridging GPs and BNNs).* Given a BNN with a single hidden layer such that the weights are independent with a finite-variance prior. In his thesis, Neal (1996) showed that it converges, in the limit of infinite width, to a GPs. The work of Lee et al. (2018a) further demonstrated the same link in the case of deeper networks.

2.2 Deep Ensembles and Bayesian Models

Deep ensembles (Lakshminarayanan et al., 2017) were introduced as an alternative to BNNs with the advantage of being simple to implement, parallelizable, and with very few hyperparameters tuning. The fundamental goal of deep ensembles was to achieve a reliable quantification of the predictive uncertainty (refer to Section 2.3). One main advantage of deep ensembles is to achieve improved accuracy (Ovadia et al., 2019), which grows in correlation with the increasing number of models in the deep ensemble. In our work, in addition to their known performance in terms of accuracy, we will focus as well on the uncertainties computed with this type of models.

2.2.1 The controversy over ensembles as Bayesian

There has been a large debate on whether the term "Bayesian" could be rightfully associated to deep ensembles. On one hand, ensembles are considered non-Bayesian approaches (Lakshminarayanan et al., 2017; Malinin and Gales, 2018; Ovadia et al., 2019) as they differ from classical Bayesian methods such as variational inference or Markov chain Monte Carlo (MCMC). Indeed, a deep ensemble, once trained, consists of multiple point estimates that are generally distinct due to the stochasticity of the initialization of the models and the optimization process. Therefore, the deep ensemble framework could be seen different from the Bayesian formulation.

On the other hand, deep ensembles could be seen as performing Bayesian model averaging, and they have an inherent Bayesian interpretation (Wilson and Izmailov, 2020; Izmailov et al., 2021; Kirsch et al., 2021): Equation 2.12 is true for deep ensembles with $\boldsymbol{\theta}_i$ being the weights of the model i in the ensemble after training. In addition, Izmailov et al. (2021) find that deep ensembles approximate well the posterior distribution, and are closer to the "Bayesian ideal" than most of the common and well accepted approximate Bayesian models, such as MC-Dropout. Moreover, it is argued that, since most the approaches designated as Bayesian provide in fact approximate inference due to the intractability of the posterior distribution, ensembles could deservedly be considered as Bayesian approaches. Furthermore, (Wilson and Izmailov, 2021) believes that the way the literature is being divided into "Bayesian" vs "non-Bayesian" is quite arbitrary.

In the case of deep ensembles, one can argue that Equation 2.12 is applicable, and that it is equivalent to sampling from an implicit variational posterior distribution $q_{\hat{\omega}}(\boldsymbol{\theta})$ the submodels $\boldsymbol{\theta}_i$ of the deep ensemble. In this thesis, we do not oppose ensemble to BDL. Instead, we study these two methods within the same framework, without feeling the need to make a clear distinction. In what follows, we will also occasionally use the term Bayesian methods in a very loose sense, including ensembles, even if this assumption remains questionable from a theoretical point of view. Finally, a connection could be made between MC-Dropout and deep ensembles: submodels are sampled from the model with Dropout layer, thus the weight are shared by the submodels (Gal, 2016; Lakshminarayanan et al., 2017).

2.2.2 Different techniques to ensembling in DL

Even though deep ensembles allow achieving state-of-the-art performances on different tasks and across a variety of tasks, they come with some drawbacks. Specifically, both training and inference times, along with the storage requirements for the models in the ensemble, grow linearly with the number of models in the ensemble. This has given rise to several initiatives aimed at mitigating their inherent limitations, some of which will be explored in the next part of the discussion.

TreeNet (Lee et al., 2015)**.** Motivated by the success of deep ensembles, Lee et al. (2015) explores the best strategies to create ensembles. Especially, they focus on reducing the parameters count for an ensemble thanks to parameter sharing, and introduce a "diversity-encouraging" loss function.

The former contribution is driven by analyzing in the computer vision field (for CNN models) demonstrating that lower layers tend to learn simple features (Girshick et al., 2014; Zeiler and Fergus, 2014; Lenc and Vedaldi, 2015), which are comparable between models in the ensemble. As a result, *TreeNet* was introduced as an ensemble with multiple "heads" and where the lower layers are shared between all these heads (Figure 2.2). The latter contribution consists of taking into account the predictions of the ensemble in the learning process rather than only considering the average, and thus reducing the diversity of the ensemble. Therefore, the authors use an *oracle set-loss* which is driven by the *multiple choice learning* (Guzmán-rivera et al., 2012). In a nutshell, the oracle set-loss associated with a single input is the cross-entropy loss of the most correct predictor, with the possibility to sum/average the cross-entropy losses of the top-k correct predictors. Notably, TreeNet trained with the oracle set-loss outperformed the deep ensemble in the paper benchmarks.

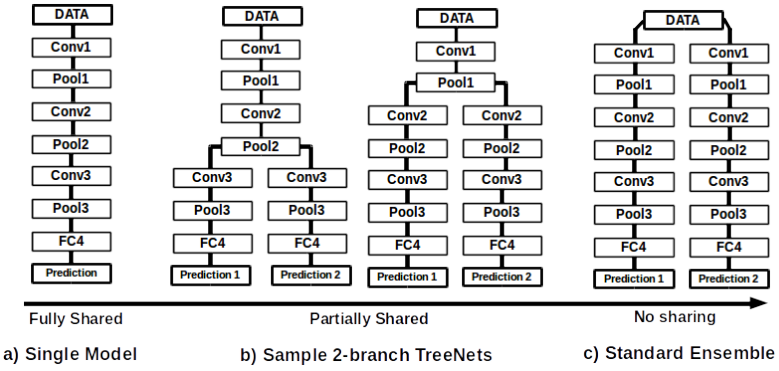
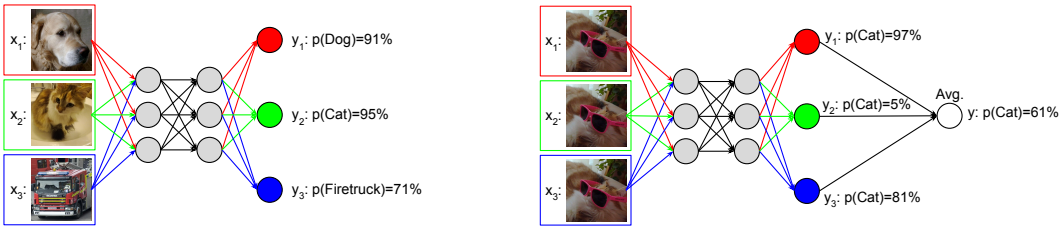


FIGURE 2.2: Illustration of TreeNet. Credits to Lee et al. (2015).

MIMO (Havasi et al., 2021). Another approach to tackle the aforementioned limitations of deep ensemble is from the neural network pruning literature[4], where it's possible to reduce the parameter count by up to 90% without compromising accuracy. Especially, the *lottery ticket hypothesis* (Frankle and Carbin, 2019) which states that within large, randomly initialized neural networks, there are smaller subnetworks (referred to as *winning tickets*) that can achieve similar or better performance compared to the original full network when trained, often more efficiently. Multi-inputs multi-outputs (MIMO) build upon this hypothesis claiming that a neural network is able to fit a range of independent winning tickets simultaneously. From a practical perspective, only the input and output layers are modified to allow feeding the network M inputs and getting their respective outputs (as illustrated in Figure 2.3). Once the network is trained, we can get an ensemble-like output by feeding the same input to the M subnetworks of MIMO.



(A) Training (B) Testing

FIGURE 2.3: Illustration of MIMO in the training and test modes. Credits to Havasi et al. (2021).

[4]For an extensive overview about neural network pruning, refer to (Blalock et al., 2020).

Packed-Ensemble (Laurent et al., 2022)**.** Designing an ensemble in a compact manner could solve its computational inefficiency. Deliberately, and contrary to MIMO, the subnetworks in Packed-Ensemble are not shared. This is perhaps due to the motivation of the authors to imitate a deep ensemble to the fullest extent. To achieve comparable results to a deep ensemble with Packed-Ensembles, not only the parameters' count is reduced, but also the number of forward passes as it only requires a single forward pass (due to the compactness of the model) to produce the outputs of the ensemble (where the number of forward passes is equivalent to the number of its elements). Thanks to the use of grouped convolutions (Krizhevsky et al., 2012), the authors mainly focused on ensembling the convolution layers of DL models. The results show that the proposed ensemble yield comparable results in deep ensembles with fewer parameters and while being faster.

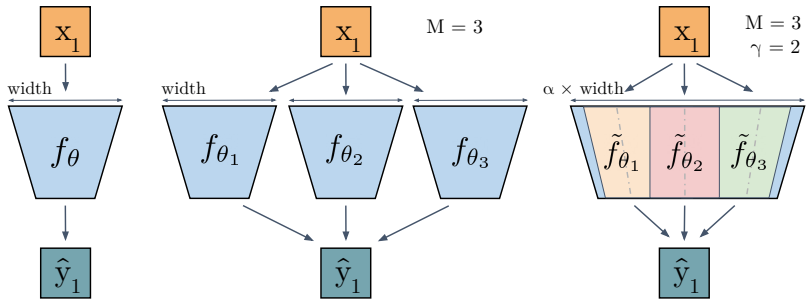


FIGURE 2.4: Illustration of Packed-Ensemble (right) compared to a single model (left) and a deep ensemble (middle). Credits to Laurent et al. (2022).

2.3 Exploring Uncertainty: Types and Quantitative Approaches

When evaluating a model prediction, in the case of classification for example, one can rely solely on the predicted class. By doing so, the confidence of the model on its predictions is ignored completely. Unless the model is deployed in a strictly controlled environment, this approach is far from being ideal. Evaluating the confidence of a deep learning model is becoming more and more crucial. Some relevant applications include, but are not limited to: AI safety (Amodei et al., 2016), autonomous driving (Kendall and Gal, 2017; Franchi et al., 2022), active learning (Settles, 2012), medical use cases (such as radiology) (Lambert et al., 2022).

With standard models, it is common to rely on softmax-probabilities as an alias of prediction confidence. However, this is problematic for a multitude of reasons. Firstly, deep learning models are proved to be overconfident, making the probabilities overestimated. Additionally, it was shown in Nguyen et al. (2015) that models are easily fooled and can predict confident predictions with the dominant class having an almost certain probability score, on both irregular images similar to random noise, and on regular images with compressible patterns such as repetition and symmetry. Therefore, a reliable measure of uncertainty of the model prediction is important.

One advantage of using Bayesian models is their ability to evaluate and distinguish between different sources of uncertainty. Achieving this becomes possible thanks to the posterior distribution over the weights $p(\boldsymbol{\theta} \mid \mathcal{D})$. This gives rise to the decomposition of total uncertainty as the sum of aleatoric and epistemic uncertainties (Senge et al., 2014; Gal, 2016; Kendall and Gal, 2017; Depeweg et al., 2018; Hüllermeier and Waegeman, 2021). In fact, Senge et al. (2014) were one of the first to advocate for the distinction of epistemic and aleatoric uncertainties for machine learning applications.

Total uncertainty. It measures the entire uncertainty associated with a given input samples. Interestingly, it can be decomposed into two distinct main parts: irreducible and reducible.

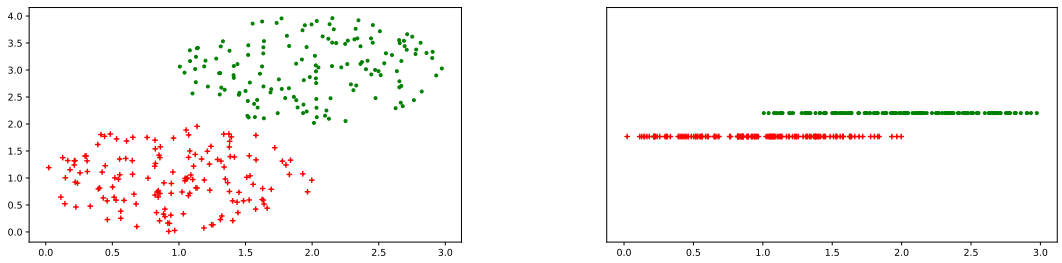
Aleatoric uncertainty. One source of uncertainty depends on the data and arises from the inherited noise in the generating process of the data and/or its acquisition. It is alternatively known as *data uncertainty*, and it is irreducible as it originates from the data. In theory, ambiguous noisy samples should yield high aleatoric uncertainty. Without further modifications, plain (calibrated) softmax classifiers could capture very well aleatoric uncertainty while they fail to distinguish the other types of uncertainties (Henning et al., 2021). It can be seen as well as the variability of the outcomes resulting from the randomness of the data-generating process (Senge et al., 2014).

Epistemic uncertainty. Also referred to as *model uncertainty* and is related to the lack of knowledge about the true state of the model parameters $\boldsymbol{\theta}$, more exactly, the best matching state (*i.e.* with the least inductive bias[5]). It measures the spread of the outputs $\mathrm{p}(y \mid x, \boldsymbol{\theta})$ with $\boldsymbol{\theta} \sim \mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$, and therefore the disagreement of the different outcomes. Contrary to aleatoric uncertainty, this type of uncertainty can be reduced by observing more samples, resulting in a shrinkage of the posterior distribution (Depeweg et al., 2018). Our work will focus mainly on this type of uncertainty and properties it should satisfy (Section 4.2). Notably, softmax classifiers do not capture epistemic uncertainty as this source of uncertainty relies on the posterior distribution and the notion of prior. Additional information will be addressed subsequently.

2.3.1 Uncertainties and dependencies

Based on their definitions, aleatoric and epistemic uncertainties could seem completely independent, however, this is not the case. Inspired by the work of Hüllermeier and Waegeman (2021), we will investigate the dependence of these uncertainties.

To showcase the interdependence between aleatoric and epistemic uncertainties, we study a simple two-dimensional problem of two linearly separated clusters (classes), as illustrated in Figure 2.5a. On each cluster, the points are sampled *i.i.d.* from the uniform distribution over the cluster disk. The two circles do not overlap, and thus we do not expect any sort of aleatoric uncertainty while separating the two classes with a simple classifier. In addition, epistemic uncertainty is very low in this case as, for example, a simple linear model could solve the classification problem easily and small changes in the slope coefficient will also lead to a perfect classification as well. However, projecting the circles on the x-axis (equivalently on the y-axis) results in an overlap of the classes and thus a high aleatoric uncertainty (Figure 2.5b).



(A) data in 2D

(B) 1D projection (x-axis)

FIGURE 2.5: Visualization of points drawn uniformly on two disks (left) and their projection on the x-axis (right). The offset on the y-axis between the two clusters is added for the 1D representation only for visibility.

[5]The prior assumptions encoded in the model about the underlying data distribution.

As a result, embedding the data in higher dimensions will likely reduce aleatoric uncertainty while increasing epistemic uncertainty, as the model will need more datapoints to accurately separate the two clusters, if possible. It is worth mentioning that when talking about higher dimensions, we refer mainly to informative and non-redundant dimensions (*i.e.* having access to non-observed features) as these could cause the curse of dimensionality. This could be the case, for example, if additional dimensions are added in our 2D data. We will explore this property in depth from a theoretical and a practical point of view later in this thesis, especially when comparing smaller and larger configurations of the same models.

Besides, it has been shown by Valdenegro-Toro and Saromo (2022) that aleatoric and epistemic uncertainty interact and are not completely independent. However, de Jong et al. (2024) challenge this finding regarding whether this interdependence is inherent to the data used or it is attributed to uncertainty disentanglement (more on this in Section 2.3.4).

2.3.2 Decomposing uncertainties

As shown previously, total uncertainty consists of two important components: aleatoric and epistemic. An interdependence was also discussed raising the question on how to compute these uncertainties in the case of a Bayesian neural network. Depeweg et al. (2018) discuss the decomposition of uncertainties based on information theory, which was mentioned in (Houlsby et al., 2011) without assigning the terms to the corresponding source of uncertainty. We will focus on quantification measures inspired by information theory and will discuss further some alternatives.

The sources of uncertainties could be illustrated by developing Equation 2.3:

$$\underbrace{\mathrm{p}(y \mid x, \mathcal{D})}_{\text{Total}} = \int \underbrace{\mathrm{p}(y \mid x, \boldsymbol{\theta})}_{\text{Aleatoric}} \underbrace{\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})}_{\text{Epistemic}} d\boldsymbol{\theta} \tag{2.15}$$

Total uncertainty. First, the total uncertainty $\mathcal{U}^t_{\mathcal{D}}(x)$, for an input x, is associated with the predictive posterior $\mathrm{p}(y \mid x, \mathcal{D})$ which is computed by the marginalization resulted from the BMA (Equation 2.3). This uncertainty is quantified as the entropy $\mathbb{H}$ of the predictive posterior, and in some way, it measures the uncertainty of the average prediction of the model:

$$\begin{aligned} \mathcal{U}^t_{\mathcal{D}}(x) &= \mathbb{H}(Y \mid x, \mathcal{D}) \\ &= -\sum_y \mathrm{p}(y \mid x, \mathcal{D}) \log\left(\mathrm{p}(y \mid x, \mathcal{D})\right) \end{aligned} \tag{2.16}$$

Aleatoric uncertainty. In contrast, aleatoric uncertainty $\mathcal{U}^a_{\mathcal{D}}(x)$ is computed as the conditional entropy of Y given $\boldsymbol{\Theta}$. This is equivalent to averaging the entropies of the output of "each" possible model parameterized with $\boldsymbol{\theta}$:

$$\begin{aligned} \mathcal{U}^a_{\mathcal{D}}(x) &= \mathbb{H}(Y \mid \boldsymbol{\Theta}, x, \mathcal{D}) \qquad (2.17) \\ &= -\sum_y \int \mathrm{p}(y, \boldsymbol{\theta} \mid x, \mathcal{D}) \log\big(\mathrm{p}(y \mid \boldsymbol{\theta}, x, \mathcal{D})\big) d\boldsymbol{\theta} \\ &= -\int \mathrm{p}(\boldsymbol{\theta} \mid x, \mathcal{D}) \Big(\sum_y \mathrm{p}(y \mid \boldsymbol{\theta}, x, \mathcal{D}) \log\big(\mathrm{p}(y \mid \boldsymbol{\theta}, x, \mathcal{D})\big)\Big)\, d\boldsymbol{\theta} \\ &= -\int \mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D}) \mathbb{H}(Y \mid x, \boldsymbol{\theta})\, d\boldsymbol{\theta} \\ &= \mathbb{E}_{\boldsymbol{\theta} \sim \mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})}[\mathbb{H}(Y \mid x, \boldsymbol{\theta})] \qquad (2.18) \end{aligned}$$

Epistemic uncertainty. Finally, the dependence between Y and $\boldsymbol{\Theta}$, given the training set $\mathcal{D}$, could be quantified through *(conditional) mutual information* $\mathbb{I}$ (Equation 2.19), which is equivalent to computing the different between $\mathcal{U}^t_{\mathcal{D}}(x)$ and $\mathcal{U}^a_{\mathcal{D}}(x)$, thanks to the symmetry of the mutual information exploited by Houlsby et al. (2011). This decomposition makes computing epistemic uncertainty easy and practical since it is straightforward to compute the entropies in the output space rather than in the weights space:

$$\begin{aligned}
\mathcal{U}^e_{\mathcal{D}}(x) &= \mathbb{I}(\boldsymbol{\Theta}; Y \mid x, \mathcal{D}) && (2.19)\\
&= \mathbb{H}(\boldsymbol{\Theta} \mid \mathcal{D}) - \mathbb{H}(\boldsymbol{\Theta} \mid Y, x, \mathcal{D}) && (2.20)\\
&= \mathbb{H}(Y \mid x, \mathcal{D}) - \mathbb{H}(Y \mid \boldsymbol{\Theta}, x, \mathcal{D}) && (2.21)\\
&= \mathcal{U}^t_{\mathcal{D}}(x) - \mathcal{U}^a_{\mathcal{D}}(x) && (2.22)
\end{aligned}$$

Proposition 2.2: Uncertainties bounds

The uncertainties $\mathcal{U}^t_{\mathcal{D}}(x), \mathcal{U}^a_{\mathcal{D}}(x)$ and $\mathcal{U}^e_{\mathcal{D}}(x)$ are bounded between 0 and $\log(C)$ with C being the number of classes in the classification problem.

Epistemic uncertainty is maximal when total uncertainty is maximal and aleatoric uncertainty is minimal. Indeed, this is the case when the model is uncertain on average, and each parameterization of the model yields a confident prediction. Therefore, mutual information measures disagreement between the sampled models from $\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$, and thus the disagreement of the outputs.

Remark *(Model dependence).* The uncertainty measures are dependent on a model $\mathcal{M}$: $\mathcal{U}^t_{\mathcal{D},\mathcal{M}}(x), \mathcal{U}^a_{\mathcal{D},\mathcal{M}}(x)$ and $\mathcal{U}^e_{\mathcal{D},\mathcal{M}}(x)$. For simplicity and in the absence of ambiguity, the model notation will be ignored.

The importance of reliably quantifying uncertainties. The evaluation of the performance of a model in terms of accuracy is straightforward. However, the same cannot be said about benchmarking the uncertainty of a model. We believe that in the absence of a proper baseline for uncertainties, one should quantify the quality of the measured uncertainties based on some score as it is the case with model calibration (Section 3.3), or at least based on how they satisfy a set of theoretical properties, especially in the case of epistemic uncertainty (Section 4.2). On top of that, the visualization of uncertainties could further highlight the importance of evaluating the different types of uncertainties and understanding their implications and aims (Section 2.5).

2.3.3 Computing uncertainties in practice

Since our focus is mainly on BMA methods such as MC-Dropout and Deep Ensembles, computing uncertainties relies on Monte-Carlo estimator for the expectations in regard to $\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$. In the case of MC-Dropout, the dropout masks are kept stochastic at test time and T forward passes are computed with therefore different "models". This is similar to sampling T models from the variational posterior distribution (Equation 2.12). Similarly, due to the stochasticity of the training step, the models in a Deep Ensemble are diverse and considered samples from some unknown posterior distribution. In this case, the number of samples T is the number of models in the deep ensemble. Accordingly, $\boldsymbol{\theta}_i$ will refer to a sampled model in the case of MC-Dropout or a member of the ensemble in the case of deep ensembles.

The integrals in the definitions of uncertainties could be approximated with Monte Carlo simulation as discussed in (Gal, 2016; Depeweg et al., 2018). By applying Proposition 2.1, we have

the estimates of the uncertainties for T samples:

$$\mathrm{p}(y \mid x, \mathcal{D}) \approx \hat{\mathrm{p}}(y \mid x, \mathcal{D}) = \frac{1}{T}\sum_{i=1}^{T}\mathrm{p}(y \mid x, \boldsymbol{\theta}_i) \tag{2.23}$$

$$\mathcal{U}^t_{\mathcal{D}}(x) \approx \hat{\mathcal{U}}^t_{\mathcal{D}}(x) = -\sum_{y}\hat{\mathrm{p}}(y \mid x, \mathcal{D})\log\left(\hat{\mathrm{p}}(y \mid x, \mathcal{D})\right) \tag{2.24}$$

$$\mathcal{U}^a_{\mathcal{D}}(x) \approx \hat{\mathcal{U}}^a_{\mathcal{D}}(x) = -\frac{1}{T}\sum_{i=1}^{T}\sum_{y}\mathrm{p}(y \mid x, \boldsymbol{\theta}_i)\log\left(\mathrm{p}(y \mid x, \boldsymbol{\theta}_i)\right) \tag{2.25}$$

$$\mathcal{U}^e_{\mathcal{D}}(x) \approx \hat{\mathcal{U}}^t_{\mathcal{D}}(x) - \hat{\mathcal{U}}^a_{\mathcal{D}}(x) \tag{2.26}$$

Remark *(Uncertainties in practice).* In our experiments, the above approximations will be used to measure and report the different types of uncertainties.

2.3.4 Importance of information theory amid criticism

In this section, we will shortly review the commonly used uncertainty decompositions and address several objections regarding the use of the information theoretical decomposition (Section 2.3.2).

Information theoretical decomposition. In Section 2.3.2, we discussed how the different sources of uncertainties are computed based on the information theory framework (reintroduced in Equation 2.27). Although there exist different formulations allowing this decomposition, we argue that the information theoretical approach has its merits and will be used in this work, until mentioned otherwise. Consequently, we will briefly highlight two additional and commonly used formulations of uncertainties: Gaussian logits and Bregman decompositions (Kendall and Gal, 2017; Depeweg et al., 2018; Hüllermeier and Waegeman, 2021; Lahlou et al., 2023; de Jong et al., 2024; Mucsányi et al., 2024).

$$\underbrace{\mathbb{I}(\boldsymbol{\Theta}; Y \mid x, \mathcal{D})}_{\text{Epistemic}} = \underbrace{\mathbb{H}(Y \mid x, \mathcal{D})}_{\text{Total}} - \underbrace{\mathbb{H}(Y \mid \boldsymbol{\Theta}, x, \mathcal{D})}_{\text{Aleatoric}} \tag{2.27}$$

Variance decomposition. It refers to the use of the variance (or its square root) instead of the mutual information as a measure of epistemic uncertainty. It is also possible to compute the variance of the logits instead of the (softmax) probabilities (Depeweg et al., 2018; de Jong et al., 2024) as illustrated in Equation 2.28 for the former.

$$\underbrace{\mathbb{V}_{\boldsymbol{\theta}\sim\mathrm{p}(\boldsymbol{\theta}\mid\mathcal{D})}[\mathbb{E}_Y[\mathrm{p}\left(Y \mid x, \boldsymbol{\theta}\right)]]}_{\text{Epistemic}} = \underbrace{\mathbb{V}_Y[\mathrm{p}\left(Y \mid x, \mathcal{D}\right)]}_{\text{Total}} - \underbrace{\mathbb{E}_{\boldsymbol{\theta}\sim\mathrm{p}(\boldsymbol{\theta}\mid\mathcal{D})}[\mathbb{V}_Y[\mathrm{p}\left(Y \mid x, \boldsymbol{\theta}\right)]]}_{\text{Aleatoric}} \tag{2.28}$$

Bregman decomposition. Founded on the *Bregman divergence*, Pfau (2013) formalized a generalized Bias-Variance decomposition. Lahlou et al. (2023) explore the fact that the KL-divergence is a special case of the Bregman divergence, if the convex function is chosen to be the negative entropy function, and analyze this decomposition from the lens of risks. They show that the *excess risk* is an appropriate measure of epistemic uncertainty in the sense that it properly reflects the model misspecification.

Benefits of using the Information Theoretic approach. Thanks to the boundedness of the Shannon entropy, epistemic uncertainty computed with the mutual information is consequently bounded. As a result, epistemic uncertainty in this case is interpretable since we can easily distinguish between an epistemically confident or uncertain prediction, which is not always the case with the variance decomposition, more precisely, it is not upper-bounded if computed from logits. In addition, the lower bound $\mathbb{I}(\boldsymbol{\Theta}; Y \mid x, \mathcal{D}) = 0$ is achieved if and only if $\boldsymbol{\Theta}$ and Y are independent conditionally to $(x, \mathcal{D})$, offering a comprehensive theoretical framework.

Critiques of mutual information. Building on the findings of (Lahlou et al., 2023; Wimmer et al., 2023; Mucsányi et al., 2024; Sale et al., 2024), many papers question the use of the mutual information as a reliable measure of epistemic uncertainty. The criticism toward the use of mutual information could be summarized into three main categories:

1. **Empirical incoherence:** When measured with the mutual information (Equation 2.22), epistemic uncertainty fails to verify, in practice, the reducibility principle in the presence of more datapoints as shown in Wimmer et al. (2023) and further explored in our work Fellaji and Pennerath (2023). Contrary to Wimmer et al. (2023), we do not attribute this incoherence to the information theoretical approach, but rather to the learning process, the loss function and the difficulty of calibrating epistemic uncertainty (more on that in Chapter 4).
2. **Model misspecification:** Defined as the difference between the Bayes predictor (which minimizes the risk for all the possible functions) and the optimal predictor (which minimizes the risk in the hypothesis space). Lahlou et al. (2023) challenge the assumption that deep learning models in practice, and more precisely (approximate) Bayesian methods, do not give rise to model misspecification. Especially, the misspecification arises in the low data regime and stems from the finite computational budget and the bias induced by the optimizer (SGD). However, Lahlou et al. (2023) question the use of the variance or the entropy of the posterior predictive as a measure of epistemic uncertainty, which is in fact a measure of total uncertainty.
3. **Uncertainty disentanglement:** Mucsányi et al. (2024) focus on studying the rank correlation of the two sources of uncertainty, with the ideal case being that they are disentangled and thus uncorrelated. They show that both the information theoretical framework and Bregman decomposition result in correlated uncertainties. The takeaway is that "there is no general uncertainty", meaning that the ideal uncertainty measure depends on the task. Nevertheless, as discussed in Section 2.3.1, aleatoric and epistemic uncertainty are not completely independent. However, when compared empirically to the variance decomposition in de Jong et al. (2024), the information theoretical decomposition seems to give a better disentanglement of epistemic and aleatoric uncertainties, even though the two decompositions are still inadequate to separate completely the two sources of uncertainty.

2.4 Probability of Probabilities

In deep learning classifiers, the softmax normalization function is widely used to map the logits (raw outputs of the model) to a probability vector. By doing so, the logits are projected into a simplex (Definition 2.2) in a deterministic manner. However, it was discussed that the softmax layer results in overconfident probabilities (Szegedy et al., 2015; Guo et al., 2017; Wilson and Izmailov, 2020). Although model calibration (Section 3.3) could solve the overconfidence of the softmax probabilities, it has some pitfalls such as ignoring the specific calibration of epistemic uncertainty. Therefore, exploring other approaches in the literature with a focus on epistemic uncertainty is encouraged. One emerging idea is to have a flexible mapping for the probability-normalization layer relying directly on the logits.

Definition 2.2: Simplex

In an n-dimensional space ($\mathbb{R}^n$), the $(n-1)$-simplex $\boldsymbol{\Delta}^{n-1}$ is defined as:

$$\boldsymbol{\Delta}^{n-1} = \left\{ \boldsymbol{z} = (z_1, \ldots, z_n) \in [0,1]^n \mid \sum_{i=1}^{n} z_i = 1 \right\}$$

In Sensoy et al. (2018); Malinin and Gales (2018), the authors endorse the idea of having a Dirichlet distribution (Definition 2.3) parameterized with the logits as the last normalization layer. Despite some differences motivating their works and how they formulated their ideas, the end goal of both papers is to have a model where the normalization layer explicitly parameterize a distribution over distributions on a simplex, and therefore the output is a second order probability distribution. These types of models are sometimes referred to as *Dirichlet-based uncertainty* (DBU) models (Deng et al., 2023). Since Dirichlet distribution models second order probabilities, these models embed epistemic uncertainty.

Definition 2.3: Dirichlet distribution

The Dirichlet distribution, in an n-dimensional space, is a continuous multivariate probability distribution parameterized with a vector of strictly positive values $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ and defined over the simplex $\boldsymbol{\Delta}^{n-1}$. The probability density function of $\texttt{Dir}(\boldsymbol{\alpha})$ is:

$$\forall \boldsymbol{x} \in \boldsymbol{\Delta}^{n-1},\ f(\boldsymbol{x}; \boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{n} x_i^{\alpha_i - 1}$$

with B is the multivariate Beta function. The value $\alpha_0 = \sum_i \alpha_i$ is called the *precision* of $\texttt{Dir}(\boldsymbol{\alpha})$.

For $\boldsymbol{Z} \sim \texttt{Dir}(\boldsymbol{\alpha})$, the expected value depends solely on $\boldsymbol{\alpha}$, hence, the expected probability is easily computed for these models given $\boldsymbol{\alpha}$:

$$\mathbb{E}[\boldsymbol{Z}] = \frac{\boldsymbol{\alpha}}{\alpha_0} \tag{2.29}$$

In addition, the different types of uncertainties can be computed in closed forms in the case of a Dirichlet distribution, making the measurement of uncertainties straightforward, without the need for approximations. Computing the different sources of uncertainty in the case of a second-order distribution, *i.e.* where the last layer is a Dirichlet distribution, will be discussed shortly in Section 2.4.3.

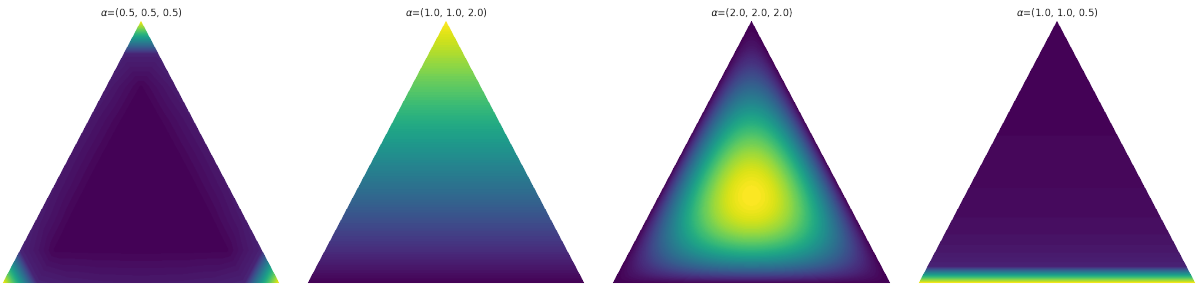


FIGURE 2.6: Visualization of $\texttt{Dir}(\boldsymbol{\alpha})$ for different values of $\boldsymbol{\alpha}$ in $\mathbb{R}^3$. Yellow represents higher concentrations.

In Figure 2.6, different configurations of $\boldsymbol{\alpha}$ are visualized in $\boldsymbol{\Delta}^2$. A sample from $\texttt{Dir}(\boldsymbol{\alpha})$ will more likely be in the yellow part of the heatmaps. Both $\texttt{Dir}(0.5,\ 0.5,\ 0.5)$ and $\texttt{Dir}(2,\ 2,\ 2)$ have the same expected value of $(1/3,\ 1/3,\ 1/3)$, however, a single sample from the first Dirichlet is most likely to be in one of the corners of the simplex whereas it is more likely to be in the center of the simplex for the second.

2.4.1 Evidential Deep Learning

Inspired by *subjective logic* (Jøsang, 1997), evidential deep learning (EDL) (Sensoy et al., 2018) aims at modeling a second-order probability and directly quantifying the notion of "beliefs" and uncertainty.

To illustrate the core idea of subjective logic (Jøsang, 1997), we start by the case of a binary classification. This logic models the lack of a perfect perspective of the world, encouraging a belief system approach based on the notion of *opinions*. For a proposition that could either be true of false, Jøsang (1997) uses the notation of *belief*, *disbelief* and *uncertainty*. An opinion ω, as illustrated in Figure 2.7, is an element $(b,\ d,\ u)$ from the simplex $\boldsymbol{\Delta}^2$, and it consists of a belief (b), disbelief (d) and ignorance/uncertainty (u). For an easy generalization to the multi-classes case, we consider the disbelief in the binary case as the belief for the second (negative) class. Therefore, the beliefs describe the certainty associated with the predictions, while the ignorance is the model uncertainty associated with these predictions. Additionally, this framework can be easily generalized to a multiclassification task with the opinion $(b_1,\ \ldots,\ b_C,\ u) \in \boldsymbol{\Delta}^C$. Thanks to its properties, Dirichlet distribution is a natural choice for the second-order distribution of the opinions on the simplex. Furthermore, Jøsang (1997) argues that it is possible, without any restrictions, to only consider $\texttt{Dir}(\boldsymbol{\alpha})$ such that $\alpha_i \geq 1$. We will discuss further on that the U-shaped Dirichlet distributions have interesting properties that could be useful (Chapter 5).

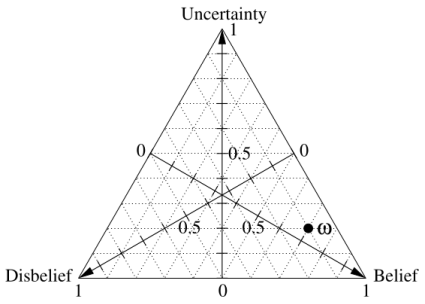


FIGURE 2.7: Example of one opinion ω in the subjective logic framework. Credits to Jøsang (1997).

The main idea of EDL consists of applying subjective logic to deep learning models, without making substantial changes to the model, and by incorporating uncertainty estimation. The *evidence* $(\boldsymbol{e})$ is defined as a positive transformation of the logits (raw outputs of the model), such as $\text{ReLU}(.)$ or $\exp(.)$, then the beliefs are computed based on the evidence of each input. For instance, in Sensoy et al. (2018), the evidence vector is obtained by applying $\text{ReLU}(.)$ to the logits. Finally, similar to Jøsang (1997), U-shaped Dirichlet distributions are excluded in EDL and thus: $\boldsymbol{\alpha} = \boldsymbol{e} + \mathbf{1}_C \geq 1$, with $\mathbf{1}_C$ is the vector of ones (of size C), and the beliefs being: $\boldsymbol{b} = \boldsymbol{e}/\alpha_0$ and $u = C/\alpha_0 \leq 1$.

The training of EDL only requires in-distribution (ID) samples, as it is the case with common deep learning models: the training data is sampled from the same "target" distribution. Different loss functions were tested, and the reported results are based on a loss function chosen empirically (the sum of squares loss). Furthermore, a regularization term is added to force unreliable predictions to have maximum uncertainty. It is important to mention that this additional regularization term is dependent on the true target, contrary to the common regularization penalties relying only on the parameters of the model and/or the model outputs. This is achieved by forcing the evidence, through the regularization term, to be null for all the classes but the true target, encouraging certainty for this class: $\text{D}_{\text{KL}}(\texttt{Dir}(y + (1-y) \odot \boldsymbol{\alpha}) \parallel \texttt{Dir}(\mathbf{1}_C))$, with $\odot$ is the pointwise product and y is the one-hot encoded true target.

However, Bengs et al. (2022) show that EDL does not encourage the model to predict the uncertainty term in the opinion $(u = 1 - \sum_i b_i)$ faithfully. This is due in part to the regularization

coefficient and the lack of an objective ground-truth for the uncertainty. Deng et al. (2023) further investigated EDL and its pitfalls and showed that samples with high data uncertainty are over-penalized with EDL when the targets are one-hot encoded. This leads to an undervaluation of the data uncertainty and hence affecting the quality of the uncertainty estimations. The proposed solution, I-EDL, involves the Fisher information matrix (FIM) which measures the significance of each samples in the training set and adjusts accordingly the objective loss terms. EDL assumes that the classes for a given input are drawn from an isotropic Gaussian distribution, which in practice is not the case as argued by Deng et al. (2023). We refer interested readers to the original papers for more comprehensive details.

2.4.2 Prior Networks

As discussed in Section 2.3, there are two main sources of uncertainties: aleatoric and epistemic. The work of Malinin and Gales (2018) is driven by two objectives: tackling the overconfidence of deep learning models, and further distinguish an additional source of uncertainty. Prior networks (PNs) tackle these two questions.

With Bayesian models, we have an implicit conditional distribution over distributions on a simplex thanks to the posterior distribution $\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$. Meanwhile, prior networks aim to explicitly parameterize this distribution over distributions. A natural choice to model a categorical distribution is the Dirichlet distribution, as it serves as a prior over the probabilities of the categorical distribution. Malinin and Gales (2018) refer to a prior network that parameterized a Dirichlet as *Dirichlet Prior Network* (DPN). Similar to Sensoy et al. (2018), the output of the model, after applying the positive mapping, are used to parameterize the Dirichlet distribution. However, and in contrast to Sensoy et al. (2018), the formalism of prior networks does not exclude the U-shaped Dirichlet distributions.

Distributional uncertainty is regarded as a component of model uncertainty in the Bayesian framework, and thus Malinin and Gales (2018) consider three types of uncertainties: aleatoric, model, and distributional uncertainties. The latter results from the discrepancy between the distributions of the test and training data, and referred to by *data shift* as well (Quiñonero-Candela, 2009). According to Malinin and Gales (2018), aleatoric (data) uncertainty is considered *"known-unknown"* since the model "knows" that a high uncertainty is associated with "unknown" (ambiguous) data. Under the same formalism, distributional uncertainty, the special component of epistemic, is high when the model could fail to classify the sample because it is drawn from an unknown distribution, hence it is considered *"unknown-unknown"*. Additionally, Deng et al. (2023) emphasize that distributional uncertainty cannot be disentangled from aleatoric and epistemic uncertainties in the case of BNNs.

The trainings of PNs and EDL are fundamentally different. PNs are trained with ID samples and also OOD (Out-of-distribution) samples, which are inputs drawn from a different distribution than of ID samples (more details on OOD in Section 6.1). For ID samples, the training is based on the targets of the training set, whereas the uniform vector is used as objective for the OOD samples. In addition, the $\exp(.)$ is used to map the logits to $\boldsymbol{\alpha}$ making the expected value of the Dirichlet distribution equal to the softmax function.

In terms of equations, by starting from the BMA (Equation 2.15) and designating the outputs by the categorical distribution $\boldsymbol{\mu}$, we can distinguish the three types of uncertainties: data $\mathrm{p}(y \mid \boldsymbol{\mu})$, distributional $\mathrm{p}(\boldsymbol{\mu} \mid x, \boldsymbol{\theta})$ and model $\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$ uncertainties:

$$\mathrm{p}(y \mid x, \mathcal{D}) = \iint \underbrace{\mathrm{p}(y \mid \boldsymbol{\mu})}_{\text{Aleatoric}} \underbrace{\mathrm{p}(\boldsymbol{\mu} \mid x, \boldsymbol{\theta})}_{\text{Distributional}} \underbrace{\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})}_{\text{Model}} d\boldsymbol{\mu}\, d\boldsymbol{\theta}$$

Remark *(Evaluating epistemic uncertainty).* Although the formalism of prior network distinguishes the three sources of uncertainties, the authors simplify the choice of the model by using a point-estimate $\hat{\boldsymbol{\theta}}$ and thus ignoring epistemic uncertainty:

$$\mathrm{p}(y \mid x, \mathcal{D}) \approx \mathrm{p}(y \mid x, \hat{\boldsymbol{\theta}}) = \mathbb{E}_{\mathrm{p}(\boldsymbol{\mu} \mid x, \hat{\boldsymbol{\theta}})}[\mathrm{p}(y \mid \boldsymbol{\mu})]$$

To fully benefit from the above distinction, it is possible to have Bayesian model. For the approximation of $\mathrm{p}(y \mid x, \mathcal{D})$ in this case, a nested Monte-Carlo sampling could be used: after sampling N parameters $\boldsymbol{\theta}_i$ from the posterior $\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$, one can sample M samples for each $\boldsymbol{\theta}_i$ from the Dirichlet distribution $\mathrm{p}(\boldsymbol{\mu} \mid x, \boldsymbol{\theta}_i)$.

2.4.3 Measuring uncertainties for a Dirichlet distribution

Having a Dirichlet distribution as the model output allows computing analytically, and in closed forms, the different types of uncertainty associated with second-order distributions. In a nutshell, instead of reasoning on the posterior distribution of the weights, we will focus on the Dirichlet distribution. We refer to (Deng et al., 2023, Appendix B) for more details on the following equations on *Dirichlet based uncertainties* (DBU).

Total uncertainty. The entropy of the expected value of the Dirichlet distribution, with $\boldsymbol{\alpha}$ computed as a function of the model and the input x:

$$\mathcal{U}^t_{\mathcal{D}}(x) = \mathbb{H}\Big(\mathbb{E}_{\boldsymbol{\theta} \sim \mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})}[\mathrm{p}(Y \mid x, \boldsymbol{\theta})]\Big) \implies \mathcal{U}^t_{\mathcal{D}, \mathrm{DBU}}(x) = \mathbb{H}\Big(\mathbb{E}_{\mu \sim \mathtt{Dir}(\boldsymbol{\alpha})}[\mathrm{p}(Y \mid \mu)]\Big)$$

Using the expectation of the Dirichlet distribution (Equation 2.29):

$$\mathcal{U}^t_{\mathcal{D}, \mathrm{DBU}}(x) = -\sum_{i=1}^{C} \frac{\alpha_i}{\alpha_0} \log\left(\frac{\alpha_i}{\alpha_0}\right) \tag{2.30}$$

Data uncertainty. It could be measured thanks to the expected entropy:

$$\mathcal{U}^a_{\mathcal{D}}(x) = \mathbb{E}_{\boldsymbol{\theta} \sim \mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})}[\mathbb{H}(Y \mid x, \theta)] \implies \mathcal{U}^a_{\mathcal{D}, \mathrm{DBU}}(x) = \mathbb{E}_{\mu \sim \mathtt{Dir}(\alpha)}[\mathbb{H}(Y \mid \mu)]$$

With $\psi(.)$ is the digamma function:

$$\mathcal{U}^a_{\mathcal{D}, \mathrm{DBU}}(x) = -\sum_{i=1}^{C} \frac{\alpha_i}{\alpha_0} \left(\psi(\alpha_i + 1) - \psi(\alpha_0 + 1)\right) \tag{2.31}$$

Distributional uncertainty. As discussed in Remark 2.4.2, DBU models do not estimate model uncertainty in the parameters-space since a single model is used. Similarly to epistemic uncertainty in the case of BNNs (Equation 2.22), distributional uncertainty ($\mathbb{I}(\boldsymbol{y}; \boldsymbol{\mu} \mid x, \mathcal{D})$) could be approximated then by the difference between total uncertainty and data uncertainty.

Definition 2.4: Differential Entropy

Let X be a continuous random variable and f its probability density function. The differential entropy of X can be seen as a generalization of the entropy, and is defined as:

$$\mathcal{H}(X) = \mathbb{E}[-\log(f(X))]$$

Differential entropy. An alternative measure of the distributional uncertainty is the differential entropy since it is more related to the sharpness of the Dirichlet distribution: shaper distributions yield low entropy whereas high values are associated with uniform distribution (Deng et al., 2023). It can be computed analytically for a distribution $\texttt{Dir}(\boldsymbol{\alpha})$:

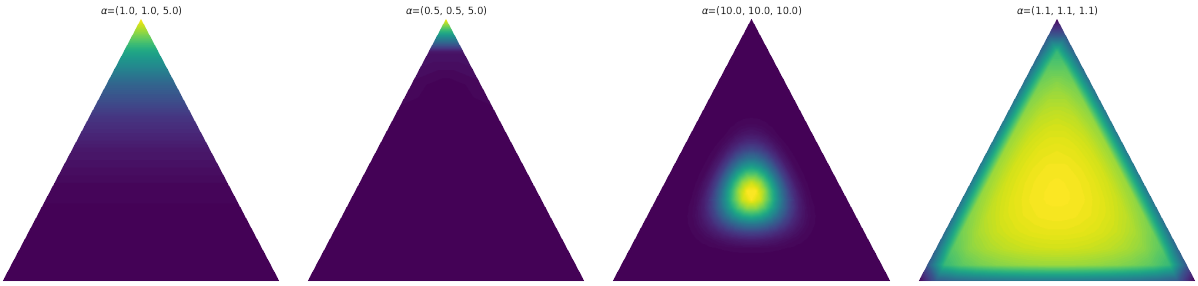
$$\mathcal{H}(\texttt{Dir}(\boldsymbol{\alpha})) = -\log(\Gamma(\alpha_0)) + \sum_{i=1}^{C} (\log(\Gamma(\alpha_0)) - (\alpha_i - 1)\,(\psi(\alpha_i) - \psi(\alpha_0)))$$

With $\Gamma(.)$ is the gamma function.

> **Remark** *(Epistemic uncertainty for DBU models in practice).* In our experiments, we will use distributional uncertainty as a proxy for epistemic uncertainty when DBU models are explored, thanks to its tractable value.

2.5 Visualizing uncertainties in the simplex

To further build the intuition behind the different sources of uncertainties, we expand on the work of Malinin and Gales (2018), by discussing the implications of the uncertainties in the output space and illustrating these consequences in the simplex. Furthermore, we link each case to possible values of the parameter of the Dirichlet distribution $\boldsymbol{\alpha}$, making the connection to the case where the model is learning a second-order distribution (Section 2.4).



(A) Confident prediction (B) Overconfident prediction (C) High data uncertainty (D) High epistemic uncertainty

FIGURE 2.8: Visualization of different sources of uncertainties.
Inspired by Malinin and Gales (2018).

For a confident prediction (Figure 2.8a), the Dirichlet distribution is centered around the same predicted class. Since $\boldsymbol{\alpha}$ are the unnormalized and positively mapped outputs of model, this translates in the model being highly "activated" for this specific class. The concentration around the predicted class is more pronounced when $\boldsymbol{\alpha}$ consists of scalars strictly below 1 for all classes but this class (Figure 2.8b). This property of Dirichlet distribution will be explored further.

The data uncertainty is presented in Figure 2.8c. It occurs when each model is predicting a uniform prediction and thus the mass of the Dirichlet distribution is centered in the middle of the simplex. It could be outlined by a Dirichlet distribution parameterized by an almost constant vector $\boldsymbol{\alpha} \approx (\alpha, \ldots, \alpha)$ with $\alpha \geq 1$. In addition, the higher α, the higher aleatoric uncertainty.

Finally, epistemic uncertainty captures the uncertainty about the posterior $p(\boldsymbol{\theta} \mid \mathcal{D})$ and hence embodies the diversity of the predictions. This disagreement is exemplified by diverse predictions such that every value in the simplex is plausible (Figure 2.8d) according to existing work (Malinin and Gales, 2018; Wimmer et al., 2023). Such an outcome is achieved in the context of a Dirichlet distribution characterized, again, with an almost *symmetric* Dirichlet distribution, with α around

1. This is the case for example in Malinin and Gales (2018) where a high epistemic uncertainty is expressed with the uniform distribution on the simplex, thus $\alpha = 1$ which results in the maximum differential entropy.

We believe that the disagreement of the predictions is not a sufficient condition for a high epistemic uncertainty: *the predictions should be in disagreement, yet each must maintain a high degree of certainty in its outcome.* Even though epistemic uncertainty is higher in Figure 2.8d compared to Figure 2.8c due to the flatness of the distribution, these two examples reflect a high aleatoric uncertainty, as we are dealing with symmetric Dirichlet distributions with $\alpha > 1$. However, epistemic uncertainty is more related to $0 < \alpha \leq 1$, and since differential entropy reaches its upper bound in the case of a uniform distribution, we argue that the expected evolution of epistemic uncertainty related only to the disagreement of the outputs, as discussed previously, is more related to meaure of differential entropy rather than distributional uncertainty. We will critically examine and address this status quo later in this thesis from the perspectives of uninformative priors (Chapter 5).

2.6 Modeling Epistemic Uncertainty

As aforementioned, deterministic models, with a single forward pass, do not allow disentangling epistemic and aleatoric uncertainties by default and only capture the latter. A prominent direction of research focus on enriching these models with epistemic uncertainty estimation through a secondary model. In order to obtain the desirable outcome with the secondary model, the feature-space should satisfy a few properties.

2.6.1 Feature collapse

Perhaps one pitfall of deterministic models and deep learning models in general is that the notion of distance is not guaranteed in the feature-space. (Amersfoort et al., 2020) were the first to use the term *feature collapse* to refer to OOD samples being mapped in the same manifold as ID samples in the feature-space. The model in this case is insensitive to changes in the inputs when studied in the feature-space, and hence they are indistinguishable in the representation space. In order to avoid feature collapse, the model is required to be *sensitive*. However, a model that is highly sensitive is likely not *smooth*: the model exhibits abrupt and unpredictable changes in its output in response to small variations in the input data, leading to poor generalization and potential overfitting.

Balancing smoothness and sensitivity is key to building models that avoid feature collapse while maintaining smoothness. In theory, these concepts are ensured by imposing a *bi-Lipschitz* constraint (Proposition 2.3) to the feature mapping (Amersfoort et al., 2020; Liu et al., 2020a; Mukhoti and Kirsch, 2023). The lower bound in Equation 2.32 makes sure that the model (more precisely the feature extractor) is sensitive, whereas the upper bound ensures smoothness.

Proposition 2.3: bi-Lipschitz constraint

Let g be a function mapping inputs to the feature space. g satisfies the bi-Lipschitz condition if:

$$\exists\, 0 < L_1 < 1 < L_2 \quad \forall (x_1, x_2)\,, \quad L_1 \|x_2 - x_1\| \leq \|g(x_2) - g(x_1)\| \leq L_2 \|x_2 - x_1\| \tag{2.32}$$

In the following, we will discuss some methods that focus on uncertainty modeling with a smooth and sensitive feature space.

2.6.2 Deterministic Uncertainty Quantification

Deterministic Uncertainty Quantification (DUQ) (Amersfoort et al., 2020) uses RBF networks that combine a deep learning model as a feature extractor and an RBF network. They were the first to showcased that the RBF networks, once stabilized, could yield competitive accuracies to softmax networks while giving a reliable uncertainty estimate. The training forces the feature mapping to be sensitive, allowing the RBF to measure the uncertainty of the prediction as the distance to the centroids of the classes. The RBF is parameterized with a learnable weight matrix per class and the RBF hyperparameter. The model parameters, the RBF weight matrix and the centroids are jointly learned during training.

To guarantee the sensitivity in the feature space, the authors used *gradient penalty* which was first referred to as double backpropagation in Drucker and Le Cun (1992). Intuitively, the goal is to regularize the norm of the Jacobian matrix which leads to the regularization of the Lipschitz constant. This is possible when the norm of the Jacobian is bounded (Equation 2.33). Amersfoort et al. (2020) discussed one-sided penalty and two-sided penalty with the latter resulting, empirically, in better sensitivity and generalization. Furthermore, DUQ has competitive performance on the OOD detection task by using a single model and a single forward pass.

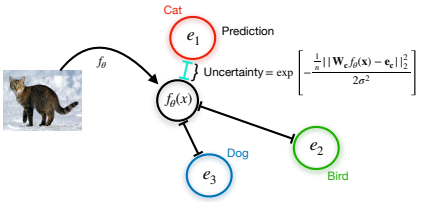


FIGURE 2.9: Illustration of the DUQ architecture. Credits to Amersfoort et al. (2020).

$$g(x+\epsilon) - g(x) \approx \epsilon \nabla_x g(x) \implies \|g(x+\epsilon) - g(x)\| \leq \sup\left(\|\nabla_x g\|\right) \|\epsilon\| \tag{2.33}$$

2.6.3 Spectral-normalized Neural Gaussian Process

Similar to DUQ, the goal of Spectral-normalized Neural Gaussian Process (SNGP) (Liu et al., 2020a) is to estimate in a single pass a reliable measure of uncertainty by building upon a smooth and sensitive feature mapping, referred to as *distance awareness*. The regularization of the feature mapping is however done differently: instead of using gradient penalty, SNGP relies on spectral normalization. Especially, for layers that have residual skip connections $h(x) = x + \text{block}(x)$, such that block(.) is α-Lipschitz with $0 < \alpha \leq 1$, the authors show in the paper that h is bi-Lipschitz. Therefore, spectral normalization is applied on the weights of the block. Moreover, the authors argued that the two-sided gradient penalty is not ideal when working with residual blocks as it can force the block(.) toward zero and thus resulting in an identity mapping.

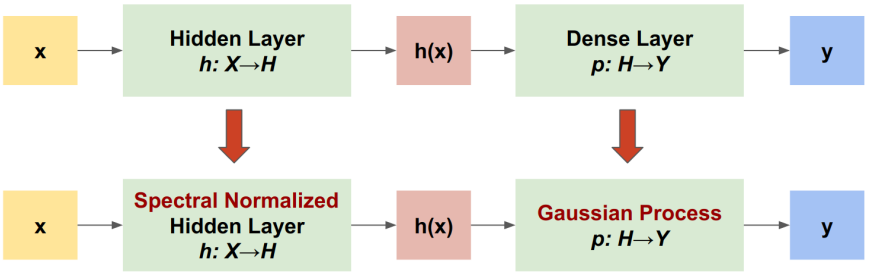


FIGURE 2.10: A comparison between a deterministic model and SNGP. Credits to https://www.tensorflow.org/tutorials/understanding/sngp

Additionally, instead of using an RBF, SNGP relies on a GP. To overcome the intractability of the GP posterior, a Laplace approximation based on the random Fourier feature expansion is used (Rasmussen and Williams, 2006). Importantly, the authors showed that when only the classification

block in the DL model was replaced with a GP, the uncertainty was not as desirable since they are still dependent on the decision boundaries of the model rather than the data distribution. This shows that the model, and more precisely the feature extraction part, should be distance-aware.

2.6.4 Deep Deterministic Uncertainty

Unfortunately, there are some challenges associated with DUQ and SNGP. For instance, they do not allow disentangling the different sources of uncertainties as the model only provide one uncertainty measurement, arguably predictive uncertainty, based on some distance in the latent space. Another complication is the changes in the training pipeline added by these methods, making the convergence quite challenging. For these issues and more, and building on the foundational work of Lee et al. (2018b), Deep Deterministic Uncertainty (DDU) was introduced in Mukhoti and Kirsch (2023).

The idea of DDU is simple: once a regularized model with spectral normalization is trained, the last dense layer is replaced with a Gaussian Discriminant Analysis (GDA) for feature-space density estimation. They showed empirically that a complex method for uncertainty estimation, as used in DUQ and SNGP, is not necessarily and that the conditions of smoothness and sensitivity are sufficient leading to competitive results with fairly simple methods. Importantly, a regularized feature space is necessarily for a reliable uncertainty estimate and for learning the GDA head. The model is thus able to estimate aleatoric uncertainty using the entropy of the softmax probabilities, and epistemic uncertainty by relying on the trained GDA.

2.7 Conclusions

In this chapter, we formalized the notion of uncertainty in machine learning and showed the distinction between the different sources of uncertainties: aleatoric and epistemic. While different types of models allow measuring these uncertainties, such as BDL and EDL, it remains important to investigate whether these models provide a faithful uncertainty estimation. In order to achieve that, many challenges arise on how to make the models uncertainty-aware given one-hot encoded ground truth.

Next, we will explore the calibration aspect of machine learning models Although the reliability of aleatoric uncertainty is perhaps a well studied problem through model calibration, the calibration of epistemic uncertainty is more challenging and has been explored to a lesser extent in the literature. One important component of the epistemic calibration is the choice of the prior distribution. In the next chapter, we will first elaborate on the notion of a large choice for the prior distribution, exploring its significance and impact. Then, we will discuss the model calibration in machine learning, focusing on how it refines the model to better fit the data, and illustrate the challenges of epistemic calibration.

CHAPTER 3
PRIORS AND CALIBRATION

"You know who you are, but know not who you could be."
William Shakespeare

As explored in the formulation of BNNs (Section 2.1.1), the prior distribution is an important building block in the Bayesian framework as it encodes our beliefs and knowledge. Yet, the choice of a prior distribution for Bayesian deep learning model is quite challenging and ambiguous, to say the least. Moreover, the reliability of the predictions in particular, and of the sources of uncertainty in general, holds a significant role for decision-making, and it is studied from the lens of calibration. Still, up until recently, the calibration aspect was mainly analyzed from the lens of aleatoric uncertainty. Recent work, such as (Bengs et al., 2022; Mortier et al., 2023; Jürgens et al., 2024; Fellaji et al., 2024), examined the calibration of epistemic uncertainty emphasizing its importance. In this chapter, we take a closer look at the notion of prior distribution and its implications on the learning process, and how we can approach the analysis of the calibration from an epistemic point of view.

3.1 Priors in Deep Learning

Perhaps the notion of priors could seem not mentioned in the context of standard deep learning models as it is for BNNs, however it is studied under the field of model initialization. Glorot and Bengio (2010) paved the way on showing the importance of the initialization of weights and its dependency on the activation layer. For bounded activation functions, saturation is undesirable as it leads to negligible gradients especially for lower layers, limiting the effect of backpropagation for these layers and hence they demonstrate lower sensitivity to parameter updates. Globally, this phenomenon is known as the *vanishing gradient*, which is also due to a small initialization of the weights regardless of the saturation of the activation function. On the other hand, a large initialization of the weights results in *exploding gradient* (Bengio et al., 1994; Pascanu et al., 2013) where the gradient is substantially large leading to divergence in the training process.

Though the exploding gradient has been examined in previous research such as *gradient norm clipping* (Pascanu et al., 2013) and *Stable ResNet* (Hayou et al., 2021), we concentrate in this section on the effect of the initialization. Under a set of hypotheses for the training data $\mathcal{X}$ (*i.i.d.*, zero-mean and finite variance) and for the parameters of the model (pairwise independence, zero-mean, finite variance, choice of the activation function), an appropriate initialization aims at controlling the

flow in both the forward and backward passes such that going from one layer/block to another, the mean and the variance are preserved in both ways (Glorot and Bengio, 2010; He et al., 2015).

To prevent overfitting when training machine learning models, a regularization term is often added to the loss function. More specifically, L2 regularization (weight decay) and L1 regularization (lasso) are some of the widely adopted regularizations that penalize large weights. L2 regularization discourages large weights by adding the square of the weights to the loss function, while L1 regularization promotes sparsity by adding the absolute values of the weights. One can link the weight regularization to the prior in the Bayesian approach $p(\boldsymbol{\theta})$ by interpreting the former as a Gaussian prior and the latter as a Laplace prior when optimizing the MAP estimate $\hat{\boldsymbol{\theta}}_{\text{MAP}}$.

Remark *(L2 regularization in practice).* From a practical aspect, Loshchilov and Hutter (2019) found that the weight decay in *Adam* optimizer (Kingma and Ba, 2017) (and in adaptive gradient algorithms in general) is different from the formulation of the L2 regularization as detailed previously, mainly because it uses weight decay as part of the gradient update, which can interfere with the adaptive learning rate. They propose *AdamW* which restores the equivalence between the two, by decoupling weight decay from the gradient update, allowing for more effective regularization and better performance.

3.2 Selecting Priors: Vast Choices vs. Practical Limitations

According to the comprehensive survey about priors in BDL conducted by (Fortuin, 2021), a notable observation emerges: contrary to the posterior distribution and the BMA, the prior distribution $p(\theta)$ is often underrepresented, regardless of its importance, in previous work in the field. Even so, most BDL models adopt vague priors, in particular isotropic Gaussian priors (Neal, 1996; Fortuin et al., 2022), without critically examining its fit for the given context. Perhaps this unquestionable choice is due to the high dimensionality of the weights of BDL models: since the interpretation of weights and biases within a neural network is unclear, picking a proper prior that accurately reflects our beliefs becomes fairly difficult (Neal, 1996). Another possibility is to select the prior from a function space perspective, as induced by the choice of a range of elements such as the model family (architecture), the non-linearity functions used, and the prior distribution.

3.2.1 Drawbacks of prior misspecification

In Section 2.1.1, the model-evidence (Equation 2.5) was briefly discussed, especially as a criterion for the model selection in the work of MacKay (1991). In particular, given a specific architecture, the model-evidence $p(\mathcal{D})$ quantifies the likelihood that the data $\mathcal{D}$ is generated by this particular model. As model-evidence contributes to choosing models that generalize well, Wilson and Izmailov (2020) argue that in order to achieve a good generalization, one should take into consideration the support of the model-evidence and the inductive biases of the model, as it is nicely represented in Figure 3.1 on a hypothetical example of images (inspired by MacKay (1991)). Inductive bias refers to the set of constraints that a model incorporates to make inferences about unseen data based on the training data, and it defines the hypothesis space the model searches within. A good model (architecture) is the one that ideally supports an extensive selection of datasets, and has

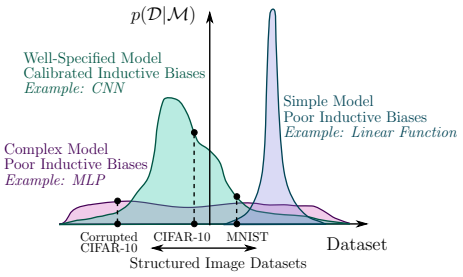


FIGURE 3.1: Representation of the importance of inductive bias and the support. Credits to Wilson and Izmailov (2020).

a high inductive bias (high model-evidence). Therefore, ranking the architectures based on the model-evidence for a specific dataset $\mathcal{D}$ is a manifestation of the no-free-lunch theorem (Wolpert and Macready, 1997): there is no universally favored architecture for all the datasets.

Significantly, the expectation of the model-evidence is a marginalization over $p(\boldsymbol{\theta})$, further illustrating the integral part of the prior distribution. This dependency can be reformulated by the no-free-lunch theorem regarding the prior distribution as stated by Fortuin (2021): there is no one-size-fits-all prior distributional for all tasks. Specifically, ranking models under prior misspecification could affect the choice of the model (Fortuin, 2021). Furthermore, given that model-evidence can be expressed as the product of the best-fit likelihood and the Occam factor (*i.e.* the width of the posterior over the width of the prior), prior misspecification could lead to larger Occam factor, and thus a low model-evidence, leading to an erroneous rank of models (MacKay, 1991, 1992a; Lotfi et al., 2022).

3.2.2 Isotropic Gaussian priors

Given an MLP with a single hidden layer and a non-linear activation function, the *universal approximation theorem* (Cybenko, 1989; Hornik et al., 1989) states that this model can approximate any continuous function on a compact subset of $\mathbb{R}^n$. Extending this fundamental theorem in machine learning, Neal (1996) studied the effect of the prior distribution on the prior over functions, in the infinite width limit. When the prior distribution over the weights are sampled from an isotropic centered Gaussian distribution with a fixed variance (inversely proportional to width of the hidden layer), the BNN convergences to a GP. Remarkably, Neal (1996) also showed that, with a Gaussian prior, each unit in the hidden layer has a negligible contribution to the predictions, and hence these small contributions do not form "hidden features" (*i.e.* a single unit has negligible contribution to the outputs) as they do not capture, individually, relevant information about inputs.

3.2.3 Beyond isotropic Gaussian priors

While standard practice often relies on the isotropic Gaussian priors, there exist alternative distributions that possess more favorable theoretical characteristics for certain tasks. In particular, heavy-tailed distributions are analyzed for their practical properties.

Priors with infinite variance. In addition to studying the effect of the Gaussian prior, Neal (1996) also analyzed non-Gaussian priors, especially non-Gaussian *symmetric stable distributions* (Definition 3.1) for $\gamma < 2$, for which the variance is infinite. For such prior distributions, the hidden layer (for an MLP with a single hidden layer) results in "hidden features" (*i.e.* the contribution of some units is significant on the outputs), contrary to the case of a Gaussian prior (Section 3.2.2). Additionally, as the stability parameter γ decreases, the contribution of a small subset of units in the learned features increases.

Definition 3.1: Stable distribution (Feller, 1966)

Let $\{Z_i\}_{i\in\{1,\dots,n\}}$ be n mutually independent random variables from the same distribution $\mathcal{R}$. $\mathcal{R}$ is said to be stable, characterized by the stability parameter $\gamma \in]0, 2]$ if: $n^{-1/\gamma}\sum_{i=1}^{n} Z_i \sim \mathcal{R}$. The centered Gaussian and Cauchy distributions are special cases of Stable distributions ($\gamma = 2$ and $\gamma = 1$, respectively).

Priors with finite variance. In the context of BNN, various prior distributions exist that may enhance performance when compared to an isotropic Gaussian prior. Fortuin (2021) discusses a few that either are heavy-tailed with a fixed variance, such as the Laplace distribution, or allow

correlation between the weights, such as the matrix Gaussian distribution. In a more recent work, Fortuin et al. (2022) analyze the distribution of the weight of SGD-trained models to better set the prior distribution: an "a posteriori" analysis for the prior distribution. They show empirically that uncorrelated heavy-tailed prior distributions are suitable for MLPs and CNNs, whereas correlated Gaussians are more adequate for ResNets. This is perhaps in line with the work of Berger et al. (2009), where the prior is studied from the perspective of expected information: as the prior becomes sharper, the expected information from the data decreases. More on that in Section 3.2.7.

Additionally, the analysis of the weights of the SGD-trained ResNets brought back into focus an old result by Neal (1996): in the case of deep models, "a combination of Gaussian and non-Gaussian priors appears most interesting". In the lower layers, the weight distribution is heavy-tailed, while in the upper layers, the distribution tends to resemble a Gaussian distribution.

In the case of Bayes by Backprop (BBB) Blundell et al. (2015), also discussed in Section 2.1.3, the authors exploit indirectly the conversation about the benefits of heavy-tailed distributions through the use of a mixture of two Gaussians as the prior, with one having a large variance and the second having a negligible variance. Two main objectives motivate this choice in their case. First, the Gaussian with the smallest variance will encourage sparsity in the model. Secondly, employing a shared fixed-form prior will facilitate the rapid use of the prior throughout the learning process, as it does not require adjusting its hyperparameters during training. Interestingly, modifying the prior during training has been shown to provide no benefits, often leading to worse performance.

3.2.4 The effect of the activation functions

Without any doubt, the choice of the model architecture, including the non-linear activation function, is translated in the prior over functions that could be learned. As previously highlighted, an MLP with a single layer is equivalent to Gaussian process in the infinite width limit. The work of Cho and Saul (2009) generalizes the link between GPs and deeper MLPs for a specific family of activation functions $\{x \mapsto \max(0,\, x^n) \mid n \in \mathbb{N}\}$.

For instance, it is shown that with $n = 1$, which is the ReLU activation function, we obtain a GP such that the kernel function is the arc-cosine kernel. Years after, Lee et al. (2018a) introduced *NNGP* (neural network Gaussian process), a framework that connects deep neural networks with GPs by drawing perfect equivalence in the limit of infinity between wide-deep neural networks and GPs. This equivalence is demonstrated by induction from the results on the single hidden layer:

$$\begin{cases} K_i^0(x_1,\, x_2) & = \mathbb{E}\left[z_i^{(1)}(x_1)\, z_i^{(1)}(x_2)\right] \\ & = \sigma_b^2 + \sigma_w^2 \frac{x_1 \cdot x_2^\intercal}{d_{in}} \\ K_i^l(x_1,\, x_2) & = \mathbb{E}\left[z_i^{(l+1)}(x_1)\, z_i^{(l+1)}(x_2)\right] \\ & = \sigma_b^2 + \sigma_w^2 \mathrm{F}_\phi\left(K_i^{l-1}(x_1,\, x_1),\, K_i^{l-1}(x_1,\, x_2),\, K_i^{l-1}(x_2,\, x_2)\right) \end{cases} \tag{3.1}$$

With, for the layer l, $K^l = diag(K_i^l)$ is the diagonal covariance matrix, $z^{(l)}(x)$ is the pre-activation for the input x, $W^l \sim \mathcal{N}(0,\, \sigma_w^2 / d_{h^l})$ is the weights matrix with a variance inversely proportional to the width of the layer l and $b^l \sim \mathcal{N}(0,\, \sigma_b^2)$ is the bias vector. What is shown in (Lee et al., 2018a) is that K^l depends recursively on K^{l-1} thanks to the function F_ϕ whose formulation is governed entirely by the non-linearity ϕ (refer to the paper for more details.). Equation 3.1 can be solved analytically only for a specific set of non-linear activation functions, otherwise numerical approximations are needed.

The resemblances of NNGP and their equivalent GP was studied in (Lee et al., 2018a), for different values for the width. Without any surprise, the larger the width, the closer the performances of the neural network and the NNGP, and the more the correlated the NNGP uncertainty and prediction error of neural network. Additionally, D'Angelo and Henning (2022) further illustrated visually, for different activation functions, that a BNN with a moderate and finite width and the corresponding NNGP result in closely similar uncertainty estimation of the posterior distribution.

3.2.5 Cold posterior effect

In BNNs, it is sometimes common to artificially temper the posterior distribution by considering $\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})^{1/T}$ instead of $\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$, for some positive scalar T. Although it differs from the traditional Bayesian thinking, (Wenzel et al., 2020) show that by cooling the posterior distribution ($T < 1$), the prediction performances are improved. This phenomenon is referred to as the *cold posterior effect*. Ideally, the Bayesian paradigm should give the best performance, however, compared to an SGD baseline and the Bayesian posterior ($T = 1$), tempering the posterior with $T \ll 1$ result in better accuracies (Wenzel et al., 2020; Adlam et al., 2020; Aitchison, 2021; Fortuin et al., 2022). In the following, we will link the cold posterior effect to the choice of the prior and give some explanations from the literature.

Let's assume a random variable Z is a mixture of three Gaussians:

$$Z \sim 0.8 \times \mathcal{N}(0,\, 0.5) + 0.1 \times \mathcal{N}(2,\, 0.2) + 0.1 \times \mathcal{N}(-2,\, 0.2)$$

and we would like to visualize the effect of the temperature on the random variable $Z^{1/T}$ (Figure 3.2). With a low temperature, the distribution of $Z^{1/T}$ is sharp, and concentrates around the central dominant Gaussian. The effect of the two other Gaussians is reduced, leading to a distribution with a single peak. However, in the case of a high temperature, all the elements in the mixture contribute to the overall distribution, hence the middle Gaussian becomes less dominant and the distribution is smoother. Therefore, the temperature parameter controls the balance between concentration and spread, with lower values emphasizing centrality and higher values promoting a more uniform and smooth mixture.

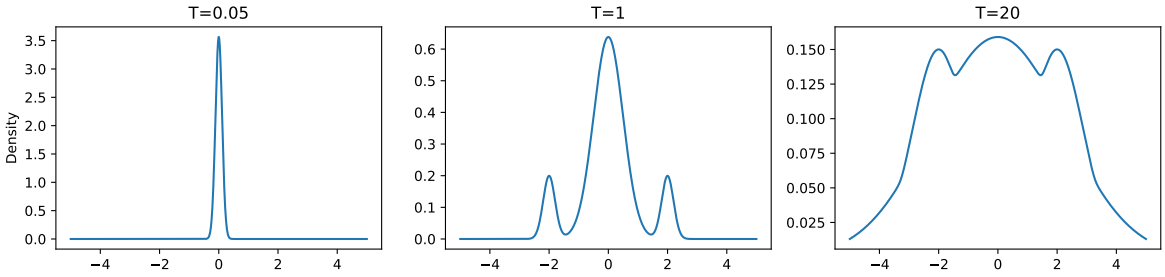


FIGURE 3.2: Effect of tempering a distribution with cold and warm temperatures.

Choice of the prior. Wenzel et al. (2020) argue that the cold posterior effect is partially due to the selection of the prior distribution. For example, the simple and popular choice of isotropic Gaussian priors are shown to be inadequate as they could assign significant prior mass to undesirable functions. Fortuin et al. (2022) validated the same hypothesis and further show that a better choice of the prior, namely heavy-tailed distributions, could lessen the cold posterior effect. However, increasing the performances of the Bayesian posterior through the choice of the prior is not task-agnostic: whereas the cold posterior effect disappears for fully-connected and CNNs in MNIST and FashionMNIST, it is not the case on CIFAR10 with a ResNet model. Furthermore, Wilson and Izmailov (2020) criticize the findings of Wenzel et al. (2020) regarding the isotropic Gaussian

prior and consider them an artifact of a poor choice of the variance. Additionally, Adlam et al. (2020) illustrate the effect of the prior distribution on aleatoric uncertainty, which is reduced when tempering the posterior. We will discuss shortly the link between temperature scaling and model calibration (Section 3.3).

Model misspecification. In the context of BNNs, every model involves some level of misspecification, especially parametric neural networks (Wilson and Izmailov, 2020). This phenomenon arises when the true data-generating process falls outside the hypothesis functional space of the model, resulting in incorrect model assumptions. As a result, even with an infinite amount of data, the best possible learner within the hypothesis space will still fail to accurately capture the underlying data-generating process. This misspecification is part of the epistemic uncertainty (Hüllermeier and Waegeman, 2021), and it could be explained by the bias induced by SGD (Kale et al., 2021; Lahlou et al., 2023). As a result, tempering the posterior mitigates this lack of knowledge about the "ideal" model.

Remark *(Cold posterior vs tempered posterior).* There exists a difference between the cold posterior and tempered posterior (Zhang et al., 2018a; Wilson and Izmailov, 2020) for which only the likelihood is tempered:

$$p_T(\boldsymbol{\theta} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta})^{1/T} p(\boldsymbol{\theta})$$

It can be interpreted as regularizing the KL divergence in Equation 2.9. In some cases, they could be equivalent (Aitchison, 2021), with adjusted prior variances for example.

3.2.6 Use case: BBB with different priors

To illustrate the difficulty of choosing a prior, we will experiment with a simple MLP model with two hidden layers under the BBB formulation (Blundell et al., 2015) and trained with the loss function in Equation 2.13 on the MNIST dataset. To this end, a range of priors is tested, Normal, Laplace and Cauchy, centered around 0, with both a reasonably low scale (0.2) and a vague scale (2). In addition, we compare also to a uniform prior and a mixture of Gaussians as reported in (Blundell et al., 2015). Figure 3.3 shows the five priors in the low scale case. We emphasize that only a balanced subset of the entire training set of MNIST is used during training[6] in order to ensure that the prior retains a meaningful influence on the posterior distribution. Finally, much like Blundell et al. (2015), we set the variational posterior distribution to a Gaussian distribution. We refer to Appendix A for implementation details and additional results.

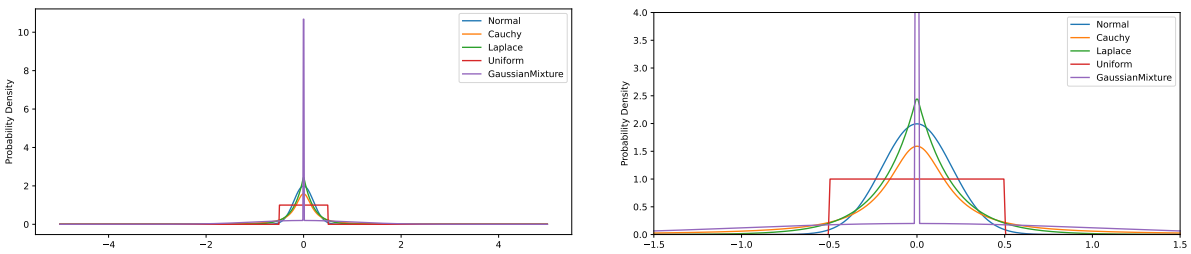


FIGURE 3.3: Visualization of the prior distributions used.
On the right, a zoomed-in view on the x and y axes.

[6]The same logic of splitting the dataset will be applied consistently across other experiments as well: the entire dataset is first divided into validation and train sets. Then, the train set is split further to form a subset that will be used for the experiments when a smaller train set is required.

Under the assumption of a Gaussian variational posterior $\mathcal{N}(\mu, \sigma)$, all the models start from the same initialized model with the same training protocol expect the prior distribution. The parameters are assumed to be uncorrelated, so the covariance matrix is diagonal. For implementation convenience, we can take advantage of the reparameterization $\sigma = \log(1 + \exp(\rho))$ and optimize for $\rho \in \mathbb{R}$, where in this case we are learning a real value ρ rather than a positive value σ. At initialization, μ is sampled from a centered uniform around zero, to ensure convergence of the model while ρ is also sampled from uniform distribution allowing small values for σ. Figure 3.4 illustrates the MLP model at initialization[7].

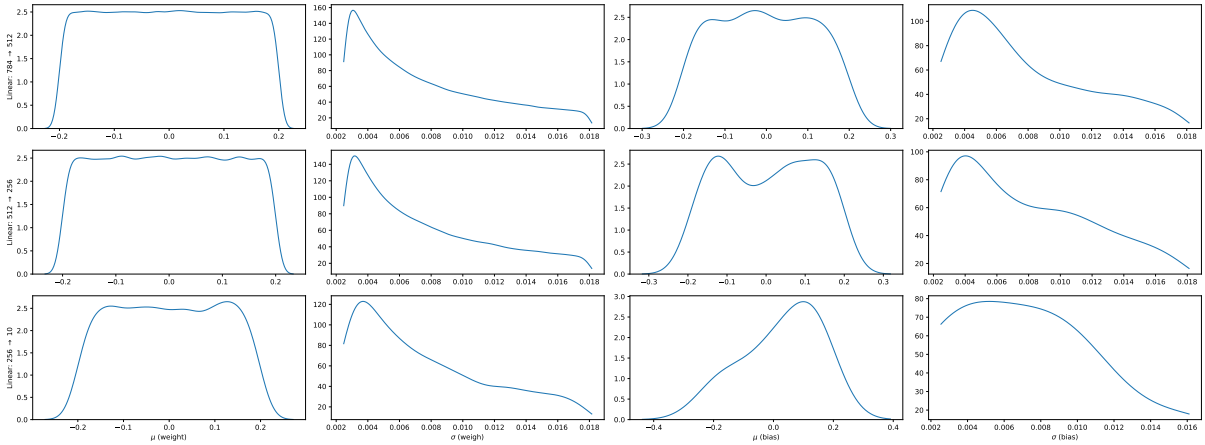


FIGURE 3.4: Visualization of σ and μ for the parameters of the MLP at initialization: rows are for the linear layers, columns are for the weights and biases.

The results of this experiment can be seen in Figure 3.5 for a training of 150 epochs. In the case of narrow priors, stable distributions perform the best compared to Laplace prior with the latter being the worst from an accuracy perspective. Mixture of Gaussians has comparable results to a narrow Cauchy distribution (again in terms of accuracy), which is in line with the discussion at the end of Section 3.2.3, as it is a heavy-tailed distribution. For vague priors, Laplace prior outperforms stable distributions and scores the highest accuracy overall. Additionally, we can see the results of the uniform prior are comparable to those of vague priors. Finally, and unsurprisingly, the ranking of the priors in the validation set is consistent for the test set as well (Table 3.1).

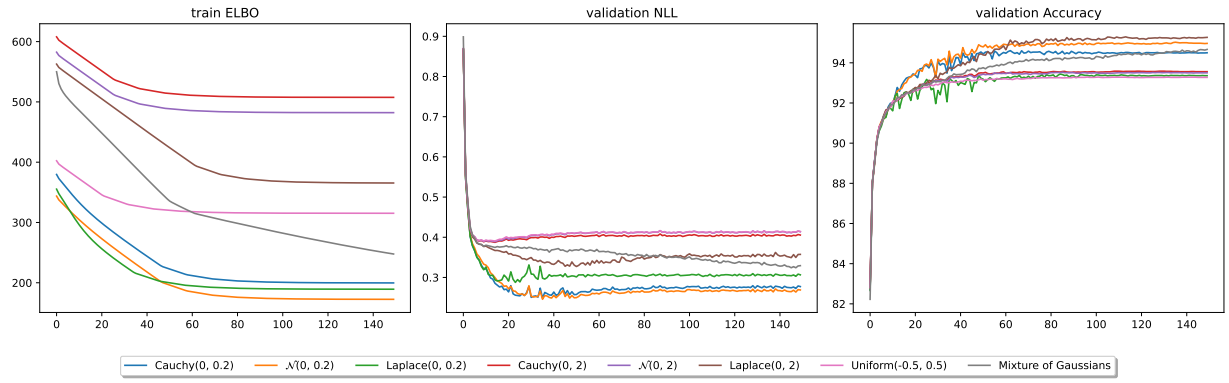
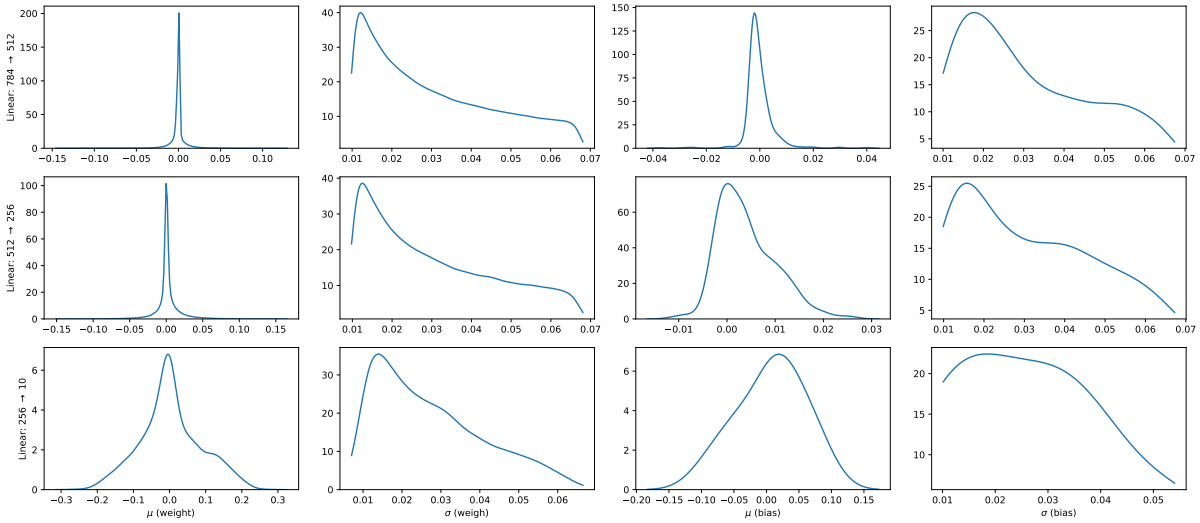
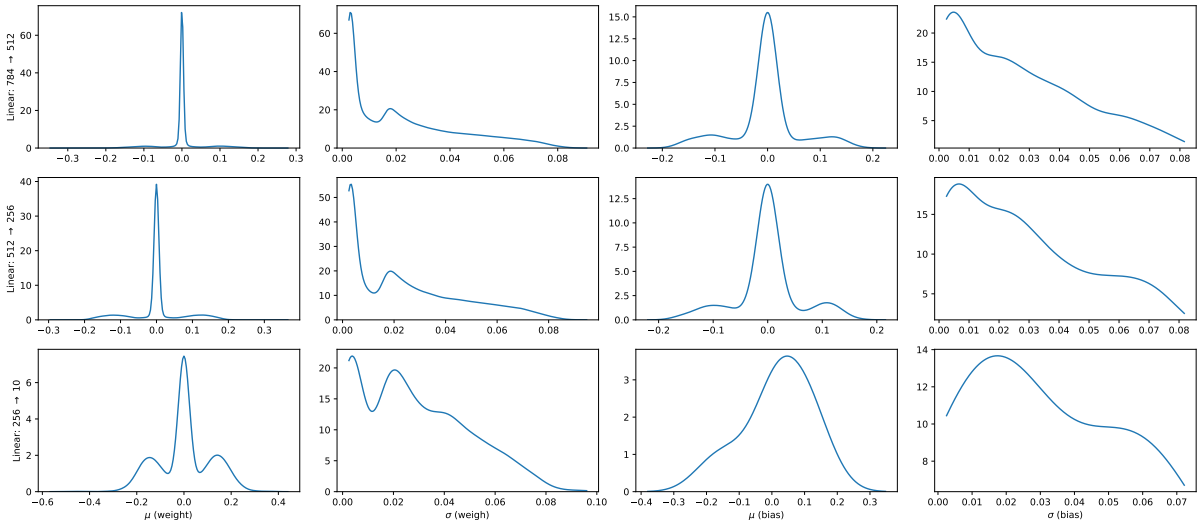


FIGURE 3.5: Learning curves on the train and the validation sets as functions of the epoch: (negative) ELBO on the train set on the left, NLL on the validation set in the middle and the accuracy on the validation set on the right.

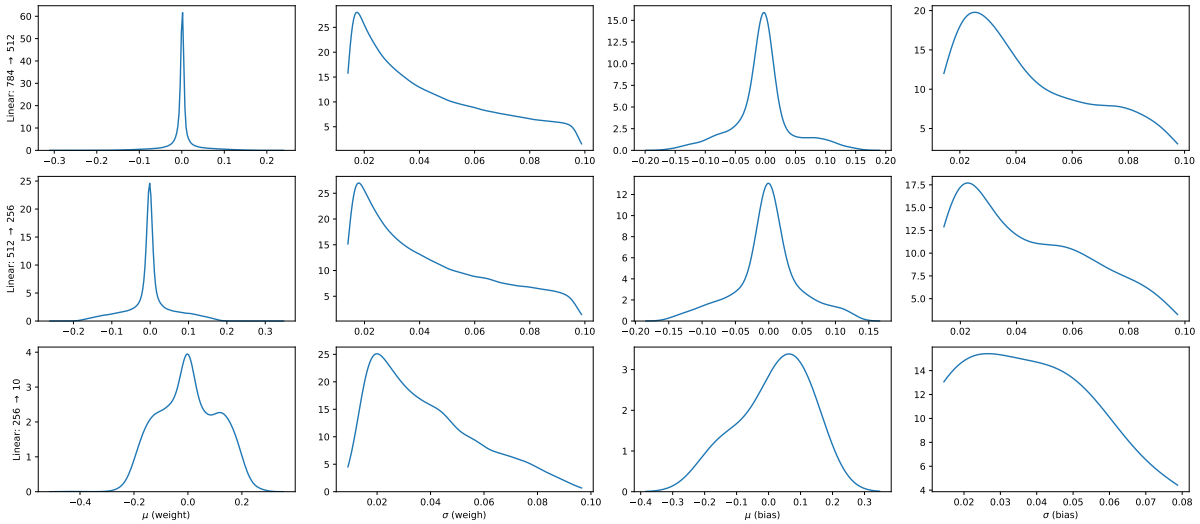
[7] In Figure 3.4, since the bias of the last layer is only a vector of dimension 10, the "uniformity" is not clear on the histogram.



(A) Cauchy(0, 0.2)



(B) Mixture of Gaussians



(C) Laplace(0, 2)

FIGURE 3.6: Visualization of σ and μ for the parameters of the trained MLPs.

Priors	NLL	Single (acc.)	BMA (acc.)	$\mathcal{U}_{\mathcal{D}}^{t}$	$\mathcal{U}_{\mathcal{D}}^{a}$	$\mathcal{U}_{\mathcal{D}}^{e}$
Cauchy$(0, 0.2)$	2.46e-1	95.29%	95.05%	2.87e-1	2.81e-1	6.22e-3
$\mathcal{N}(0, 0.2)$	2.35e-1	95.66%	95.36%	2.19e-1	2.13e-1	5.85e-3
Laplace$(0, 0.2)$	2.79e-1	93.83%	93.81%	4.46e-1	4.41e-1	5.27e-3
Mixture of Gaussians	2.85e-1	95.33%	95.31%	1.32e-1	1.25e-1	6.38e-3
Cauchy$(0, 2)$	3.66e-1	93.73%	93.78%	1.11e-1	1.07e-1	4.58e-3
$\mathcal{N}(0, 2)$	3.74e-1	93.64%	93.77%	1.10e-1	1.05e-1	4.60e-3
Laplace$(0, 2)$	3.10e-1	95.80%	95.59%	1.32e-1	1.23e-1	8.56e-3
Uniform$(-0.5, 0.5)$	3.77e-1	93.41%	93.51%	1.11e-1	1.07e-1	3.95e-3

TABLE 3.1: Evaluations on the test set. Single (acc.) is the accuracy for a single model where we consider the mean of the distributions as the weights of model (equivalent a standard DL model). BMA (acc.) is the accuracy computed based on the average of the sampled models. The number of forward passes T is set to 50 for the BMA (Equation 2.23) and uncertainties (Equations 2.24, 2.25 and 2.26). We report the average values of uncertainties on the test set.

When looking at the parameters of the variational Gaussian posterior distribution at the end of the training, one can notice that μ are closer to zero for the Cauchy prior (Figure 3.6a) compared to the mixture of Gaussians (Figure 3.6b) and the Laplace (Figure 3.6c) priors. Perhaps this could explain the similarities of the uncertainty measure between the latter two priors (Table 3.1). However, the variance of the posterior distribution is comparable in the case of Cauchy and Laplace priors whereas its shape is different in the case of the mixture of Gaussians. Similar plots for the parameters of the posterior distribution could be found in Appendix A.

This simple example illustrates the difficulty of defining priors, especially in the weights space. To reduce the subjectivity involved in selecting a prior distribution, a challenge faced by both practitioners and experts, we will next examine the concept of uninformative priors and present several commonly used examples.

3.2.7 Objective priors

As for now, we looked at priors as means of encoding our prior beliefs into the Bayesian framework, even if in practice this is not an easy task, not to say impossible. However, these beliefs are undoubtably subjective as different practitioners could set various hypotheses, let alone the challenges of defining a subjective prior. To reduce this subjectivity while preserving the regularization effect of the prior, it is common to use *objective priors* that aim at reducing the effect of the prior during inference by choosing the prior based on the assumed model rather than subjective knowledge. Different terminologies exist for this family of priors, for example, uninformative, non-subjective or default priors (Berger, 2006). In the following, we will discuss three types of (sometimes questionably) objective priors: uniform priors, Jeffreys prior and reference priors.

Uniform priors. Perhaps the naive uninformative prior to think of is the uniform prior:

$$p_{\text{uni}}(\boldsymbol{\theta}) \propto 1$$

This prior has persisted for centuries, with origins dating to the work of Bayes (1763) and de Laplace (1820). Arguably, it should include most of the mass of the likelihood function (Berger, 2006). Even though it results in vague prior, the prior in this case lacks any subjectivity. This particular prior cancels the contribution of the prior distribution in the computation of the posterior

distribution (Equation 2.4) making it depend solely on the likelihood. However, the uniform prior has bad calibration properties (Berger, 2006) and is criticized as it could lead to improper priors (does not integrate to 1) and thus "unBayesian property" (Dawid et al., 1973). Bernardo (2005) discouraged the use of flat priors as they require assumptions that are rarely verified: "the uncritical (ab)use of such flat priors should be strongly discouraged". Although it is possible to (non-linearly) transform the parameters with an injective function to a bounded interval to mitigate the improper prior (Berger, 2006), subjectivity is added through the choice of this particular mapping.

Jeffreys prior. To avoid the dependence of the prior distribution on the reparameterization, Jeffreys (1946) proposed *Jeffreys prior*, formally defined in Definition 3.2. Its definition is fundamentally motivated by the invariance under continuously differentiable transformation of the FIM.

Definition 3.2: Jeffreys prior (Jeffreys, 1946)

For a parameter-space parameterized with $\boldsymbol{\theta}$, the Jeffreys prior p_{Jeff} is defined as:

$$p_{\text{Jeff}}(\boldsymbol{\theta}) \propto \det(I(\boldsymbol{\theta}))^{1/2} \tag{3.2}$$

With $I(\theta)$ being the Fisher information matrix (FIM) given by the derivatives of the log-likelihood:

$$I(\boldsymbol{\theta})_{i,j} = \mathbb{E}_{\boldsymbol{\theta}}\left[\left(\frac{\partial \log(p(Z \mid \boldsymbol{\theta}))}{\partial \boldsymbol{\theta}_i}\right)\left(\frac{\partial \log(p(Z \mid \boldsymbol{\theta}))}{\partial \boldsymbol{\theta}_j}\right)\right] \tag{3.3}$$

Berger (2006) underlines the popularity of both the Jeffreys and uniform priors in practice compared to the other existing objective priors. Berger et al. (2015) considered the Jeffreys prior to be the optimal objective choice for regular one-parameter models from various perspectives. Nonetheless, it presents challenges when applied to multi-parameter models. Additionally, Jeffreys prior could yield improper distributions[8] and sometimes suffers from the *marginalization paradox* (Dawid et al., 1973), leading to a situation where the marginal posterior distribution cannot be recovered from the joint distribution[9]. Surprisingly, in the multivariate case, this prior was also criticized by Jeffreys himself (Bernardo, 2005).

Remark *(FIM and Hessian).* Under some regularity conditions, FIM can be computed as the expectation of the Hessian of the log-likelihood:

$$I(\boldsymbol{\theta})_{i,j} = -\mathbb{E}_{\boldsymbol{\theta}}\left[\frac{\partial^2 \log(p(Z \mid \boldsymbol{\theta}))}{\partial \boldsymbol{\theta}_i \partial \boldsymbol{\theta}_j}\right] \tag{3.4}$$

On a related note, the FIM (Equation 3.4) is also used to determine the number of effective parameters in a model, which is the number of parameters that actively contribute to the model's predictive capability (MacKay, 1991; Maddox et al., 2020; Berezniuk et al., 2020; Abbas et al., 2021). This could be linked to the *lottery ticket hypothesis* (Frankle and Carbin, 2019) by finding the *winning tickets* in the parameter space, in a data-oriented manner. Moreover, being a measure of the amount of information that the data provides about the unknown parameter, the FIM is used in the Laplace approximation to fit a Gaussian distribution to the posterior probability: the mean of this Gaussian is the MAP estimate, and its covariance matrix is given by the inverse of the FIM (MacKay, 1992b; Louizos and Welling, 2017; Ritter et al., 2018; Karakida et al., 2019; Daxberger et al., 2022), which aligns with the *Bernstein-von Mises* theorem.

[8] Improper in the sense that its integral (or sum) diverges.
[9] The paradox is further illustrated through examples in Bernardo (2005); Berger (2006).

Reference priors. An enhancement of Jeffreys priors for higher dimensional problems is proposed by Bernardo (1979) and referred to as *reference priors*, formally defined in Definition 3.3. The reference priors are formalized to further make the contribution of the data dominant while reducing the effect of the prior.

Definition 3.3: Reference priors (Bernardo, 1979)

A prior distribution is said to be a reference prior if it maximizes the mutual information between $\boldsymbol{\Theta}$ and the observations $\boldsymbol{Z}$:

$$\mathrm{p}_{\mathrm{ref}}(\boldsymbol{\theta}) \in \underset{\mathrm{p}(\boldsymbol{\theta})}{\operatorname{argmax}} \left(\mathbb{I}(\boldsymbol{\Theta}; \boldsymbol{Z})\right) \tag{3.5}$$

When dealing with a single parameter, Bernardo (1979) showed that Reference priors converges asymptotically to the Jeffreys prior, making them equivalent. In the multivariate case, the equivalence does not hold in general, and Clarke and Barron (1994) demonstrated that Jeffreys prior is the continuous prior that asymptotically maximizes the mutual information.

Equation 3.5 is equivalent to finding the prior that maximizes, on average, the Kullback-Leibler divergence of the posterior and the prior distributions:

$$\mathbb{I}(\boldsymbol{\Theta}; \boldsymbol{Z}) = \mathbb{E}_{\boldsymbol{Z}}[\mathrm{D}_{\mathrm{KL}}(\mathrm{p}(\boldsymbol{\theta} \mid \boldsymbol{Z}) \,\|\, \mathrm{p}(\boldsymbol{\theta}))] \tag{3.6}$$

In the continuous case, one could restrict the set of admissible priors to a compact set to ensure a finite maximum of the mutual information. Additionally, due to the invariance to reparameterization of mutual information, the reference prior is consequently invariant to a reparameterization of the weight space.

Yet, in addition to the difficulty of having access to the distribution of the generating process $\mathrm{p}(\boldsymbol{Z})$, especially in practice, finding the reference priors is dependent on the posterior distribution which is itself computed based on the prior distribution. By doing so, the prior distribution does not fully describe our prior belief, it is however designed to reduce the effect of the prior distribution, objectively, once the learning occurs. Even though it is possible to select the reference prior based on a subset of samples, the size of this subset matters.

Although the idea of finding the reference priors as in Definition 3.3 seems appealing, simpler approaches exist such as the *reference prior approach*, that relies on the notion of k-reference priors, and *reference posteriors*. The former consists of searching the reference priors for k observations from $\mathrm{p}(\boldsymbol{Z})$: $Z^{(k)} = (Z_1, \ldots, Z_k)$, the k-reference priors (Berger et al., 1988) are formally defined as:

$$\mathrm{p}^{(k)}_{\mathrm{ref}}(\boldsymbol{\theta}) \in \underset{\mathrm{p}(\boldsymbol{\theta})}{\operatorname{argmax}} \left(\mathbb{I}\left(\boldsymbol{\Theta}; Z^{(k)}\right)\right) \tag{3.7}$$

Thus, the asymptotic reference priors are then defined in the limit of $k \to +\infty$:

$$\lim_{k \to +\infty} \mathrm{p}^{(k)}_{\mathrm{ref}}(\boldsymbol{\theta}) = \mathrm{p}^{*}_{\mathrm{ref}}(\boldsymbol{\theta})$$

For additional details on the subject, please refer to the following relevant papers: Bernardo (1979, 2005); Berger (2006); Berger et al. (1988, 2009, 2015); Gao et al. (2022). Finally, the link between the choice of the prior distribution and the cold posterior effect was discussed in Section 3.2.5. As far as we are aware, the cold posterior effect has not been addressed in the literature from the lens of the reference priors, which could be beneficial since it aims at making the posterior mostly influenced by the data rather than the choice of the prior.

3.3 Model Calibration

Dealing with probabilistic models requires having a reliable probability estimate, especially for deep learning models which are shown to yield overconfident predictions (Szegedy et al., 2015; Sensoy et al., 2018), with this overconfidence being a common indication of overfitting (Szegedy et al., 2015; Müller et al., 2020). Moreover, as shown in Guo et al. (2017), such a high level of confidence leads to non-calibrated models, further motivating a focus on the calibration aspect of recent deep learning models. The reliability of the probabilities is defined by how they reflect reality, which is defined through the true labels (embedded in $\mathcal{D}$). In this section, we will discuss the calibration of models from an aleatoric and epistemic perspectives.

For instance, let's imagine two standard models $\mathcal{M}_1$ and $\mathcal{M}_2$, trained for a binary classification task. Assuming that their softmax-probabilities to the same input x are, for example, $p^{(1)} = [0.05, 0.95]$ and $p^{(2)} = [0.45, 0.55]$. In both models the predicted class is the same, yet the probability vectors are profoundly different. To some extent, by studying the calibration of the model, one can determine which model is better calibrated. Contrarily, using the least calibrated model could lead to dramatic outcomes in safety-critical applications for example. Unless stated otherwise, we will illustrate the calibration examples in this chapter for a binary classification setup, thus the model output will refer to the probability of the positive class, for example: $p^{(1)} = 0.95$ and $p^{(2)} = 0.55$.

3.3.1 Aleatoric Calibration

The concept of model calibration could be traced back to the 20$^{\text{th}}$ century with the work of Murphy and Winkler (1977); Dawid (1982); DeGroot and Fienberg (1983) to name a few, advocating for the importance of reliable probabilities especially in applications such as weather forecasting. Recently, Guo et al. (2017) sheds light on the calibration of modern deep learning models. In both cases, the focus is on single models and hence the calibration is evaluated from an aleatoric uncertainty perspective. Notably, the use of single probabilistic models (through the softmax layer) confounds the aleatoric and total/predictive components, making the mention of the aleatoric calibration implicit in the posterior distribution. To avoid any confusion, we use the term *model calibration* to refer to the calibration of the aleatoric uncertainty. For more details, we refer to the recent survey (Wang, 2024) providing an in-depth examination and thorough exploration of the calibration and an exhaustive list of references.

Definition 3.4: Calibrated model

A classifier is said to be (confidence) calibrated if, for a given input-output (x, Y), such that $(\hat{Y}, \hat{p})$ are the predicted class of the input x and its probability, respectively:

$$\forall s \in [0,1]; \quad \mathrm{P}(Y = \hat{Y} \mid \hat{p} = s) = s$$

The calibration is formally defined in Definition 3.4. Intuitively, we would like for the model's softmax probability to be realistic: the probability of the predicted class is close to the actual probability. While studying the realism of predicted probabilities as stated in Definition 3.4 can be challenging in practice, a discrete analysis is often preferred by splitting the interval $[0, 1]$ into B bins $\{I_b\}_b$ based on the predicted probabilities. In each bin I_b, we have n_b samples and two meaningful values are computed: the *accuracy* of the bin $\mathrm{acc}(I_b)$ which is the percentage of the correctly classified samples, and the *confidence* within the bin $\mathrm{conf}(I_b)$ which is the average of the predicted probabilities (Equation 3.8).

$$\mathrm{acc}(I_b) = \frac{1}{n_b} \sum_{i \in I_b} \delta(\hat{Y}_i = Y_i) \qquad ; \qquad \mathrm{conf}(I_b) = \frac{1}{n_b} \sum_{i \in I_b} \hat{p}i \tag{3.8}$$

Three regimes can be distinguished: calibrated, under-calibrated, and over-calibrated. A model is calibrated if its predictions match the ground truth. However, when the confidence is below (respectively higher) the accuracy, the model is under-calibrated (respectively over-calibrated).

$$\begin{aligned} \textbf{Calibrated:} \quad & \text{conf}(I_b) = \text{acc}(I_b) \\ \textbf{Under-calibrated:} \quad & \text{conf}(I_b) < \text{acc}(I_b) \\ \textbf{Over-calibrated:} \quad & \text{conf}(I_b) > \text{acc}(I_b) \end{aligned}$$

The *reliability curve/diagram* (Murphy and Winkler, 1977) is often used in the binary classification setting to visualize the calibration of a binary classifier by having the confidences on the x-axis[10] and the accuracies on the y-axis. The three regimes can be easily illustrated in the diagram with the calibrated state being on the diagonal. Although this curve is not limited under Definition 3.4 to binary classification, its interpretation remains straightforward in that case.

The aleatoric dependency in the definition of calibration can be easily illustrated. For instance, we assume that the data (inputs and outputs) are sampled from a noiseless generating process. In this regard, measuring the calibration of aleatoric uncertainty or re-calibrating an existing model can be based on the reliably available data (*i.e.* ground truth labels), as it will be discussed shortly. However, having imprecise labels or noisy inputs are sources of aleatoric uncertainty, and it will affect the model calibration. Therefore, model calibration is closely linked to aleatoric uncertainty as it adjusts model predictions to better reflect the inherent noise and variability in the data.

Remark *(Different levels of calibration).* As detailed in Kull et al. (2019); Vaicenavicius et al. (2019); Mortier et al. (2023), more strict definitions of calibration exist in the literature such as the *classwise calibration* (Zadrozny and Elkan, 2001), and the *calibration in the strong sense* (Widmann et al., 2019). We refer to Silva Filho et al. (2023); Mortier et al. (2023) for a detailed discussion.

Calibration Scores:

There exist different scores to quantitatively measure the calibration of a model, most of which are *proper scoring rules* (Gneiting and Raftery, 2007) and rely on the bins accuracy and confidence (Equation 3.8). Arguably, the *expected calibration error* (ECE) (Naeini et al., 2015) is the most common calibration measure, which is defined as:

$$\text{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} |\text{acc}(I_b) - \text{conf}(I_b)|$$

Nixon et al. (2020) criticize the use of ECE to measure calibration in the multiclassification tasks on three main reasons. First and foremost, ECE only relies on the predicted class and its softmax probability while ignoring the softmax probabilities of the $C-1$ classes. Second, uniform binning, as used in (Guo et al., 2017), could be problematic as in this case some bins do not contribute to the calibration measure. Finally, the discretization of interval $[0, 1]$, through the number of bins, could lead to approximation errors As a result, new calibration measures were proposed by Nixon et al. (2020), such as *Adaptive Calibration Error* (ACE) and *Static Calibration Error* (SCE). Contrarily to ECE, both ACE and SCE consider all the classes and not just the one with the highest probability. Hence, SCE is the direct generalization of ECE to multiclassification. In addition, ACE uses as well an adaptive calibration ranges making the bins contain equal number of predictions.

[10] For a given bin I_b, since all the predicted probabilities are bounded within I_b, $\text{conf}(I_b)$ is consequently bounded within I_b. Thus, the values of the vector $[\text{conf}(I_b)]_b$ are in ascending order.

In addition, because Brier score (Brier, 1950) is a proper scoring rule, it was shown in (DeGroot and Fienberg, 1983) that it can be decomposed into a calibration component and a refinement part. The latter being related to the area under the ROC curve, the Brier score does not solely and reliably measure model calibration.

Calibration Techniques:

Due to the overconfidence artifacts resulting from the training process of standard deep learning models, a calibrated model is obtained either from post-hoc calibration techniques or from regularizing the loss function (typically the cross-entropy loss).

Post-hoc calibration. There exist a wide range of approaches to make models better calibrated once trained. For instance, *Platt scaling* (Platt, 1999) fits a linear mapping from the models outputs to the calibrated logits, which will be used to get softmax probabilities. A special case of Platt scaling that is widely used in deep learning is *temperature scaling* (Guo et al., 2017) which consists of tempering the logits with a scalar. These methods could lead to a calibrated model in the "weak" sense as argued by Kull et al. (2019). They proposed a parametric general-purpose multiclass calibration technique *Dirichlet calibration* which is a generalization *Beta calibration* (Kull et al., 2017). Ovadia et al. (2019) studied the effect of temperature scaling on deep learning models and show that it makes the model more calibrated. (Kull et al., 2019) further compared Dirichlet calibration to temperature scaling and find that the former outperformed the latter across different configurations.

Entropy regularizers. While directly regularizing the weights of a model does not guarantee avoiding overconfident outputs, it is possible to achieve that by regularizing the model's output. *Label smoothing* (LS) (Szegedy et al., 2015) reaches this goal by adding noise to the one hot encoded labels: the target class is less certain while the other classes have non-zero probabilities. Therefore, it prevents the logit of the predicted class from being too large. Another technique is *confidence penalty* (CP) (Pereyra et al., 2017) with the aim of getting smoother and better calibrated models with less confident outputs. This is achieved by discouraging peaked distribution through penalizing low entropy output distributions. Meister et al. (2020) showed that both LS and CP are special cases of what they refer to as *generalized entropy regularization* (GER) which formalized a parametric regularizer for the model's outputs resulting in less overconfident predictions. Interestingly, link can be made between LS and temperature scaling, as shown in Müller et al. (2020), highlighting the prospect of calibrated models through a better suited loss function. Additionally, in the case of MC-Dropout, Laves et al. (2019) showed through observation and experimentation that temperature scaling outperforms confidence penalty and results in a better calibrated uncertainty. Yet, further post-hoc calibrating a model that was trained with a regularized loss could lead to worse calibrations as demonstrated empirically by Wang et al. (2021).

3.3.2 Pitfalls of model calibration and data augmentation

It is common to train modern DL models with data augmentation, such as mixup[11] (Zhang et al., 2018b), random erasing (Zhong et al., 2020) in addition to the classic transformations such as rotation, translation, resizing, flipping and cropping (Lecun et al., 1998; Simonyan and Zisserman, 2015). Although the motivations behind it is to achieve better generalization, it could negatively impact uncertainties, especially aleatoric uncertainty. Kapoor et al. (2022) discussed the relationship between the two and showed that data augmentation could lead to underconfident models, and it results in an inadequate representation of aleatoric uncertainty. To mitigate this drawback, Kapoor et al. (2022) propose tempering the posterior, as data augmentation softens the likelihood. Data

augmentation is not the only element that affect aleatoric uncertainty, likelihood tempering with a cold temperature also reduces aleatoric uncertainty, suggesting a close link between the two.

It is thus crucial to study, at least, model calibration whenever the global training schema is modified. Post-hoc calibration techniques, such as temperature scaling, do not faithfully model aleatoric uncertainty (Kapoor et al., 2022). For example, it was noted in (Ovadia et al., 2019) that temperature scaling has a limited effect (in terms of increase of uncertainty) on the shifted data, which should represent higher aleatoric uncertainty.

3.3.3 Epistemic Calibration

Thus far, we focused solely on the notion of calibration from an aleatoric perspective, it is hence important to discuss the epistemic aspect of the calibration. Although, as aforementioned, it could be relatively easy to establish a baseline to measure and calibrate aleatoric uncertainty, the same cannot be said about epistemic uncertainty. In fact, evaluating the calibration, in the epistemic sense, relies on the choice of the prior distribution. However, this choice is not unique and not objective. In the absence of such a ground-truth, it is common to assess the reliability of epistemic uncertainty through subsequent tasks (Mortier et al., 2023). Take the case of EDL for example, Jürgens et al. (2024) showed that it does not yield a calibrated and reliable epistemic uncertainty. As proven in their paper, the uncalibrated epistemic uncertainty in this model is due to the optimization process: the goal to have a model that is aware of its epistemic uncertainty while having only access hard labels. It can also be formalized in terms of *uncertainty levels*: a level-2 prediction cannot be adequately evaluated based only on level-0 observations.

Uncertainty levels. Aleatoric and epistemic uncertainty can be illustrated in a bi-level representation as shown in Figure 3.7. Here we have a binary classification task and the focus is on the estimated probability of the positive class. Furthermore, only in this paragraph, θ will refer to the probability of the positive class. While the following distinction is not limited to the binary classification, its use is for simpler and intuitive representation. In level 0, the prediction of the model consists "only" of the predicted class with no score for the confidence of the prediction (hard label prediction), and thus this level is the least important to us especially in the context of calibration. A point-estimate in the output space $\hat{\theta}$ (such as the softmax-probabilities) are considered as level 1 and provide information about aleatoric calibration: the closest $\hat{\theta}$ to the ground-truth θ^*, the better aleatoric calibration. Moreover, rather than only considering a point-estimate, epistemic uncertainty on level 2 extends this view through the use of a second-order distribution, which is a probability distribution about the point-estimate. Finally, it is worth mentioning that it is possible to obtain the information from any level if we have access to a higher level, and hence the former is a restrictive view of the latter.

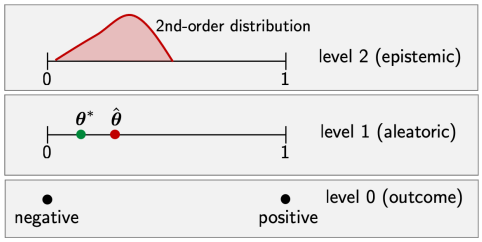


FIGURE 3.7: Illustration of the levels of uncertainty for a binary classification task, with θ is the probability of the positive class. Credits to Wimmer et al. (2023).

Credal sets. Being influenced by the subjective choice of the prior, evaluating epistemic calibration remains a challenging problem, even if we put aside the influence of the choice of the prior distribution. To this end, generalizing Bayesian inference to tackle the criticism of the choice of prior distributions, mainly noninformative priors, motivated the definition of *credal sets* (Walley, 1991; Bronevich and Klir, 2008; Shaker and Hüllermeier, 2021; Hüllermeier et al., 2022). In a

[11] A "new" datapoint is created as a convex combination of the existing datapoints for the inputs and the labels (one-hot label encoding).

nutshell, the idea behind credal sets is to replace a single prior distribution by a set of plausible prior distributions, and the (credal) set is then defined as the convex set of the BMAs resulting from the different priors. Epistemic uncertainty can be defined for credal sets as the size (volume) of the set. However, Sale et al. (2023) argue that while the volume is effective to capture epistemic uncertainty in the binary classification case, it proves less effective as the number of classes increases. The calibration aspect of this formalism was recently discussed in the work of Jürgens et al. (2025) for epistemic uncertainty by considering the convex combinations of the predictors: a credal set is calibrated (in distribution) if there exist a convex combination of the set that is calibrated in the strong sense (Widmann et al., 2019; Mortier et al., 2023).

3.4 Conclusions

In contrast to the well-defined problem of model calibration which is directly related to the predictive distribution, we illustrated in this chapter the vast choice for the prior distribution and its subjective impact on the calibration of epistemic uncertainty. As a result, it remains important to investigate the calibration of epistemic uncertainty, especially for commonly used Bayesian models for which this source of uncertainty is used. In the next chapter, we will study in depth the notion of epistemic calibration and show its differences and difficulties compared to aleatoric calibration from both a theoretical and a practical point of views. We also argue that the calibration of epistemic uncertainty can be studied through some fundamental principles, which will be formally stated.

CHAPTER 4
THEORY AND OBSERVATIONS: EXPLORING GAPS

"In theory, theory and practice are the same. In practice, they are not."
Albert Einstein

After discussing the different sources of uncertainties in Chapter 2 and the priors alongside the notion of model calibration in Chapter 3, we formulate in this chapter some fundamental properties of epistemic uncertainty. We argue that these properties are important to study the calibration of epistemic uncertainty. Empirically, we show that these theoretical properties are not fully verified in practice for some commonly used BNNs, and refer to this phenomenon as *the epistemic uncertainty hole* (Fellaji and Pennerath, 2023). Our work Fellaji et al. (2024) established the foundational concepts and initial findings, this chapter is an extended version of it, with in-depth and extensive experiments.

4.1 The Problem of Epistemic Uncertainty Calibration

Measuring reliable uncertainties is a challenging task, as discussed in Section 2.3.4, especially epistemic uncertainty. Undoubtedly, the information theoretical framework offers an ideal way to measure total uncertainty through the use of entropy, at least in a classification setting. For a uniform prediction, entropy has its maximum value and thus expresses total ignorance. This lack of certainty is either a result of aleatoric or epistemic uncertainty and could only be determined through faithfully measuring these two sources of uncertainty.

The idea of formally studying uncertainty measures from a rigorous mathematical perspective has been examined across a wide range of research papers. In Pal et al. (1993); Bronevich and Klir (2008) for example, the focus is on studying the validity of the total uncertainty, whereas aleatoric and epistemic uncertainties are examined in addition to total uncertainty in Wimmer et al. (2023); Sale et al. (2024). In the previously mentioned papers, the main purpose is to set a number of axioms which should be verified for a measure of uncertainty in order to be considered a valid measure.

Unlike model calibration which is related to the calibration of aleatoric uncertainty, the calibration of epistemic uncertainty is hard to quantify. In fact, the calibration of epistemic uncertainty relies on the choice of the prior distribution. Bengs et al. (2023); Wimmer et al. (2023) pointed out, simply and rightfully, the difficulty of evaluating the quality of an epistemic uncertainty measure as it depends on a distribution over the simplex (level 2), compared to aleatoric uncertainty which

is related to an element of the simplex (level 1). However, the defined properties for epistemic uncertainty are subject to criticism. Indeed, a recent paper (Sale et al., 2024) (with common authors) dropped some axioms regarding epistemic uncertainty that were established in Wimmer et al. (2023), such as epistemic uncertainty is maximal for the uniform distribution over the simplex, even through they contest the use of this distribution to represent epistemic ignorance.

To the best of our knowledge, the calibration of epistemic uncertainty is overlooked in the literature, in particular, the model-related principle (Section 4.2.3). Although some papers discussed the data-related principle (Section 4.2.2) and showed that it is unverified for common Bayesian deep learning models (Wimmer et al., 2023), they argued that it is due to the use of mutual information to measure epistemic uncertainty, hence related to the metric. As we will show shortly, the lack of verification of this principle is due to the miscalibration of these models rather than a problem with the mutual information measure itself. Additionally, the work of Bengs et al. (2023) associate the fail of faithfully representing epistemic uncertainty, in the case of EDL, to the loss function. Their conclusion being that, since the loss minimization operates on the level 1 given the ground truth labels (level 0), it is hard to train the model on the epistemic dimension.

In this chapter, we define rigorously two commonly known principles for epistemic uncertainty and show that they are satisfied in the information theoretical framework. We further illustrate from a practical point of view the issue of commonly used methods, Bayesian and Evidential, when estimating epistemic uncertainty. Although some observations are similar to those in Wimmer et al. (2023), our analysis in Fellaji and Pennerath (2023) was conducted independently of their work, is two-dimensional and covers DE, MC-Dropout, and EDL.

4.2 Fundamental Principles of Epistemic Uncertainty

By definition, epistemic uncertainty represents the uncertainty in a model due to incomplete knowledge, often related to uncertainty about its parameters. The knowledge being mostly related to the dataset and its size, it has a direct effect on epistemic uncertainty. It is commonly accepted in the deep learning community that training the model on more data increases its performance and generalization. However, while it is widely acknowledged that acquiring more datapoints should decrease epistemic uncertainty, this characteristic remains infrequently explored especially in the low data regime. The choice of the model could also be considered part of the incomplete knowledge and thus affecting epistemic uncertainty. As a larger model has more parameters than a smaller one, epistemic uncertainty is not invariant to this dimension and is expected to take this into consideration.

In the following, we will define properly these fundamental principles as axioms. We will use the notation $\mathcal{U}_{\mathcal{D},\mathcal{M}}(.)$ as an idealized measure of epistemic uncertainty, which only depends on the dataset $\mathcal{D}$ and the model $\mathcal{M}$ (trained on $\mathcal{D}$). More precisely, $\mathcal{U}_{\mathcal{D},\mathcal{M}}(.)$ operates on samples from the input space and thus computed per sample. The fundamental principles of epistemic uncertainty will be first stated for $\mathcal{U}_{\mathcal{D},\mathcal{M}}(.)$ and will be later shown to be valid in the particular case of mutual information. These principles were formally stated in our work (Fellaji et al., 2024, Section 3).

4.2.1 The simple case of Bayesian linear regression

To motivate the fundamental principles of epistemic uncertainty, we will examine exclusively in this section a regression task and use a *Bayesian linear regression model* $\mathcal{M}_{\mathcal{I}}$, as illustrated in Equation 4.1, with known homoskedastic variance $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and isotropic normal prior.

$$Y = \sum_{i \in \mathcal{I}} \Theta_i \, \psi_i(x) + \varepsilon \tag{4.1}$$

where the regressor functions ψ_i (*i.e.* transforming inputs to model linear or nonlinear relationships) are chosen in a large collection indexed by $\mathcal{I}$, and the parameters are independent and initially sampled from the normal prior $\Theta_i \sim \mathcal{N}(0, \sigma_0^2)$. For the sake of simplicity, suppose that the regressor functions are decorrelated:

$$\mathbb{E}\left[\psi(\boldsymbol{X})\,\psi(\boldsymbol{X})^T\right] = \mathtt{Id}$$

With $\mathtt{Id}$ being the identity matrix. It can be shown that, for a specific definite model $\mathcal{M}_\mathcal{I}$ (i.e. $|\mathcal{D}| \geq |\mathcal{I}|$), epistemic uncertainty $\mathcal{U}^v_{\mathcal{D},\mathcal{M}_\mathcal{I}}(x)$, for some (test) input x, can be estimated by the *difference of variances* under the variance decomposition approach (Equation 2.28) applied to the output space as:

$$\mathcal{U}^v_{\mathcal{D},\mathcal{M}_\mathcal{I}}(x) = \mathbb{V}[Y \mid x, \mathcal{D}] - \mathbb{V}[Y \mid \boldsymbol{\Theta}, x, \mathcal{D}] = \frac{|\mathcal{I}|}{\sigma_0^{-2} + |\mathcal{D}|\,\sigma^{-2}} \tag{4.2}$$

Analyzing the estimated epistemic uncertainty in Equation 4.2 leads to two notable observations. First, as we increase the size of the training set $\mathcal{D}$, epistemic uncertainty is expected to decrease, a property that is unsurprising and widely recognized among Bayesian practitioners. Second, for a fixed training set, epistemic uncertainty is positively correlated with the model capacity, defined as the size of the collection of indices $\mathcal{I}$. Therefore, increasing (respectively decreasing) the model capacity $\mathcal{I}$ is supposed to increase (respectively decrease) epistemic uncertainty. As priors on coefficients are independent, changing the model capacity can be done by either adding more regressor functions or by reducing the existing ones through some variable selection for example, leading to a submodel as we only consider a subset of $\mathcal{I}$. Consequently, the fundamental principles of epistemic uncertainty are defined to ensure these correlations with the size of the training set and the model capacity are as aforementioned.

While these principles are theoretically justified in the simple case of Bayesian linear regressor under some assumptions, it is natural to raise the question of whether they can be generalized to more complex models such as BDL, especially in the case of classification as it is the core of this thesis. In the following, we will examine these two principles from both theoretical and experimental perspectives.

4.2.2 Data-related principle

Arguably one of the most known property of epistemic uncertainty is what we refer to as *data-related* principle. It states that training the model on more datapoints should reduce epistemic uncertainty, and it is formally defined in Definition 4.1. This is the core property of epistemic uncertainty and should ideally be satisfied, as by training on more informative datapoints, the model has acquired more knowledge about the task. Hence, in the limit of infinite number of samples, the model is epistemically certain. We emphasize that this principle should be studied through the lens of subsets: we are systematically adding more samples to the initial training set rather than comparing to a different training set, larger and randomly selected. Training sets are thus analyzed with inclusion as a key order relation. In the following a model $\mathcal{M}$ denotes the pair $(f_{\boldsymbol{\theta}}, \mathrm{p}(\boldsymbol{\theta}))$ of the parameterized likelihood function $f_{\boldsymbol{\theta}}$ and its prior $\mathrm{p}(\boldsymbol{\theta})$.

Definition 4.1: Data-related principle

Given a model $\mathcal{M}$, a measure $\mathcal{U}^e$ of epistemic uncertainty satisfies the data-related principle with respect to a couple $(\mathcal{D}_1, \mathcal{D}_2)$ of training samples if

$$\forall x, \quad \mathcal{U}^e_{\mathcal{D}_1,\mathcal{M}}(x) \geq \mathcal{U}^e_{\mathcal{D}_1\cup\mathcal{D}_2,\mathcal{M}}(x)$$

It's important to stress that this principle as expressed by Definition 4.1 is not a universal property, but only a theoretical ideal. Indeed, one can easily imagine the existence of observations that will increase epistemic uncertainty. To understand this, let's take the very simple (albeit improbable) example of a very large data set $\mathcal{D}_1 = \{x_i, y_i\}$ of examples, all attached to the same class (i.e. $\forall i,\ y_i = 1$). A correctly trained Bayesian model $\mathcal{M}$ will systematically predict the output $Y = 1$ with almost zero predictive uncertainty and therefore, zero epistemic uncertainty as well. If we later add supplementary training samples $\mathcal{D}_2$, for which the classes are now uniformly distributed, both the aleatoric and epistemic uncertainties will increase, infringing the principle.

While it is possible to contradict the data-related principle with unrealistic counter-examples, we may wonder if it can be verified statistically, *i.e.* on average, under some mild conditions. Indeed, choosing mutual information (Equation 2.19) as the metric of epistemic uncertainty, we can show that, under the *i.i.d.* assumption of samples, the data-related principle is true in expectation (Theorem 4.1). From a practical perspective, we expect a decrease of mutual information, evaluated on the test set, when the model is trained on additional samples. We highlight that the data follows an *i.i.d.* (independent and identically distributed) assumption.

Theorem 4.1

Given a model $\mathcal{M}$, epistemic uncertainty, as measured with mutual information, verifies the data-related principle in expectation with respect to new random *i.i.d.* samples $\mathcal{D}_2$:

$$\forall (x,\ \mathcal{M},\ \mathcal{D}_1), \quad \mathcal{U}^e_{\mathcal{D}_1,\mathcal{M}}(x) \geq \mathbb{E}_{\mathcal{D}_2}\left[\mathcal{U}^e_{\mathcal{D}_1\cup\mathcal{D}_2,\mathcal{M}}(x)\right]$$

Proof. For simplicity and without loss of generality, we assume that $\mathcal{D}_2$ consists of a single observation $(x_2,\ Y_2)$. For brevity, we denote $\kappa = (x,\ \mathcal{D}_1,\ x_2)$, such that x is a "test" input for which we seek to evaluate the epistemic uncertainty reflected in its output Y. Under the *i.i.d.* assumption, we have:

$$\begin{aligned}\mathbb{I}(Y;\ \boldsymbol{\Theta} \mid x,\ \mathcal{D}_1) &= \mathbb{I}(Y;\ \boldsymbol{\Theta} \mid \kappa)\\ \mathbb{I}(Y;\ Y_2 \mid \boldsymbol{\Theta},\ \kappa) &= 0\end{aligned}$$

Hence:

$$\begin{aligned}\mathcal{U}^e_{\mathcal{D}_1,\mathcal{M}}(x) - \mathbb{E}_{\mathcal{D}_2}\left[\mathcal{U}^e_{\mathcal{D}_1\cup\mathcal{D}_2,\mathcal{M}}(x)\right] &= \mathbb{I}(Y;\ \boldsymbol{\Theta} \mid x,\ \mathcal{D}_1) - \mathbb{I}(Y;\ \boldsymbol{\Theta} \mid x,\ \mathcal{D}_1,\ x_2,\ Y_2)\\ &= \mathbb{I}(Y;\ \boldsymbol{\Theta} \mid \kappa) - \mathbb{I}(Y;\ \boldsymbol{\Theta} \mid \kappa,\ Y_2)\\ &= \mathbb{H}(Y \mid \kappa) - \mathbb{H}(Y \mid \boldsymbol{\Theta},\ \kappa) - \mathbb{H}(Y \mid \kappa,\ Y_2) + \mathbb{H}(Y \mid \boldsymbol{\Theta},\ \kappa,\ Y_2)\\ &= \mathbb{H}(Y \mid \kappa) - \mathbb{H}(Y \mid \kappa,\ Y_2) + \mathbb{H}(Y \mid \boldsymbol{\Theta},\ \kappa,\ Y_2) - \mathbb{H}(Y \mid \boldsymbol{\Theta},\ \kappa)\\ &= \mathbb{I}(Y;\ Y_2 \mid \kappa) - \mathbb{I}(Y;\ Y_2 \mid \boldsymbol{\Theta},\ \kappa)\\ &= \mathbb{I}(Y;\ Y_2 \mid \kappa) \geq 0\end{aligned}$$

□

The importance of the *i.i.d.* assumption. The data-related principle relies on the *i.i.d.* assumption of $\mathcal{D}$, and it appears important in its proof to justify the independence of Y and Y_2 given $\boldsymbol{\Theta}$ and κ. Fortunately, this assumption is widely applicable, or at least applied, in the context of machine learning.

4.2.3 Model-related principle

The second most fundamental dimension when studying the validity of an epistemic uncertainty measure is the model. Its importance is clearly visible in the formula of the measure: $\mathcal{U}_{\mathcal{D},\mathcal{M}}(.)$. The effect of the model depends on the parametric function f_θ parameterized by a vector $\boldsymbol{\theta}$, and the choice of the prior distribution $\mathrm{p}(\boldsymbol{\theta})$. It seems natural to consider the influence of the model parameters on epistemic uncertainty. However, in order to compare the comparable, the scaling of the model (*i.e.* changing its complexity) should be carried out systematically. One solution to adjust the models could be by using *submodels* (Definition 4.2).

Under the formalism of submodels, the scaling of the model is comparable to the way the data-related principle is stated. As a matter of fact, the larger model embodies the submodels, thus the expressive power of the former is higher than the latter. Moreover, the functional hypothesis space of the submodel is included in that of the larger model. Hence, the inductive bias of the models are comparable, and the inclusion is on the functional hypothesis space. While it is possible to extend the definition of submodels more broadly (through the change of variable for example), this generalization is not necessary for our analysis.

Definition 4.2: Submodels

We say model $\mathcal{M}_a = \left(f^a_{\boldsymbol{\theta}_a}, \mathrm{p}_a(\boldsymbol{\theta}_a)\right)$ is a *submodel* of model $\mathcal{M}_b = \left(f^b_{\boldsymbol{\theta}_b}, \mathrm{p}_b(\boldsymbol{\theta}_b)\right)$, denoted by $\mathcal{M}_a \leq_{\mathcal{M}} \mathcal{M}_b$, if $\boldsymbol{\theta}_a$ is a subset of parameters $\boldsymbol{\theta}_b$ so that $\boldsymbol{\theta}_b = (\boldsymbol{\theta}_a, \boldsymbol{\theta}_{b'})$ and there exists a constant vector $\boldsymbol{\theta}^0_{b'} \in \Omega_{\boldsymbol{\theta}_{b'}}$ such that:

$$\forall(\boldsymbol{\theta}_a \in \Omega_{\boldsymbol{\theta}_a}), \quad f^a_{\boldsymbol{\theta}_a} = f^b_{(\boldsymbol{\theta}_a, \boldsymbol{\theta}^0_{b'})} \text{ and } \mathrm{p}_a(\boldsymbol{\theta}_a) = \mathrm{p}_b\left(\boldsymbol{\theta}_a \mid \boldsymbol{\Theta}_{b'} = \boldsymbol{\theta}^0_{b'}\right)$$

It is straightforward to show this relation $\leq_{\mathcal{M}}$ defines a (partial) order over models. We can now define the model-related principle for epistemic uncertainty (Definition 4.3). This is a formal reformulation of the Occam Razor's principle: as the larger model (in the sense of submodels) has more parameters and its functional hypothesis space includes that of the submodel, the data can be explained by more functions in the case of the larger model and the choice of model candidates is wider. As a result, epistemic uncertainty should be higher for the larger model.

Definition 4.3: Model-related principle

Given a set $\mathcal{D}$ of samples, a measure $\mathcal{U}^e$ of epistemic uncertainty satisfies the model-related principle with respect to a couple $(\mathcal{M}_a, \mathcal{M}_b)$ of models, if

$$\mathcal{M}_a \leq_{\mathcal{M}} \mathcal{M}_b \implies \forall x, \quad \mathcal{U}^e_{\mathcal{D},\mathcal{M}_a}(x) \leq \mathcal{U}^e_{\mathcal{D},\mathcal{M}_b}(x)$$

Again this principle is not universal. It must be understood as an idealized property that we wish to verify. As with the data-related principle, it is possible to construct counterexamples of nested models that violate this second principle. However, focusing again on the mutual information metric, we can check that this principle is verified in expectation, at least in some sense. Indeed, given a model $\mathcal{M}_b$ whose parameters are decomposed in two subsets, i.e. $\boldsymbol{\theta}_b = (\boldsymbol{\theta}_a, \boldsymbol{\theta}_{b'})$, it is true that

$$\begin{aligned}
\mathcal{U}^e_{\mathcal{D},\mathcal{M}_b}(x) &= \mathbb{I}(Y; \boldsymbol{\Theta}_b \mid \mathcal{D}, x) \\
&= \mathbb{I}(Y; \boldsymbol{\Theta}_a, \boldsymbol{\Theta}_{b'} \mid \mathcal{D}, x) \\
&= \mathbb{I}(Y; \boldsymbol{\Theta}_a \mid \mathcal{D}, x) + \mathbb{I}(Y; \boldsymbol{\Theta}_{b'} \mid \mathcal{D}, x, \boldsymbol{\Theta}_a) \\
&\geq \mathbb{I}(Y; \boldsymbol{\Theta}_a \mid \mathcal{D}, x)
\end{aligned}$$

This amounts to say that on average the submodels $\mathcal{M}_a$ of $\mathcal{M}_b$ (as defined in Definition 4.2 and weighted by posterior $\boldsymbol{\theta}^0_{b'} \sim \mathrm{p}(\boldsymbol{\theta}_{b'} \mid \mathcal{D}, x)$) verify the model-related principle. However, this result is of little interest, as contrary to subsets of samples with the data-related principle, submodels $f^a_{\boldsymbol{\theta}_a} = f^b_{(\boldsymbol{\theta}_a, \boldsymbol{\theta}^0_{b'})}$ are not drawn randomly, i.e. $\boldsymbol{\theta}^0_{b'}$ is set prior to the observation of data $\mathcal{D}$ and consequently cannot be drawn from $\mathrm{p}(\boldsymbol{\theta}_{b'} \mid \mathcal{D}, x)$. Said otherwise, we cannot identify $\mathcal{U}^e_{\mathcal{D}, \mathcal{M}_a}(x)$ with $\mathbb{I}(Y; \boldsymbol{\Theta}_a \mid \mathcal{D}, x)$, as indeed,

$$\mathbb{I}(Y; \boldsymbol{\Theta}_a \mid \mathcal{D}, x) = \mathbb{E}_{\boldsymbol{\theta}^0_{b'} \sim \mathrm{p}(\boldsymbol{\theta}_{b'} \mid \mathcal{D}, x)} \left[\mathbb{I}\left(Y; \boldsymbol{\Theta}_a \mid \mathcal{D}, x, \boldsymbol{\theta}^0_{b'}\right) \right] \neq \mathbb{I}\left(Y; \boldsymbol{\Theta}_a \mid \mathcal{D}, x, \boldsymbol{\theta}^0_{b'}\right)$$

We however think that the inequality $\mathcal{U}^e_{\mathcal{D}, \mathcal{M}_a}(x) \leq \mathcal{U}^e_{\mathcal{D}, \mathcal{M}_b}(x)$ can become true under some more restrictive hypothesis to be specified. If our intuition is correct and can be proved somehow, the mutual information will be shown as an adequate measure of epistemic uncertainty with respect to the data-related (Theorem 4.1) and model-related principles, at least in expectation, mitigating or even cancelling the critics detailed in Section 2.3.4.

From a practical point of view, given a fixed training set, moving from a model to one of its submodels, we expect epistemic uncertainty to reduce on the test set. While the model-related principle could not be proved for mutual information, the experiments on conflictual loss tend to confirm that this metric empirically verifies the principle when computed from the outputs of a correctly epistemically calibrated model. For the same reasons that failing to correctly set the hyper-parameters of a model could affect the training of the model and lead to suboptimal performance (even for a simple measure as the accuracy), or the fact that DL models are overconfident and hence unreliable from a calibration perspective, we argue that questioning the learning process and the epistemic calibration of practical BNNs rather than the metric is inline with DL "established procedure", especially for a theoretical sound measure, like mutual information.

Finally, one can look at the model-related principle from the lens of the lottery ticket hypothesis (Frankle and Carbin, 2019). More precisely, the larger model offers more explanations for the training data compared to the submodel, which will match at best the performance of the parent model. To this end, the larger model is expected to have a higher epistemic uncertainty compared to the submodel. In addition, considering a properly initialized submodel is inline with the results presented by Frankle and Carbin (2019) where appropriately initialized the submodels is crucial.

Remark *(Practical evaluation).* When evaluated empirically, both principles will be evaluated in expectation on the test set, similarly to the literature on machine learning. Once a model $\mathcal{M}$ is trained on a training dataset $\mathcal{D}$, the evolution of epistemic uncertainty will be based on its average value on the (fixed) test set. Given the large size of the test set (around $10k$ samples), the average value is statistically reliable.

4.3 The Epistemic Uncertainty Hole

As highlighted earlier, we would ideally like epistemic uncertainty, as computed with mutual information, to verify the two fundamental principles. We thus expect in practice, for a Bayesian model, the decrease of epistemic uncertainty on the test set when the model is trained on additional datapoints. Besides, we explore the compliance with the model-related principle for BNNs. Our exploratory analysis demonstrates a divergence between theory and practice.

4.3.1 Impact of the training set size

Given a deep ensemble of ten ResNet18 models, we are interested first in evaluating the verification of the data-related principle for this Bayesian models. The experiment consists of taking a small

subset of CIFAR10, evaluate the trained model on the test, add new samples to the initial train subset and repeat the process. The results are presented in Figure 4.1. Without any surprise, the accuracy of the model benefits form the additional training samples. However, when looking at the evolution of epistemic uncertainty on the test set, the results are contradictory to the data-related principle. More precisely, this principle is not verified especially in the low data-regime with a tiny number of samples. Notably, this observation would not have been possible if we had started the experiment with 800 datapoints. As the accuracy curve is coherent and does not exhibit any abnormality, we refer to this observation as *the epistemic uncertainty hole* (Fellaji and Pennerath, 2023), and it will be explored further in this thesis.

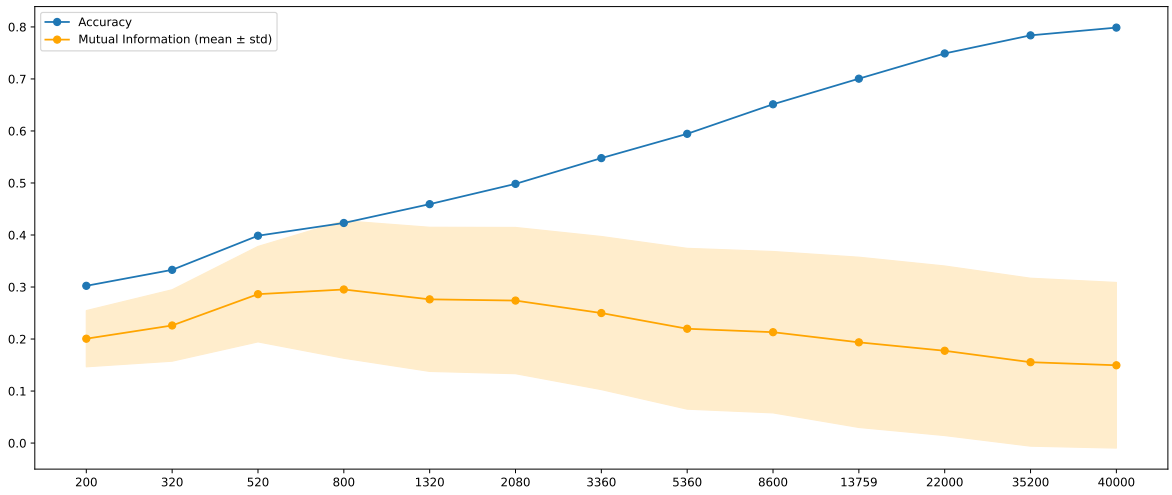


FIGURE 4.1: Evolution of the accuracy and epistemic uncertainty on the test set as a function of the size of the training set (x-axis), on a logarithmic scale, for an ensemble of 10 ResNet18. For the mutual information, we report the mean and the standard deviation on the entire test set. The values of epistemic uncertainty are normalized between 0 and 1 (division by the maximum value: $\log(10)$).

In Wimmer et al. (2023, Figure 5), similar observations are reported in the region of 1% to 5% of the training set, which roughly translated to the region of 400 to 2000 in the x-axis. The experimental setup is quite similar to ours as they also use a deep ensemble of ten models, emphasizing the validity of our observations. Although in their work they explain the epistemic uncertainty hole by the inadequacy of measuring epistemic uncertainty through mutual information, we explore a different perspective motivated by the verification of the data-related principle for mutual information (Theorem 4.1).

Due to the low accuracy of models trained on smaller datasets, we decided to split the test set into misclassified and correctly classified samples to study the evolution of epistemic uncertainty (Figure 4.2). In the low-data regime, we notice an overlap of epistemic uncertainty between the misclassified and correctly classified test samples. Indeed, the model is epistemically uncertain about both classes of samples. As the size of training set increases, the gap between them grows progressively. Moreover, after an overall increase of epistemic uncertainty on the misclassified samples, it showed a plateau whereas epistemic uncertainty on the correctly classified samples exhibit the epistemic uncertainty hole and tends to zero for large training set sizes. While epistemic uncertainty in misclassified samples shows an increasing trend, it exhibits two phases on the entire test set. This may be attributed to the improved accuracy, resulting in fewer misclassifications.

Throughout this chapter, we will explore the lack of epistemic calibration for commonly used BNNs, and argue that the hole of epistemic uncertainty is a manifestation of this phenomenon. While Bengs et al. (2023) focused on EDL, we extend the analysis from a different angle on more

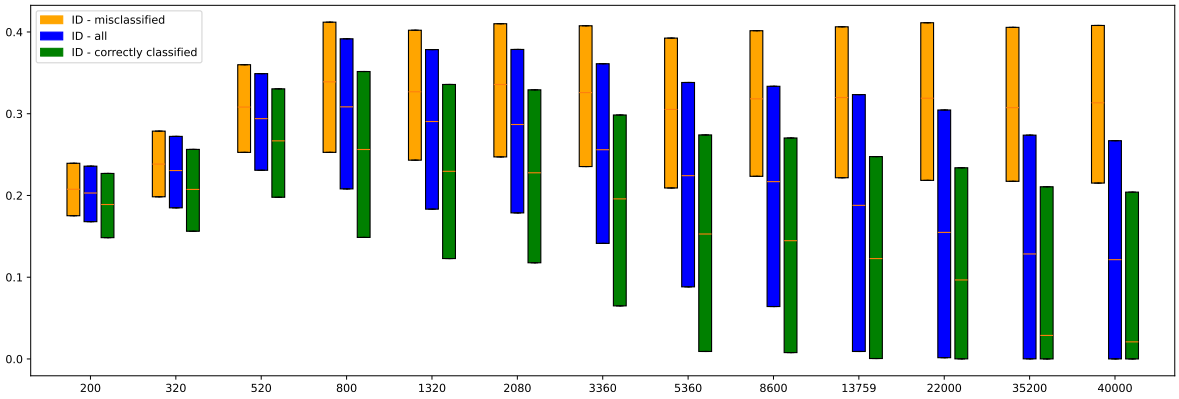


FIGURE 4.2: Box plots of normalized epistemic uncertainty on the in-distribution (ID) test set for models in Figure 4.1. ID-all (center-blue) for the entire test set, ID-mis (left-orange) and ID-good are for the misclassified and correctly classified samples by the model respectively. The box plots show the median, first and third quantiles.

models that have good predictive capabilities but for which epistemic uncertainty was not fully and rigorously investigated. In the next section, more experiments are conducted with the goal studying the two fundamental principles of epistemic uncertainty.

4.3.2 A dual-dimensional exploration

The previous experiment on CIFAR10 with a DE of ResNet18 models showed the epistemic uncertainty hole resulting from varying the size of the training set, in contradiction to the data-related principle. In this section, we will show that, for commonly used Bayesian models, not only the data-related is invalidated, but also the model-related principle. The focus of the analysis below is on simple tasks with simple models.

Let's take the simplest case of an MLP model with two hidden layers. One can get a subset of an MLP by simply reducing the size of the hidden layers, resulting in a less complex model. We will thus vary the size of the MLP in order to evaluate the model-related principle. Additionally, the protocol for studying the data-related principle will be similar to Section 4.3.1.

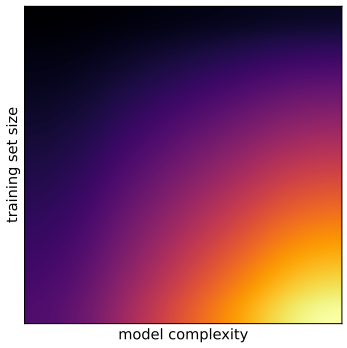


FIGURE 4.3: Ideal distribution of epistemic uncertainty. Darker colors for low uncertainty.

In a nutshell, the effect of the size of the training set and the model complexity will be studied on different metrics, especially epistemic uncertainty as measured by mutual information, and the results will be displayed in the form of two-dimensional heatmaps (as in Figure 4.3). The model complexity is presented on the x-axis and the size of the training set is on the y-axis. For example, the ideal distribution of epistemic uncertainty is illustrated in Figure 4.3 in accordance with the data and model related principles: the maximum of epistemic uncertainty is achieved for the largest model trained with the smallest training set, which represents the lower right corner in the heatmap.

In addition to the non-linearity applied after each linear layer, we included Dropout layers. They have a regularization effect and allow also a Bayesian interpretation of the model through MC-Dropout (MC-D) (Gal et al., 2017). Furthermore, two supplementary families of Bayesian models were tested: Deep Ensembles (DE) (Lakshminarayanan et al., 2017) and EDL (Sensoy et al., 2018). Moreover, the effect on epistemic uncertainty of model calibration, from an aleatoric perspective, was tested with label smoothing (LS) (Szegedy et al., 2015) and confidence penalty

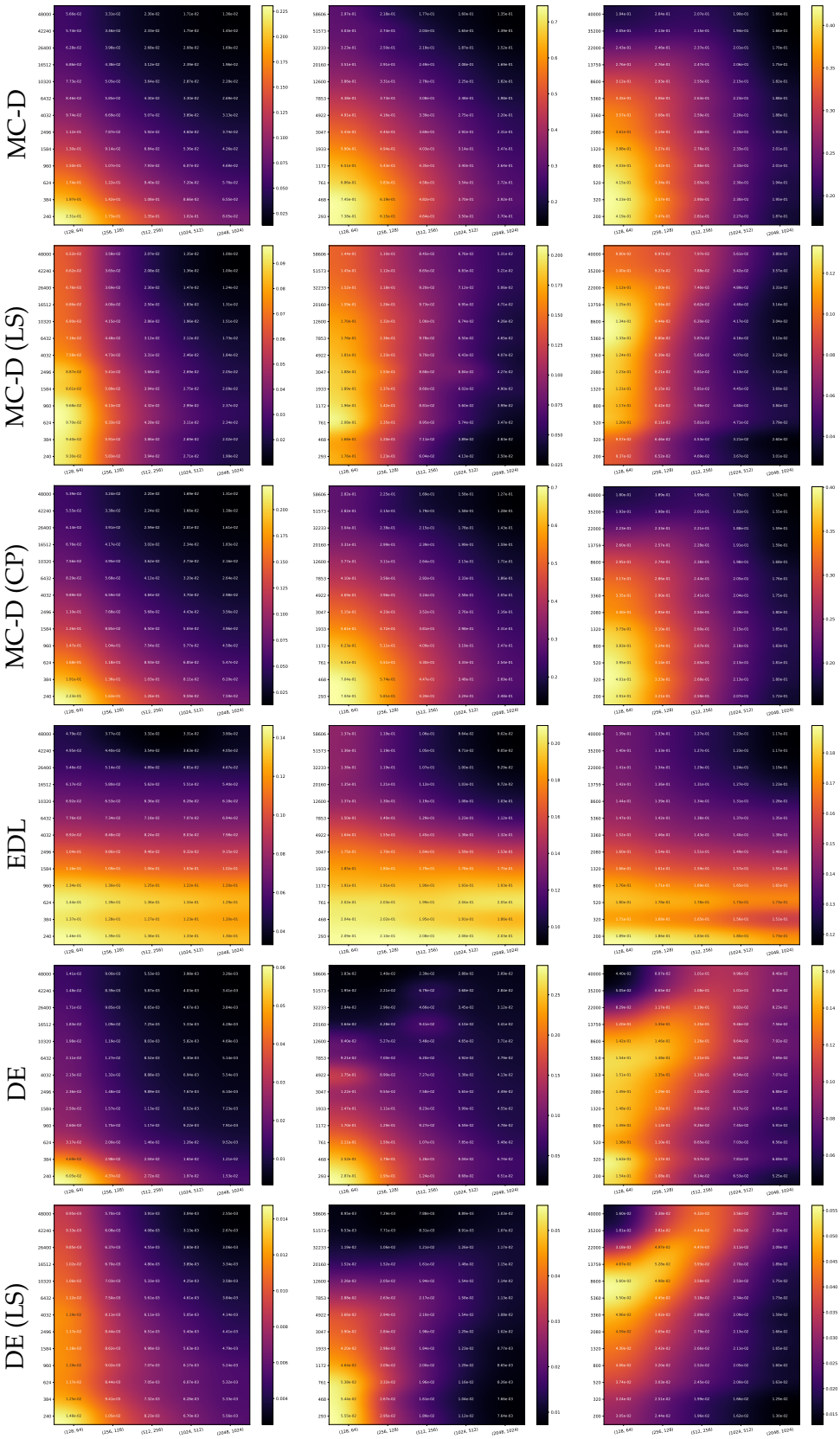


FIGURE 4.4: Mean of epistemic uncertainty on the test set after training the model with the model complexity is presented on the x-axis and the size of the training set is on the y-axis.

(CP) (Pereyra et al., 2017). As these calibration techniques lead to realistic and less confident outputs, we are thus interested in evaluating their influence on the model uncertainty.

As the MLP model could lead to a high inductive bias in the case of CIFAR10, a pretrained ResNet34 was used where only the classification part was changed and the features part was kept frozen. This is equivalent, from an implementation point of view, of using the feature blocks as a feature extractor to encode the images once for all, and then use the encoded images to train the MLPs on the classification task. The performance gain is significant as we avoid computing image embeddings with the frozen ResNet for each batch and less GPU memory is required.

The results are shown in Figure 4.4 with a summary of each heatmap in Table 4.1. The latter aggregates quantitatively the verification of the data and model related principles by measuring the frequency with which the answer to the following question is true: if we increase the size of the training set (respectively the model complexity), will epistemic uncertainty decrease (respectively increase)? These observations were first shown in Fellaji and Pennerath (2023); Fellaji et al. (2024). In the following, these results will be reviewed with additional elements of analysis.

		MC-D	MC-D (LS)	MC-D (CP)	EDL	DE	DE (LS)
Data-related principle	MNIST	100%	78%	98%	92%	100%	92%
	SVHN	92%	55%	92%	88%	88%	65%
	CIFAR10	77%	38%	77%	92%	42%	20%
Model-related principle	MNIST	0%	0%	0%	6%	0%	0%
	SVHN	0%	0%	0%	15%	17%	19%
	CIFAR10	12%	0%	8%	6%	15%	12%

TABLE 4.1: Quantitative summary of Figure 4.4. MC-D for MC-Dropout, DE for Deep Ensembles, LS for label smoothing and CP for confidence penalty. The percentages for data-related (respectively model-related) principle represent the frequencies with which jumping to the next larger dataset (respectively to the next smaller model) results in a decrease of mutual information.

Overall, the tested Bayesian models suffer from the hole of epistemic uncertainty in the two dimension, with the paradox being higher for the model-related principle. By focusing on the data-related principle, the evolution of epistemic uncertainty seems somehow consistent for MNIST, with an exception for the MC-Dropout with LS. MNIST being a simple classification task, the incoherence becomes larger as we deal with more complex task. Not only are the results worse, but the peak of epistemic uncertainty does not occur with the smallest training set resulting in a change of phases somehow similar to what we noticed in the case of ResNet18 (Figures 4.1 and 4.2).

The effect of the model complexity is the most problematic with nearly all the configurations failing completely. For these models, epistemic uncertainty is generally decreasing when increasing the model complexity. Since mutual information can be viewed as a measure of divergence of the outputs, a collapsing value indicates similarities in the outputs for complex models. Similar to the observations in ResNet models (Section 4.1), the evolution of the model accuracy in the heatmaps is as expected (Figure B.1): increasing with the size of the training set and slightly with model complexity. Additionally, there is no noticeable difference in accuracy between the models when tested on the same dataset.

Finally, the calibration of aleatoric uncertainty has a negative impact on the calibration of epistemic uncertainty: the hole of epistemic uncertainty is more noticeable for calibrated models, when LS or CP were used. By looking at the values of epistemic uncertainty in the heatmaps, we notice a reduction in terms of the mean on the test set for the LS versions. This confirms the observation highlighted in the work of Müller et al. (2020): label smoothing encourages the model

to treat each incorrect class as having an equal probability, therefore it artificially increases the measured aleatoric uncertainty while decreases epistemic uncertainty. The calibration measure ECE (Figure B.3) shows that for MNIST, LS leads to a higher ECE score overall whereas in the low data regime for SVHN and CIFAR10 it results in better calibrated models.

4.4 The Collapse of Epistemic Uncertainty

For commonly used BNNs, the evolution of epistemic uncertainty is paradoxical from the perspective of the fundamental principles. On one hand, adding more samples to training set could lead to an increase in epistemic uncertainty. On the other hand, these models fail at faithfully expressing their epistemic uncertainty for complex models. The latter pitfall is more pronounced in practice, leading to a collapse for epistemic uncertainty as the model becomes more expressive.

In this part of the discussion, we will investigate broadly from a macro and micro perspectives the evolution of epistemic uncertainty. We will visualize, for different models, the histograms on the entire test set of epistemic uncertainty as function of the two dimensions (Section 4.4.1). By doing so, we gain additional insights on epistemic uncertainty and its collapse. Furthermore, we look at the models predictions, from a logits and a softmax-probabilities point of views, for a selection of inputs and again from a two-dimensional aspect (Section 4.4.2). The selection of the inputs will be motivated such that they result in a high or low epistemic uncertainty.

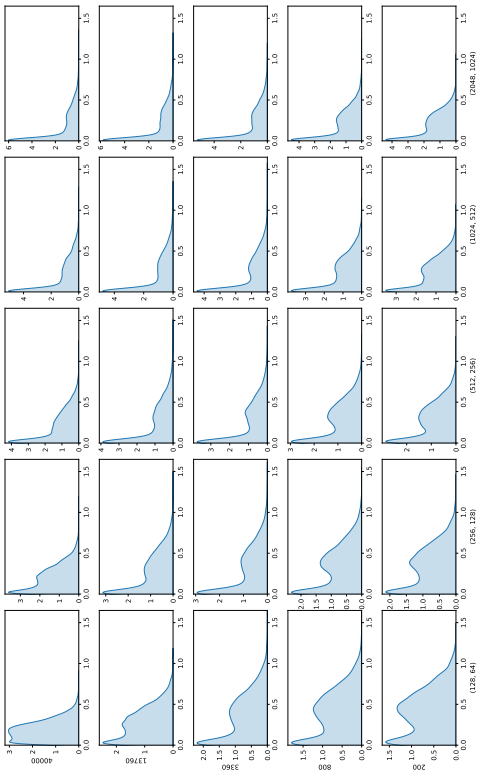
4.4.1 Expanding the macro perspective

The previous section illustrated the evolution of different metrics by comparing the average value on the test set. Although this value is statistically valid within the context of our analysis, we will further discuss the paradoxes regarding the principles of epistemic uncertainty by focusing on a subset of the experiments. The macro analysis consists of analyzing the histogram of epistemic uncertainty on the entire test set rather than the single average value as in Figure 4.4. To this end, we will mainly explore the models trained on CIFAR10 as they showed the most contradictions. For conciseness, the analysis will not cover all the sizes of the training sets reported in the previous section. We make sure that the selection in the y-axis is equally spaced. Additionally, the histograms of epistemic uncertainty on the test sets are plotted for four models: MC-Dropout, MC-Dropout with LS, DE and EDL. The results are shown in Figure 4.5.

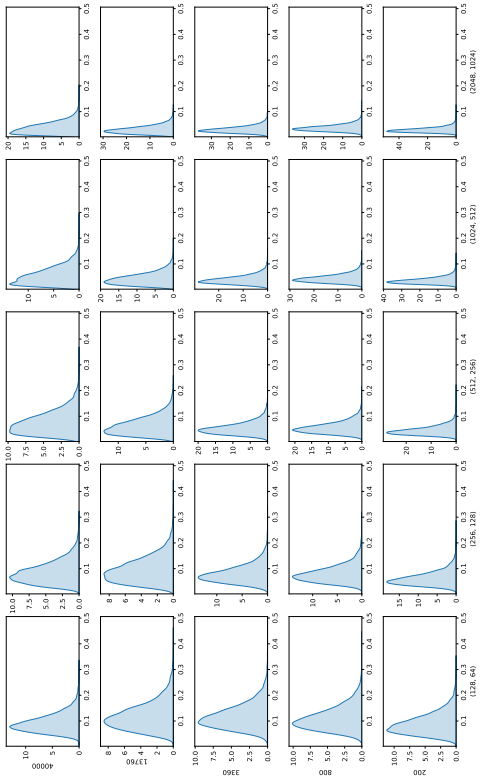
Overall, and expect for EDL, we notice that epistemic uncertainty concentrates around zero while its support shrinks when model complexity increases. The results with MC-Dropout (Figure 4.5a) and DE (Figure 4.5d) show that epistemic uncertainty concentrates around zero while its support shrinks when model complexity increases. The support is narrower for DE than for MC-Dropout. Moreover, MC-Dropout with LS (Figure 4.5b) showcased the less variability for epistemic uncertainty and the smallest values. Finally, the case of EDL (Figure 4.5c) is quite interesting as the histograms show a strong similarity in both shape and support. Distributional uncertainty, used a proxy for epistemic uncertainty, appears artificial as it seems uninfluenced by the two considered dimensions: the value at its peak decreases only slightly as the model is trained on more data. We can also notice that the position of the peak decreases as the size of the training of set increases, and slightly as the model complexity increases in contradiction with to the model-related principle.

4.4.2 Focusing on specific inputs

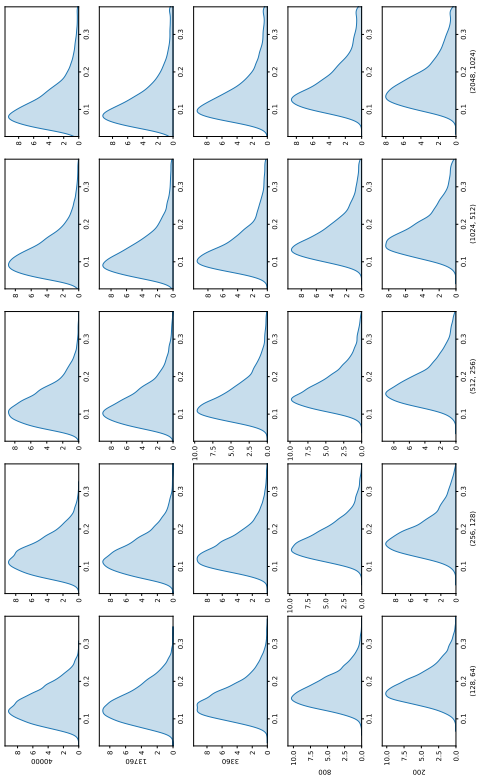
In the previous discussions, we analyzed the results from a macro perspective by either reporting the average performance on the test set or by analyzing the histograms for a set of configurations. In this section, we shift our focus to a micro perspective, analyzing the results in detail through a closer examination of a few specific examples. The main idea is to select critical inputs, that either



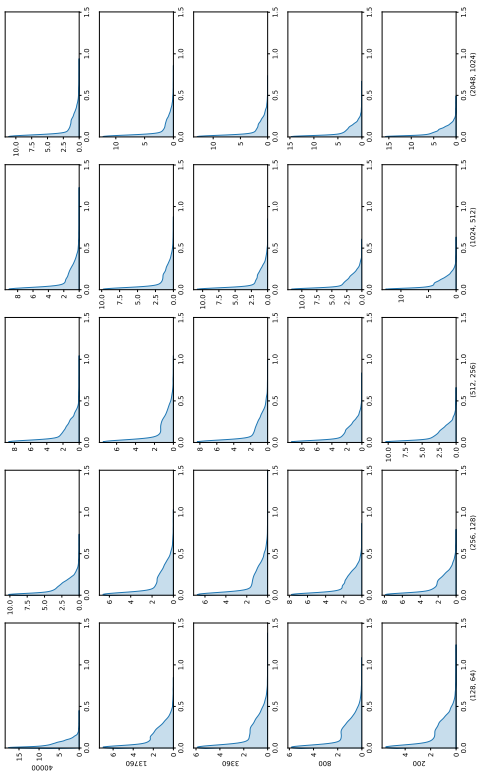
(A) MC-Dropout



(B) MC-Dropout LS



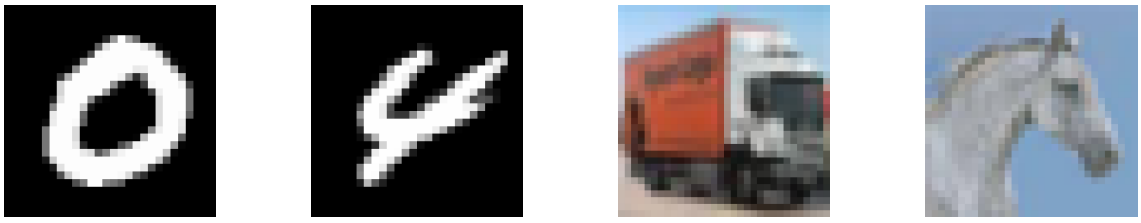
(C) EDL



(D) DE

FIGURE 4.5: Histograms of epistemic uncertainty on the test set of CIFAR10. The limits of the x-axis are the same on each figure, and the plots are rotated 90 degrees counterclockwise for a clearer presentation.

maximize of minimize epistemic uncertainty, and to inspect how the model outputs change as we increase the size of the training set and the complexity of the model. The choice of the inputs (Figure 4.6) is made using a heuristic approach: the inputs are selected such that they maximize (or minimize) epistemic uncertainty for all the models. The analysis will be carried out for MNIST and CIFAR10[12] by considering two types of models: MC-Dropout and DE.



(A) MNIST (6114 – 0) low epistemic uncertainty

(B) MNIST (6769 – 4) high epistemic uncertainty

(C) CIFAR10 (8931 – 9) low epistemic uncertainty

(D) CIFAR10 (6158 – 7) high epistemic uncertainty

FIGURE 4.6: Examples of images resulting in high or low epistemic uncertainty from the test sets of MNIST and CIFAR10. The indices and the true labels are reported under each image (index – label).

We look at the outputs from the lens of logits and of softmax probabilities, on the left and the right of each figure respectively, and rotated 90° counterclockwise (as in Figure 4.7 for example). The first ten columns represent the output of either the individual models in the DE, or a sampled model in the case of MC-Dropout, thus $\boldsymbol{\theta}_i \sim \mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$. The last column is the average of the first ten curves. On the rows, we have different sizes for the training set (Similar to Section 4.4.1.) and each subfigure illustrate the evolution as a function of the complexity of the model.

For the samples with low epistemic uncertainty, we notice for both (Figures 4.6a and 4.6c) that, to some extent, all the curves look similar from a softmax perspective with perfectly confident predictions. As a matter of fact, in addition to the models correctly classifying the images, the total uncertainty associated with these images is nearly zero, regardless of the size of the training set and the model complexity, hence epistemic uncertainty is also negligible. As the results in the case of DE for Figure 4.6a and MC-Dropout for Figure 4.6c are comparable to Figure 4.7 and Figure 4.8, respectively, we leave them out for the sake of brevity (see Figure C.6 and Figure C.4). The gap between the logit for the predicted class and the others is significantly higher for these images leading to confident predictions.

A general remark about the logits is that their amplitude is significantly higher for the correctly and confidently classified samples than for the hard-to-classify samples. In addition, the amplitude tends to increase also with model complexity in the case of CIFAR10 especially in the high data-regime, whereas the increase is less noticeable for MNIST. This observation is in line with the use of logits as a proxy for estimating the model's confidence (Liu et al., 2020b). Moreover, when comparing the curves of the logits, the ranking is highly correlated for the same input when using MC-Dropout and DE.

By looking now at the samples leading to a high epistemic uncertainty (Figures 4.9, 4.10, 4.11 and 4.12), the disagreement between the predictions is mostly visible in the low-data regime and even more observable with MC-Dropout. Due to the sensibility of the softmax function to the gap between the logits, as the size of the training set increases, the gap between the softmax probabilities will also increase, leading to more confident predictions. It is important to highlight

[12] As a reminder, here is the mapping from indices to labels for CIFAR10: {0: airplane, 1: automobile, 2: bird, 3: cat, 4: deer, 5: dog, 6: frog, 7: horse, 8: ship, 9: truck}.

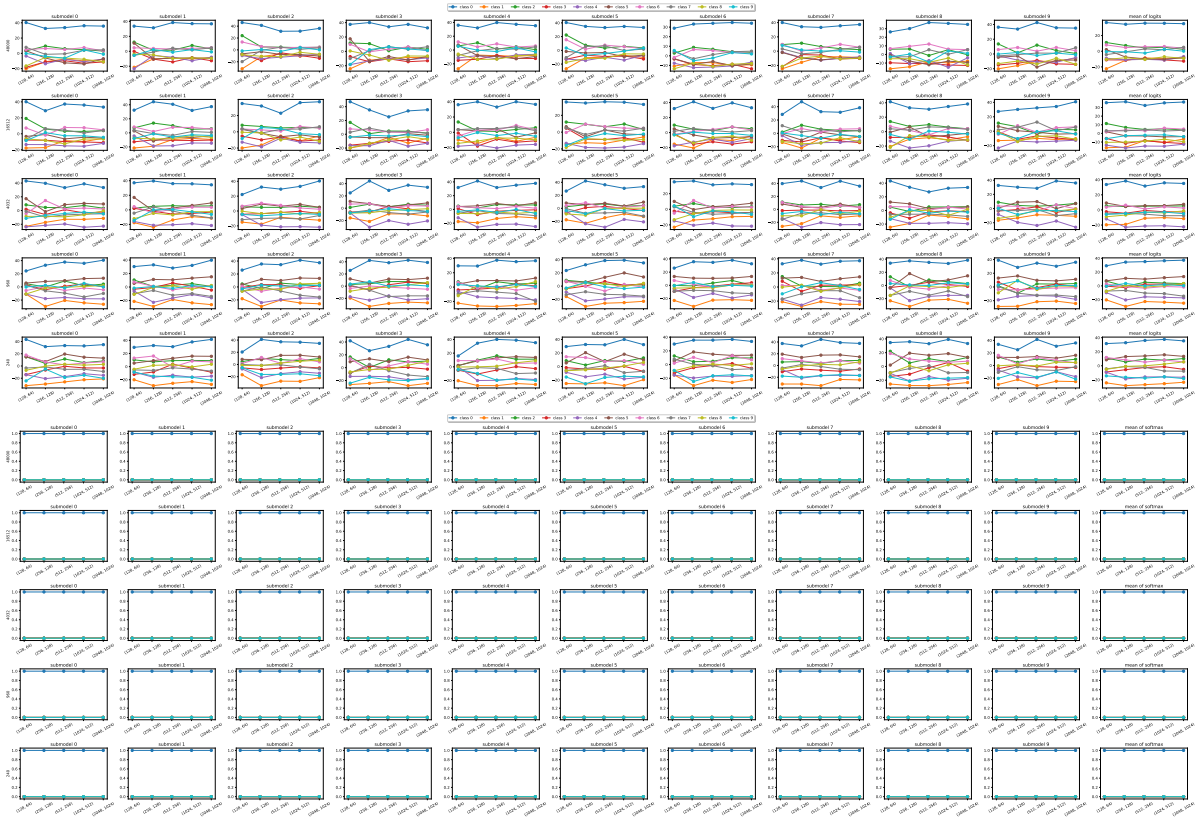


FIGURE 4.7: Predictions by MC-Dropout for Figure 4.6a (the true label is 0). Logits on the top and Softmax on the bottom. See Figure C.2 for a higher-resolution version.

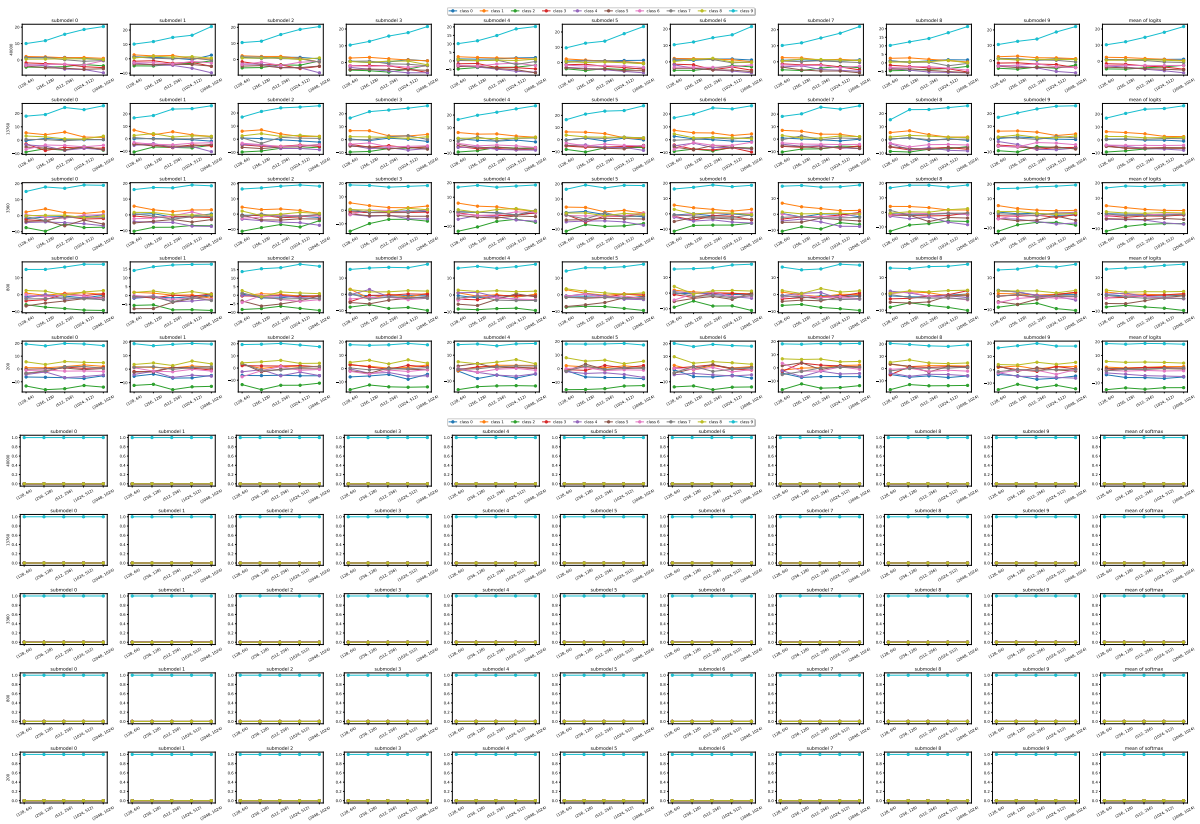


FIGURE 4.8: Predictions by DE for Figure 4.6c (the true label is 9). Logits on the top and Softmax on the bottom. See Figure C.8 for a higher-resolution version.

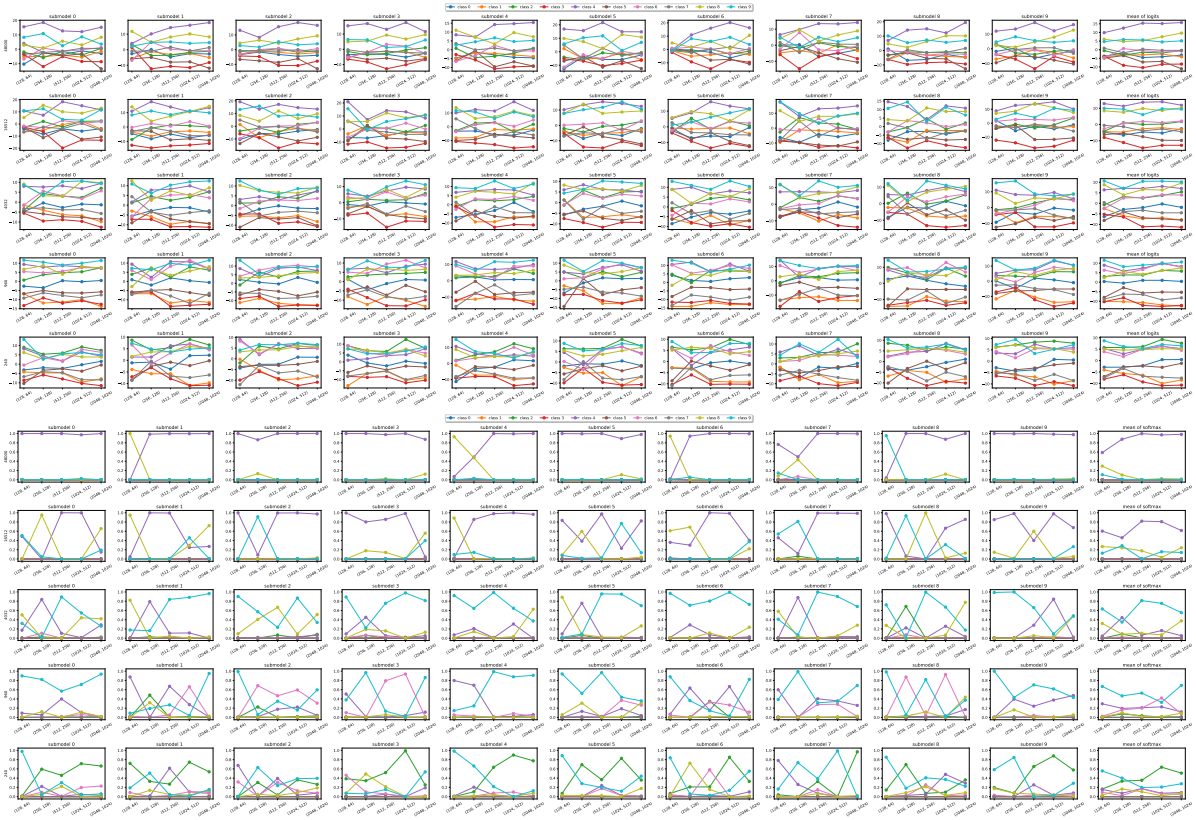


FIGURE 4.9: Predictions by MC-Dropout for Figure 4.6b (the true label is 4). Logits on the top and Softmax on the bottom. See Figure C.3 for a higher-resolution version.

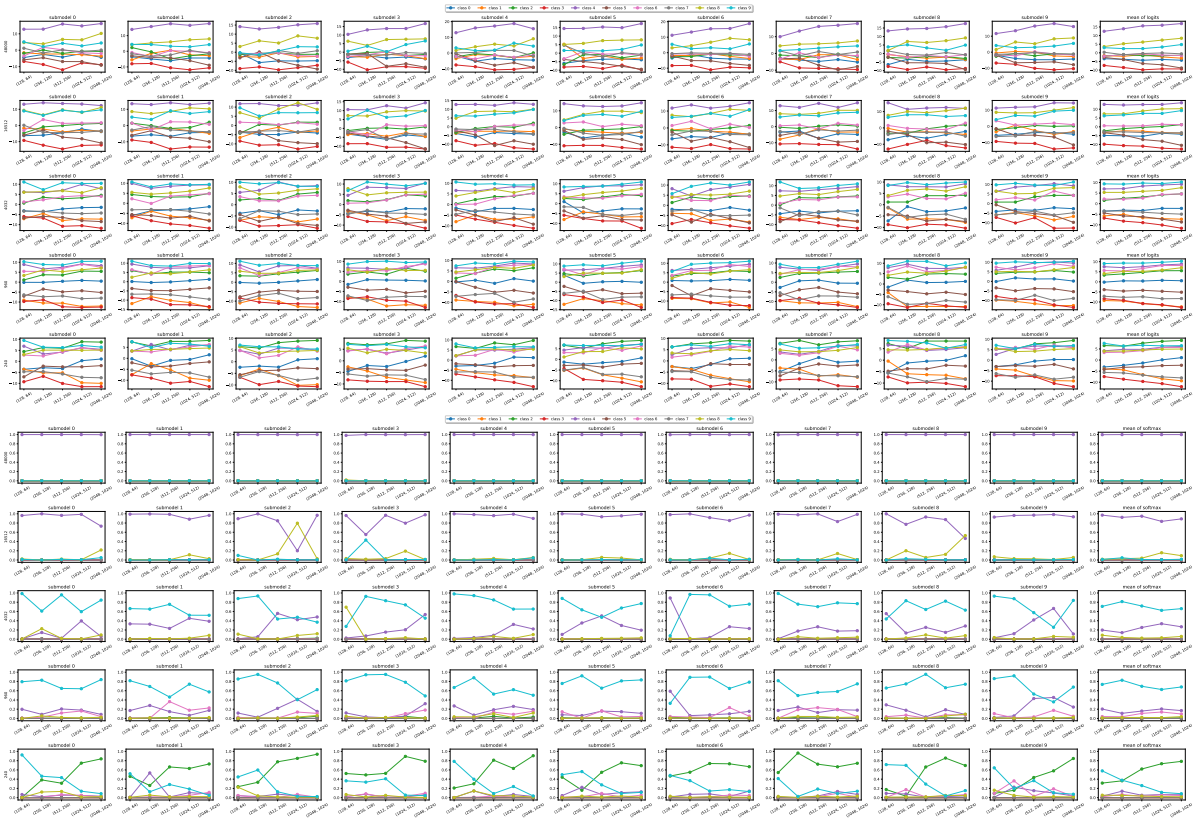


FIGURE 4.10: Predictions by DE for Figure 4.6b (the true label is 4). Logits on the top and Softmax on the bottom. See Figure C.7 for a higher-resolution version.

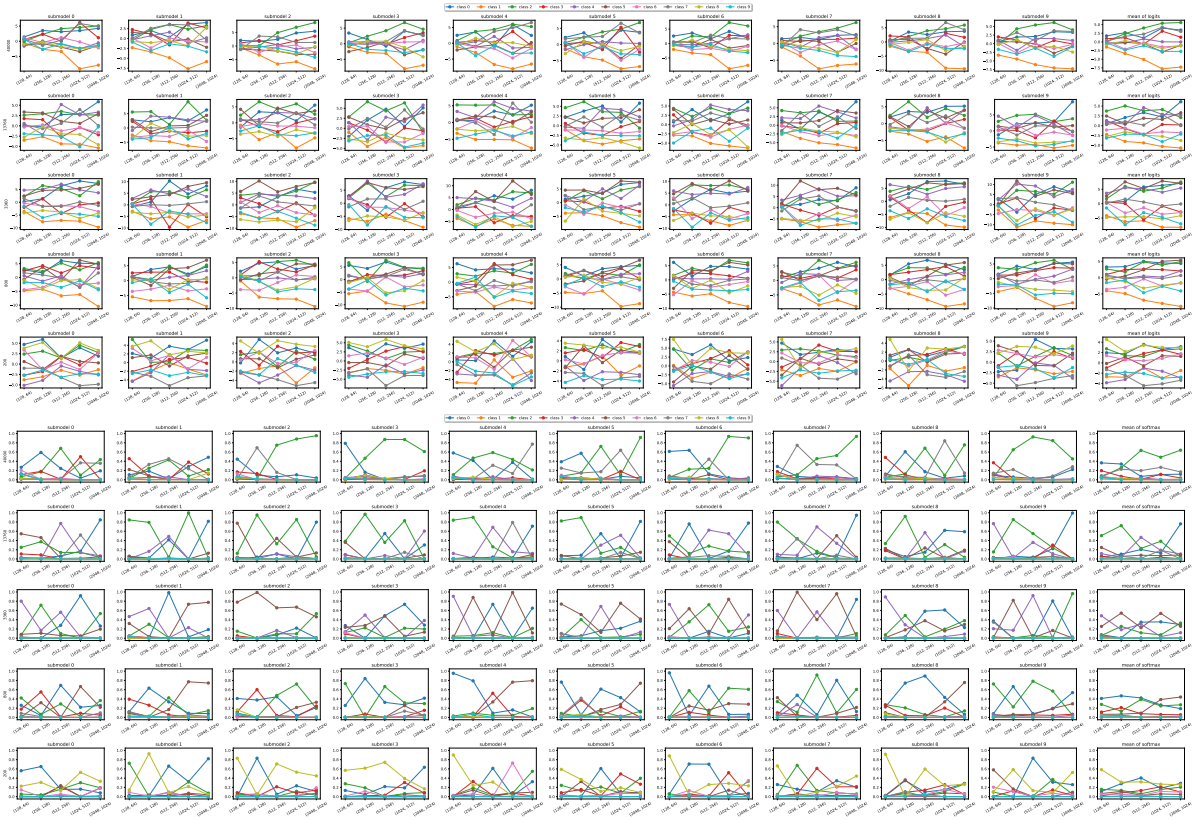


FIGURE 4.11: Predictions by MC-Dropout for Figure 4.6d (the true label is 7). Logits on the top and Softmax on the bottom. See Figure C.5 for a higher-resolution version.

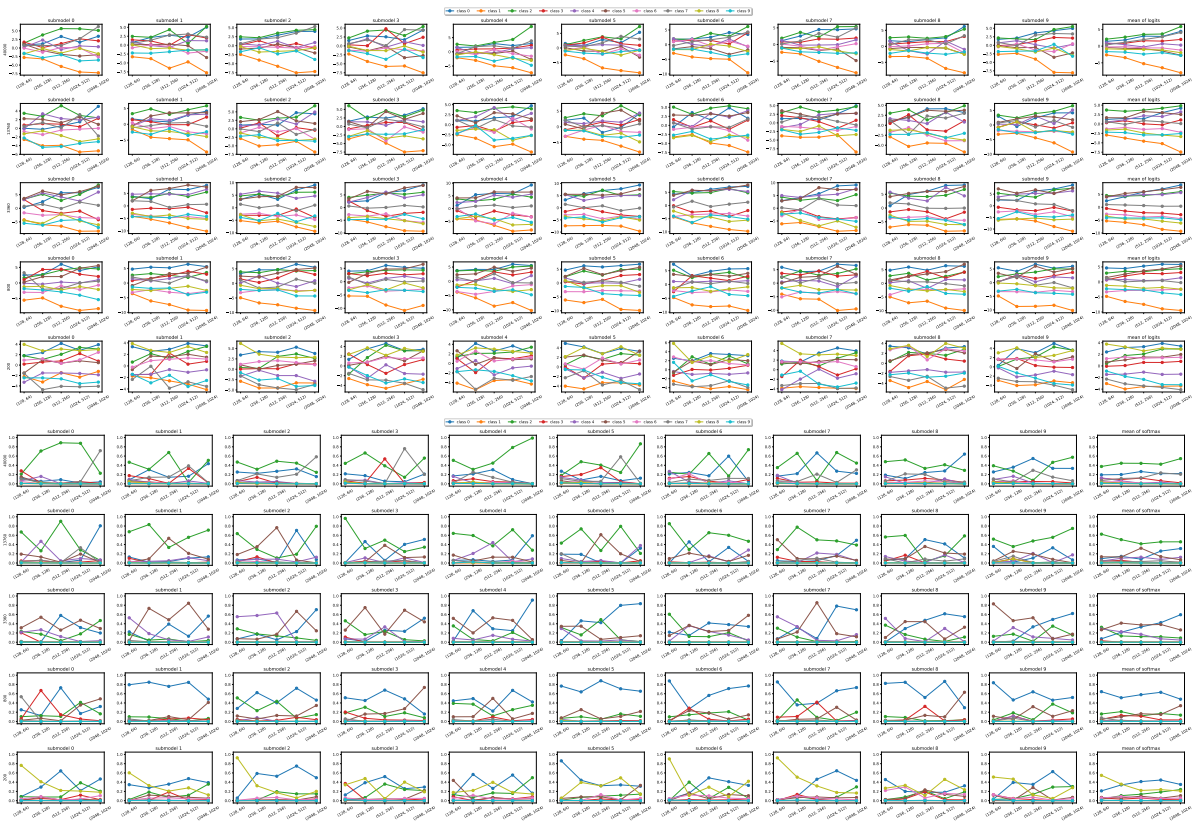


FIGURE 4.12: Predictions by DE for Figure 4.6d (the true label is 7). Logits on the top and Softmax on the bottom. See Figure C.9 for a higher-resolution version.

that these examples illustrate the incapacity of the model to describe its uncertainty about a specific prediction, especially from a total perspective for "hard" inputs.

It is interesting to see that by training on more datapoints, for both MC-Dropout and DE, the models will wrongly classify the "4" (Figure 4.6b) as "2", "8" and "9" before confidently and correctly classify the input. On the image in Figure 4.6d, the models were unable to correctly classify the horse with the top predicted classes being "bird", "airplane" and "dog".

In conclusion, one of the main motivations behind this analysis was to determine whether the uncertainty stays consistent or varies irregularly when modifying the size of the training set or the model, and also whether the patterns observed at the macro level hold true for specific inputs. The results show that the uncertainty remains stable, with the logits varying regularly. Therefore, we can confirm that, at least for the tested images, the insights drawn from the heatmaps based on the average of the test set are also valid for individual examples. Additionally, we emphasize that although the samples in Figure 4.6 are selected objectively, the discussion regarding the micro analysis focused on these samples could be different if we had analyzed different inputs. However, it gives a better intuition about the effect of the size of the training set and the model complexity, and we believe similar trends will appear on the test set.

4.5 Evaluating Aleatoric Uncertainty

So far, we mainly analyzed the evolution of epistemic uncertainty for different models, and how it changes in accordance with the two fundamental principles. We shift in this section to the calibration of aleatoric uncertainty. The goal is to track the effect of varying the inputs on aleatoric uncertainty. In practice, these changes will be applied through two strategies: rotating samples and evaluating ambiguous inputs. Importantly, despite the emphasis on aleatoric uncertainty in this section, epistemic uncertainty will also be evaluated under these strategies, as a significant rotation is likely to lead to greater epistemic uncertainty.

For simplicity, we will only consider a single configuration from the heatmap: we select the model with the hidden layers $(512, 256)$ and which is trained on the entire training set. In this case, the comparison with results from the literature will be straightforward as they generally only report their findings for models trained with all the available training samples. Additionally, we will primarily focus on MNIST as the two aforementioned strategies can be interpreted easily. Furthermore, four models will be tested: MC-Dropout, MC-Dropout with LS, EDL and DE.

4.5.1 Rotating samples

Inspired by Sensoy et al. (2018), we will explore the evolution of uncertainties when selected inputs are rotated with different angles from 0 to 180 degrees. Three critical classes will be tested:

- **1-to-1**: "1" for which a 180° rotation will result in a "1" (Figure 4.13).
- **6-to-9**: "6" for which a 180° rotation will lead to a different class "9" (Figure 4.14).
- **All-0**: "0" which is (pseudo-)invariant to any rotation (Figure 4.15).

In all the figures, each row represents a different model, with the last row showing the image and its rotations. The four columns are $\max\left(\mathrm{p}(Y \mid x, \mathcal{D})\right)$, *total uncertainty*, *aleatoric uncertainty* and *epistemic uncertainty*. In each plot, we will have the measure plotted in blue as a function of the degrees of rotations, and the orange curve is a binary mask representing whether the predictions match the target. Be aware that the scale of the y-axis varies across the subplots, so it is important to consider this when interpreting the data.

Across all figures, we observe that aleatoric uncertainty is generally higher than epistemic uncertainty, with the gap is more pronounced for MC-Dropout with LS and EDL. As the beginning and the end of rotations corresponds to actual class, the models are confident and have low uncertainties in these phases (Figures 4.13 and 4.14). However, we observe much higher uncertainties at the end phase for **All-0** (Figure 4.15). We suspect that this observation is sample-dependent and/or might be caused by the rotation algorithm[13]. A low uncertainty is usually near zero for MC-Dropout and DE, but around 1.5 for total uncertainty in the case of MC-Dropout with LS. Moreover, the uncertainty curves are relatively noisy and display a lack of smoothness and irregularities particularly at large rotation angles. At these rotations, the inputs can hardly be considered as digits, hence aleatoric uncertainty is perhaps not the ideal uncertainty to look at (more on this in Section 6.1).

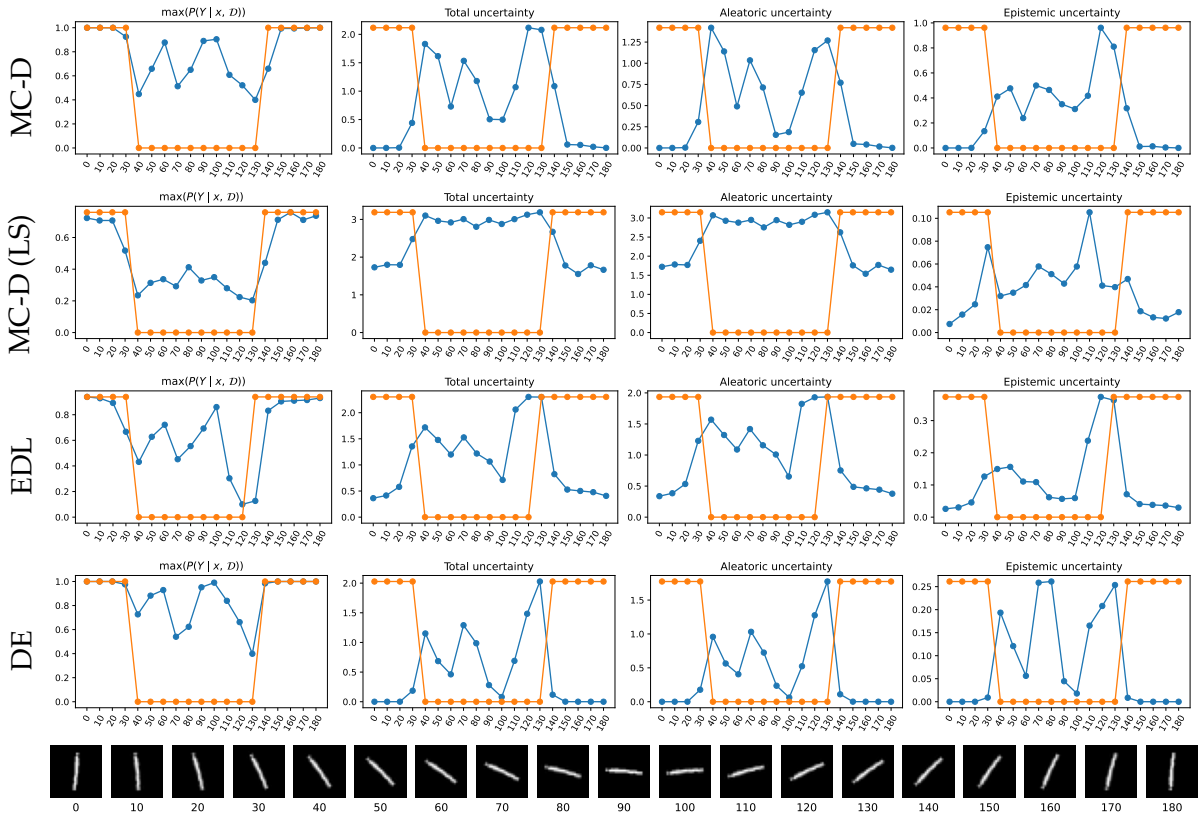


FIGURE 4.13: **1-to-1**: Rotating the sample of index 176 from the test set of MNIST. The orange curve indicates whether the predictions match the target.

Regarding **1-to-1**, the digit "1" is correctly classified by all the models outside the range of rotations from 40° to 130°. The only exception is EDL at a rotation of 130°. Yet, this prediction is correct with a low confidence. Unfortunately, we do not end up with the same representation as in Sensoy et al. (2018, Figure 1) when rotating the digit "1" (Figure 4.13). After careful investigation, this can be attributed to two primary factors. First, the reported uncertainty is measured in the Dempster-Shafer Theory of Evidence (DST) framework, computed through the belief masses. Second, and perhaps most importantly, by looking at the provided code[14], the rotated digit "1" used is from the training set and not from the test set, thus unreliable for generalization. We argue that using a digit from the test set will produce similar curves as reported in Figure 4.13. The curves of **6-to-9** (Figure 4.14) are somehow similar to those of **1-to-1** (Figure 4.13). The only major

[13] https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.rotate.html with `mode='nearest'`

[14] https://muratsensoy.github.io/uncertainty.html

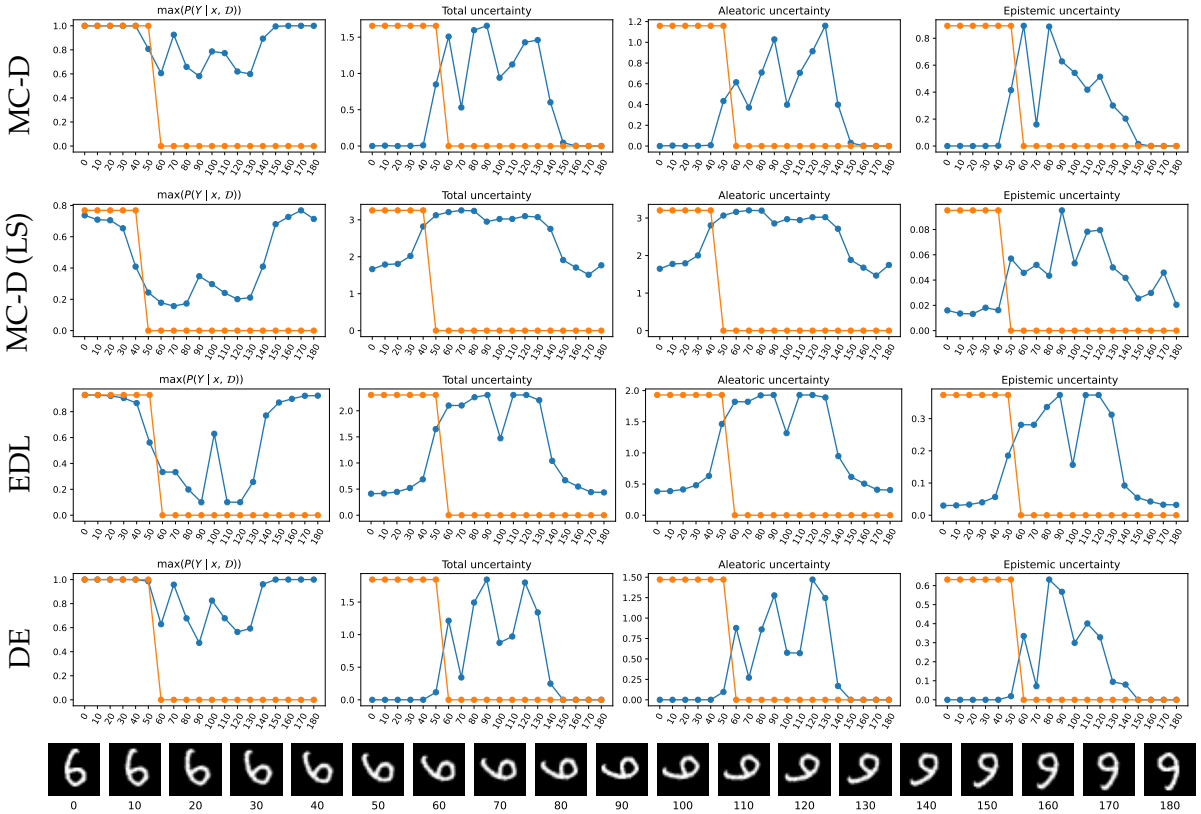


FIGURE 4.14: **6-to-9**: Rotating the sample of index 21 from the test set of MNIST. The orange curve indicates whether the predictions match the target.

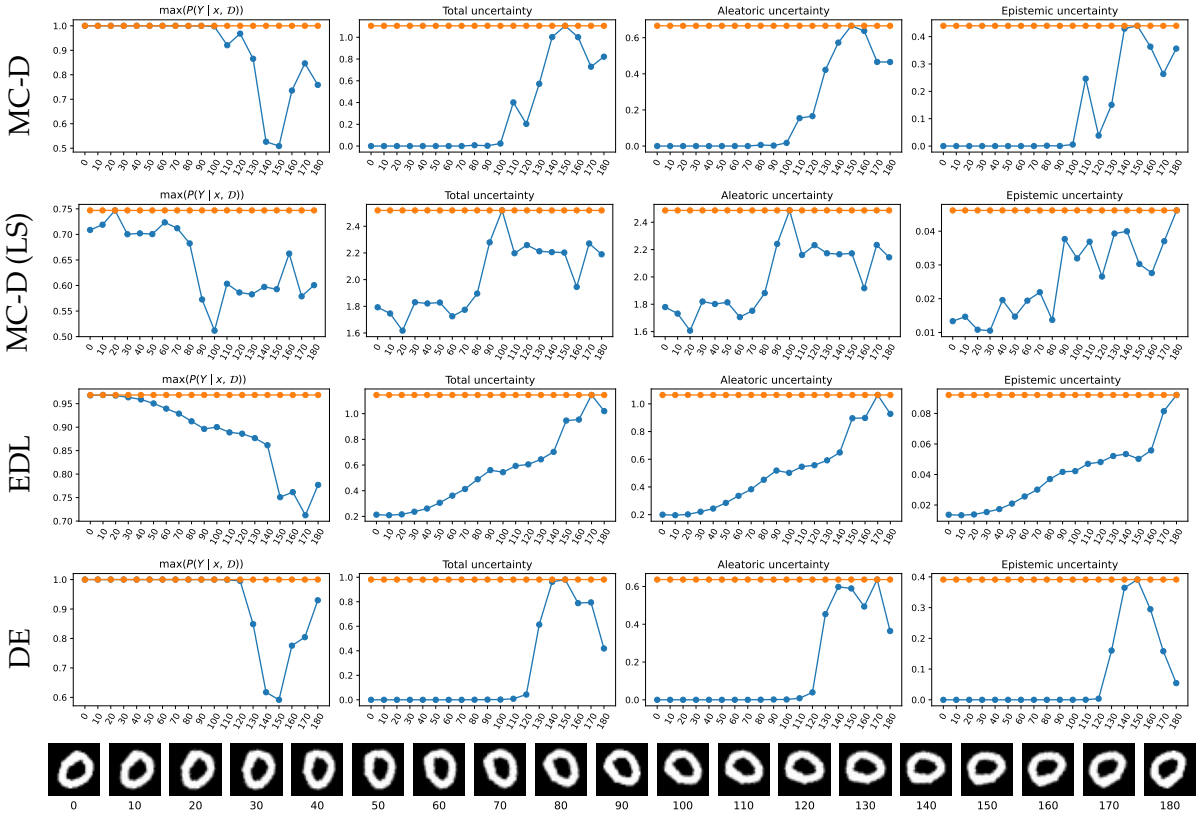


FIGURE 4.15: **All-0**: Rotating the sample of index 6114 from the test set of MNIST. The orange curve indicates whether the predictions match the target.

exception is the binary mask which does not show the predictions being correctly "9" for large rotations. Finally, all the rotations in **All-0** are correctly classified as "0" (Figure 4.15).

On these digits, while the case of **All-0** and the small rotations on **1-to-1** and **6-to-9** are expected to be aleatorically uncertain, the same cannot be said about the remaining cases for which the disentanglement is not ideal as aleatoric uncertainty absorbs the epistemic component. For the tested models, the decomposition of uncertainties is more noticeable in the case of MC-Dropout LS and EDL, and it is the least severe for MC-Dropout.

4.5.2 Ambiguous-MNIST

Another aspect to evaluate aleatoric uncertainty is by using ambiguous samples. Mukhoti and Kirsch (2023) introduced *Ambiguous-MNIST* which includes images with several possible labels and thus higher aleatoric uncertainty compared to MNIST images. We selected visually some images from Ambiguous-MNIST and report the result in Figure 4.16. These samples are selected in order to be the most unrecognizable by humans. The visualization is similar to the previous section with the only difference being the selected samples on the x-axis instead of the rotated images.

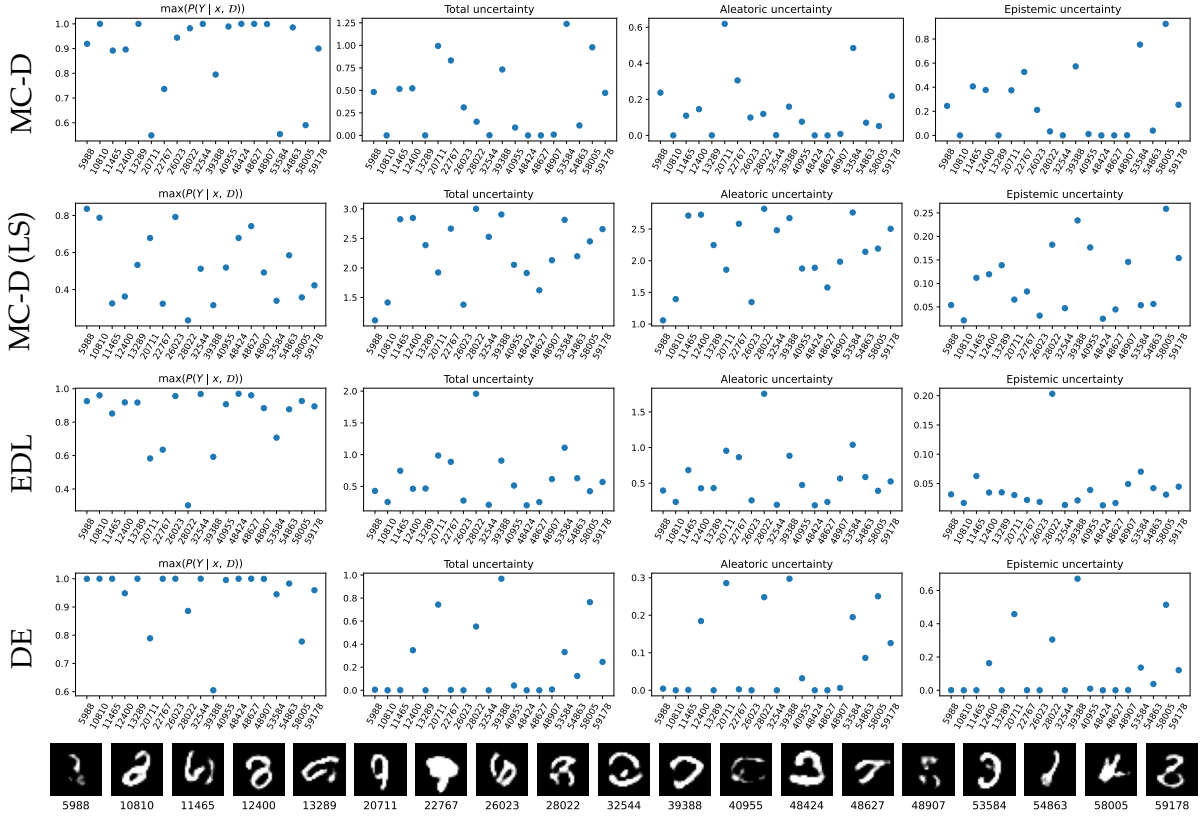


FIGURE 4.16: Results on inputs from Ambiguous-MNIST.
The integers under the images represent their indices in the dataset.

Aleatoric uncertainty is significantly higher than epistemic uncertainty for both EDL and MC-Dropout with LS, with its values comparable to those of the rotated images in the middle phase (rotations around 90°). Surprisingly, many inputs have zero total uncertainty with MC-Dropout and DE, implying that all the models in the DE and the sampled models in MC-Dropout are predicting confidently the same class. In addition to that, the maximum probability is highly confident for remaining class of inputs with their source of uncertainty being mostly epistemic. Whether the selected inputs are closer to ambiguous or out-of-distribution examples is something that will be discussed further in this thesis (Section 6.1.6).

4.6 Conclusions

Given the defined principles of epistemic uncertainty, we showed the lack of epistemic calibration for the commonly used Bayesian models, with the model-related principle being almost never verified. While the accuracy of the models appears consistent and with no unexpected evolution, the aforementioned observations and discussions raise serious questions on the awareness of epistemic uncertainty for these models. In the next chapter, we will conduct a more in-depth analysis to identify the potential root cause of the epistemic miscalibration of DE. Additionally, a proposed solution will be tested using the same experimental protocol outlined in this chapter.

CHAPTER 5
UNINFORMATIVE PRIORS: BEYOND UNIFORMITY

"Man is essentially ignorant, and becomes learned through acquiring knowledge."
Ibn Khaldun

The measured epistemic uncertainty should be calibrated in order to assess the model uncertainty. Unfortunately, this is not the case for the commonly tested BNNs as they violate the data-related principle, and they almost never verify the model-related principle. As discussed in Chapter 3, the choice of the prior is crucial, especially from an epistemic perspective, since it is directly related to the model parameters. Additionally, given the numerous options available for the prior distribution, the choice appears challenging. As a result, although it may be suboptimal, considering an uninformative prior could be advantageous. Furthermore, we observed in Chapter 4 how the different models lack diversity in the outputs, which could explain partially the hole of epistemic uncertainty. In light of the above, we propose solutions to the epistemic uncertainty hole based on objective priors.

5.1 Motivations and Intuitions behind Conflictual Loss

Our solutions are grounded in established classical results in the field of non-informative priors, while ensuring they lead to diversity in the outputs. In this section, we provide motivations and intuitions behind our approach before developing in the next section a detailed theoretical derivation leading to this solution, while highlighting at the same time the underlying approximations.

5.1.1 The Need of Boosting Output Diversity

The analyses in the previous chapter highlighted the lack of epistemic uncertainty for the commonly used BNNs under the two fundamental principles. On both the aggregated results (Section 4.3.2) and the histograms (Section 4.4), we can see that epistemic uncertainty has low values, regardless of the size of the training set and the model complexity. More precisely, as the model-related principle is almost never verified, it indicates a collapse of epistemic uncertainty for large models. This observation is a direct manifestation of the lack of diversity in the predictions, especially given that mutual information is a measure of variability or conflict (Wimmer et al., 2023). Yet, diverse predictions are expected when the model fails at confidently predicting the output, as the model expresses its ambiguity through a wider range of possible outcomes.

Arguably, the models tend to provide similar predictions, even in the case of a misclassified sample or an input with high epistemic uncertainty, as illustrated in Section 4.4.2. For example, in the case of the DE, although the (sub)models were trained independently, they have resembling predictions, indicating similarities in "explaining" the inputs. While deep ensembles were introduced to offer a reliable predictive uncertainty (Section 2.2), the same cannot be said about epistemic uncertainty, as shown empirically in the previous chapter. To this end, we will explore deep ensembles and aim at favoring diversity in the outputs, hence consequently having a reliable epistemic uncertainty for these models.

In order to achieve this goal, our approach consists of artificially boosting diversity in the ensemble by making each model in the ensemble specialized in a specific class. By doing so, each model in the ensemble is favoring a unique class, especially when the model cannot reliability predict the correct class. The models within the deep ensemble are trained independently, each with a different optimization objective, hence promoting diversity among the ensemble members. Accordingly, one can reasonably assume that the ensemble exhibits a meaningful degree of model diversity. From a practical perspective, this specialization is achieved in the form of a regularization term of the associated class in the output space.

In the coming sections, we will formalize this diversity-promoting loss function, and we will refer to it as *Conflictual loss* (Fellaji et al., 2024). Being a stronger form of diversity, a conflict between the predictions is achieved especially when the model cannot provide reliable predictions, leading to a high epistemic uncertainty.

Remark *(Specialization and Credal sets).* If we look at this specialization from the lens of Credal sets, although the formalism is different, it can be seen as if each element of the set is trained with a different prior (in the output space), leading to a convex set that covers more surface in the simplex.

5.1.2 The Conflictual Loss as a Heuristic

To create conflict in the ensemble, the idea is to regularize the model in the output space by favoring a single class per each member of the ensemble. This regularization is to be added to the cross-entropy loss. In the simplest case, we assume the deep ensemble has C models (*i.e.* the same number of models as the number of classes). The Conflictual loss for model $i \in \{1, \ldots, C\}$[15] parameterized by $\boldsymbol{\theta}_i$ can be defined as:

$$\mathcal{L}_{\text{CL}}^{(i)}(\boldsymbol{\theta}_i, \mathcal{D}) = -\left(\sum_{(x,y)\in\mathcal{D}} \log(\mathrm{p}(y \mid x, \boldsymbol{\theta}_i)) + \lambda \log(\mathrm{p}(i \mid x, \boldsymbol{\theta}_i)) \right) \tag{5.1}$$

With $\lambda \geq 0$ a hyperparameter for the Conflictual loss.

For an input-output tuple (x, y) from the training dataset $\mathcal{D}$, the loss function consists of two main parts. The first component is the cross-entropy loss which represents the "data" term, as commonly computed in the classification tasks. The second part represents the regularization term and incorporates the specialization of the model i. While it appears as if this model will be biased toward the class i, the optimization objective aims at learning to predict the correct class y in addition to the class i (if the two are different), with a confidence controlled by the hyperparameter λ. Consequently, a *Conflictual DE* will refer to such ensemble of C models where each model is specialized in a different class. Furthermore, as the predicted class depends on the BMA of the Conflictual DE, the specialization per model will be averaged out.

[15] In the experimental results, the indices range from 0 to $C-1$ for both the classes and the submodels.

When analyzing Equation 5.1, link can be made with label smoothing (Szegedy et al., 2015), with the only difference being that, in the former, only a single class is considered whereas in the latter the regularization takes into account all the classes. Under the formalism detailed in Meister et al. (2020), these regularizations can be expressed as the Kullback-Leibler divergence between the model prediction (p_θ) and some baseline distribution ($\mathcal{Q}$): $D_{KL}(\mathcal{Q} \,\|\, p_\theta)$. In label smoothing, the baseline distribution is the discrete uniform distribution, but for Equation 5.1, it can be set as the discrete Dirac distribution centered at class i. Therefore, while label smoothing forces the model toward a higher predictive uncertainty solution, Conflictual loss forces the Conflictual DE toward a higher, and potentially calibrated, epistemic uncertainty solution.

5.2 Formal Derivation of Conflictual Loss and Approximations

This section outlines the logical progression leading to the definition of the Conflictual loss as defined in Section 5.1.2. Each subsection represents a step in the reasoning.

5.2.1 Transferring Prior on the Output

Our objective is to encourage the model to align with the data-related principle, namely that epistemic uncertainty decreases as the training set size increases. In the following, we choose to measure epistemic uncertainty using the mutual information between the model's output and its parameters, and we focus on classification problems from an input space $\mathcal{X}$ to a finite set of classes $\mathcal{Y} = \{1, \ldots, C\}$. At this point, no particular assumption is made regarding the structure of the parametric classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$, with $\boldsymbol{\theta}$ denoting the parameter vector.

Reformulation of the ELBO

First, observe that epistemic uncertainty for a given example x depends solely on the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$. Furthermore, the first principle requires that epistemic uncertainty be maximal in the absence of training data, in which case the posterior reduces to the prior. Therefore, based on these observations, ensuring the data-related principle, and by extension, achieving maximum epistemic uncertainty in the absence of data, is only possible through an appropriate choice of the prior distribution $p(\boldsymbol{\theta})$. Therefore, it is important to choose a suitable prior. Yet, setting a prior in the parameter space is challenging (as detailed in Section 3.2), making this choice in the output space (consisting of the categorical distributions $f_{\boldsymbol{\Theta}}(x)$) easier and more interpretable.

Formally, let's consider the stochastic process $\boldsymbol{\Pi} : x \in \mathcal{X} \mapsto f_{\boldsymbol{\Theta}}(x)$ mapping a given input x to the categorical distribution $\boldsymbol{\Pi}(x) = f_{\boldsymbol{\Theta}}(x)$, which can be seen as a random variable. Since the model f_θ is fixed, the distribution of $\boldsymbol{\Pi}(x)$ is entirely defined through the prior $p(\boldsymbol{\theta})$ of $\boldsymbol{\Theta}$. Hence, the desired ideal prior $p^*(\boldsymbol{\theta})$ induces an ideal prior for the outputs $p^*(\boldsymbol{\pi}(x))$. Moreover, the maximization of epistemic uncertainty for the input x (Equation 2.19) is an optimization problem that does not depend anymore on $p^*(\boldsymbol{\theta})$ if given $p^*(\boldsymbol{\pi}(x))$, since

$$
\begin{aligned}
\mathcal{U}^e(x) &= \mathbb{I}(\boldsymbol{\Theta}; Y \mid x) \\
&= \mathbb{E}_{Y,\boldsymbol{\Theta}}\left[\log\left(\frac{p(Y, \boldsymbol{\Theta} \mid x)}{p(Y \mid x)\, p(\boldsymbol{\Theta})}\right)\right] \\
&= \mathbb{E}_Y\left[\mathbb{E}_{\boldsymbol{\Theta}}\left[\log\left(\frac{p(Y \mid \boldsymbol{\Theta}, x)}{p(Y \mid x)}\right)\right] \,\middle|\, Y\right] \\
&= \mathbb{E}_Y\left[\mathbb{E}_{\boldsymbol{\Pi}(x)}\left[\log\left(\frac{p(Y \mid \boldsymbol{\Pi}(x))}{p(Y \mid x)}\right)\right] \,\middle|\, Y\right] \\
&= \mathbb{I}(\boldsymbol{\Pi}(x); Y)
\end{aligned}
$$

Therefore finding the ideal prior can be solved in a two steps process, in the following order:

1. Determine the prior $\mathrm{p}^*(\boldsymbol{\pi})$ that maximize epistemic uncertainty regardless of the input x:

$$\mathrm{p}^*(\boldsymbol{\pi}) \in \underset{\mathrm{p}(\boldsymbol{\pi})}{\operatorname{argmax}} \left(\mathbb{I}(\boldsymbol{\Pi}; Y)\right)$$

2. Deduce a prior $\mathrm{p}^*(\boldsymbol{\theta})$ over the parameters that induces the prior $\mathrm{p}^*(\boldsymbol{\pi}(x))$ at the output for all inputs x.

Let's focus for now on the second step, as the first will be discussed shortly. Since the prior on the output $\mathrm{p}(\boldsymbol{\pi}(x))$, induced by the prior $\mathrm{p}(\boldsymbol{\theta})$, depends on the input x, the goal of the second step is to minimize, in expectation with respect to $\boldsymbol{X}$, the divergence between $\mathrm{p}(\boldsymbol{\pi}(\boldsymbol{X}))$ and $\mathrm{p}^*(\boldsymbol{\pi})$. Moreover, the prior distribution $\mathrm{p}(\boldsymbol{\theta})$ is exclusively relevant to compute the posterior $\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})$, which is approximated thanks to variational inference by optimizing for the variational distribution $q^*(\boldsymbol{\theta}) \approx \mathrm{p}^*(\boldsymbol{\theta} \mid \mathcal{D})$ that maximize the ELBO (Equation 2.9):

$$\begin{aligned}\mathrm{ELBO}(q) &= \sum_{(x,y)\in\mathcal{D}} \mathbb{E}_{q(\boldsymbol{\theta})}[\log\left(\mathrm{p}(y \mid x, \boldsymbol{\theta})\right)] - \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{\theta}) \,\|\, \mathrm{p}^*(\boldsymbol{\theta})) \\ &= \sum_{(x,y)\in\mathcal{D}} \mathbb{E}_{q(\boldsymbol{\pi}(x))}[\log\left(\mathrm{p}(y \mid \boldsymbol{\pi}(x))\right)] - \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{\theta}) \,\|\, \mathrm{p}^*(\boldsymbol{\theta}))\end{aligned}$$

Where $q(\boldsymbol{\pi}(x))$ is the distribution of $\boldsymbol{\pi}(x)$ resulting from $q(\boldsymbol{\theta})$. We see that the prior appears only in the KL divergence, making it possible to express this term as an average divergence on the model's output, *i.e.* between $q(\boldsymbol{\pi}(x))$ and $\mathrm{p}^*(\boldsymbol{\pi})$. Additionally, since the prior is chosen independently of the input, $\mathrm{p}^*(\boldsymbol{\pi})$ does not depend on x. These considerations allow the ELBO to be expressed uniquely in terms of the outputs:

$$\mathrm{ELBO}(q) = \sum_{(x,y)\in\mathcal{D}} \mathbb{E}_{q(\boldsymbol{\pi}(x))}[\log\left(\mathrm{p}(y \mid \boldsymbol{\pi}(x))\right)] - \mathbb{E}_{\boldsymbol{X}}[\mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{\pi}(\boldsymbol{X})) \,\|\, \mathrm{p}^*(\boldsymbol{\pi}))] \tag{5.2}$$

Monte Carlo approximation of the prior

One practical obstacle is that the theoretical expectation over $\boldsymbol{X}$ cannot be computed directly. A viable workaround is to approximate it with an MC estimate (Proposition 2.1) on the training data: with a sufficiently large dataset, we can replace the expectation with an empirical average.

$$\begin{aligned}\mathrm{ELBO}(q) &\approx \sum_{(x,y)\in\mathcal{D}} \mathbb{E}_{q(\boldsymbol{\pi}(x))}[\log\left(\mathrm{p}(y \mid \boldsymbol{\pi}(x))\right)] - \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}} \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{\pi}(x)) \,\|\, \mathrm{p}^*(\boldsymbol{\pi})) \\ &= \sum_{(x,y)\in\mathcal{D}} \left(\mathbb{E}_{q(\boldsymbol{\pi}(x))}[\log\left(\mathrm{p}(y \mid \boldsymbol{\pi}(x))\right)] - \frac{1}{|\mathcal{D}|}\mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{\pi}(x)) \,\|\, \mathrm{p}^*(\boldsymbol{\pi}))\right)\end{aligned}$$

In this form, the ELBO can be easily implemented and optimized by using stochastic gradient ascent for example. Still, the use of Monte Carlo estimate remains questionable. First, the data-related principle is most relevant in low-data regimes, where the limited sample size $|\mathcal{D}|$ may lead to approximation errors due to the law of large numbers not yet taking full effect. More fundamentally, using the expectation on X raises questions since the prior $\mathrm{p}^*(\boldsymbol{\pi}(x))$ should be the same for all inputs x, including samples for which $\mathrm{p}(x) = 0$. To mitigate this issue, the Monte Carlo average can be enriched with additional and various (unlabeled) datapoints $\mathcal{D}'$:

$$\mathrm{ELBO}(q) \approx \sum_{(x,y)\in\mathcal{D}} \mathbb{E}_{q(\boldsymbol{\pi}(x))}[\log\left(\mathrm{p}(y \mid \boldsymbol{\pi}(x))\right)] - \frac{1}{|\mathcal{D}| + |\mathcal{D}'|}\sum_{x\in\mathcal{D}\cup\mathcal{D}'} \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{\pi}(x)) \,\|\, \mathrm{p}^*(\boldsymbol{\pi})) \tag{5.3}$$

However, we did not use this improvement in the experiments we conducted, in order to keep our evaluation protocol as simple as possible.

5.2.2 Conflictual Output Prior

The next step is to select the prior distribution $p^*(\boldsymbol{\pi})$, with a support in the simplex $\boldsymbol{\Delta}^{C-1}$, over the model's categorical output $\boldsymbol{\pi}$, with the primary goal of maximizing epistemic uncertainty. If no additional constraints are imposed, the solution to this optimization problem is well known in the literature[16] : the only distribution that yields maximal epistemic uncertainty (and zero aleatoric uncertainty) is the mixture of Dirac distributions located at the vertices of the simplex.

$$\forall \boldsymbol{\pi} \in \boldsymbol{\Delta}^{C-1}, \quad p^*(\boldsymbol{\pi}) = \frac{1}{C}\sum_{i=1}^{C} \delta_i(\boldsymbol{\pi}) \qquad \text{with } \delta_i^{(j)} = \delta(i, j) \tag{5.4}$$

For which mutual information attains its maximal, as each Dirac component in the prior assigns a probability mass of 1 to a unique class and 0 to all the others, resulting in mutually contradictory predictions:

$$\mathbb{I}(\boldsymbol{\Theta}; Y) = \mathbb{H}(Y) - \mathbb{H}(Y \mid \boldsymbol{\Theta}) = \log(C) - 0$$

Although this choice is optimal, it is nevertheless too extreme to be practically acceptable. Indeed, Bayes' rule implies that the support of the posterior must be a subset of the support of the prior. Since the support of $p^*(\boldsymbol{\pi})$ is restricted to the extreme points of the simplex, the same will hold true for any resulting posterior. Bayesian inference thus becomes infeasible, as it can never converge to the optimal output distribution[17] $\boldsymbol{\pi}^*$ whenever it lies in the interior of the simplex (which is typically the case). To overcome this limitation, a prior $p^*(\boldsymbol{\pi})$ must satisfy several criteria (Definition 5.1).

Definition 5.1: Necessary criteria for an ideal prior

A Conflictual output prior should meet the following criteria:

Class immiscibility: Most of the probability mass must be concentrated on the vertices of the simplex $\boldsymbol{\Delta}^{C-1}$ in order to convey large epistemic uncertainty. In particular the prior should have the same modes as Equation 5.4, *i.e.* the C vertices $\boldsymbol{\delta}_i$ of the simplex.

Positivity: The prior density must be non-zero across the entire simplex to allow convergence to $\boldsymbol{\pi}^*(x)$, regardless of its value.

Symmetry: The prior must be invariant under class permutations, since the absence of specific knowledge about the classes implies complete symmetry among them.

Smoothness: The prior must be a differentiable function over the simplex $\boldsymbol{\Delta}^{C-1}$ to allow for gradient-based optimization.

One way to satisfy both the *class immiscibility* and the *symmetry* criteria is by using a mixture of identical distributions with each having a unique mode at a different class:

$$p^*(\boldsymbol{\pi}) = \frac{1}{C}\sum_{i=1}^{C} p_i^*(\boldsymbol{\pi})$$

[16] See for instance (Gal, 2016, page 54).

[17] It corresponds to the maximum likelihood distribution $\boldsymbol{\pi}^*(x) = \lim_{|\mathcal{D}|\to+\infty} \boldsymbol{\pi}_{MLE}(x)$ in the limit of infinite data.

The most natural distribution that satisfies all these constraints, in addition to the *positivity* and the *smoothness* criteria, is undoubtedly the Dirichlet distribution:

$$\mathrm{p}_i^*(\boldsymbol{\pi}) = \mathtt{Dir}(\mathbf{1}_C + \alpha \cdot \boldsymbol{\delta}_i) = \frac{\Gamma(\alpha + C)}{\Gamma(\alpha + 1)} \pi_i^\alpha$$

Leading to the prior that we will denote p_α^*:

$$\mathrm{p}_\alpha^*(\boldsymbol{\pi}) = \frac{\Gamma(\alpha + C)}{C\,\Gamma(\alpha + 1)} \sum_{i=1}^{C} \pi_i^\alpha \tag{5.5}$$

The parameter $\alpha > 0$ controls the similarity to the Dirac mixture (Equation 5.4). When $\alpha = 1$, the distribution is uniform over the simplex, resulting in relatively low epistemic uncertainty and higher aleatoric uncertainty. As α increases, the mode densities rise, the probability mass in the center diminishes, epistemic uncertainty increases, and aleatoric uncertainty decreases. In the limit as α tends to infinity, the distribution converges to the Dirac mixture. The two first rows of Figure 5.1 respectively illustrate the cases $\alpha = 5$ and $\alpha = 1$.

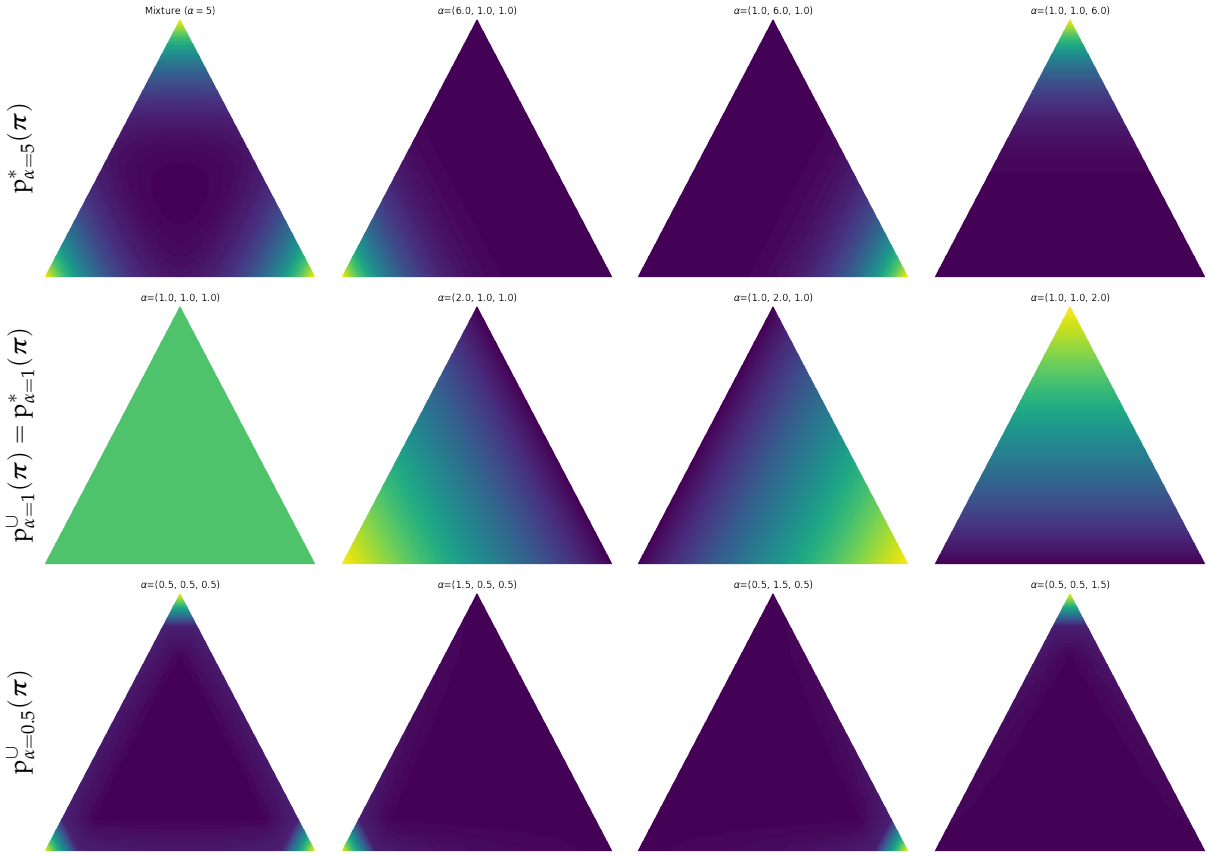


FIGURE 5.1: Illustrative examples of $\mathrm{p}_\alpha^*(\boldsymbol{\pi})$ and $\mathrm{p}_\alpha^\cup(\boldsymbol{\pi})$ per row, with $C = 3$. The first column corresponds to the mixture, whereas the components of the mixture are in the remaining three columns.

Another possibility to produce a prior with similar shape, is the following parameterization, also based on a mixture of Dirichlet distributions, this time with $0 < \alpha \leq 1$:

$$\mathrm{p}_\alpha^\cup(\boldsymbol{\pi}) = \frac{1}{C} \sum_{i=1}^{C} \mathtt{Dir}(\alpha \cdot \mathbf{1}_C + \boldsymbol{\delta}_i) \tag{5.6}$$

The constraint $0 < \alpha \leq 1$ is required to ensure the *class immiscibility* criterion (Definition 5.1). The case $\alpha = 0.5$ is illustrated on the last row of Figure 5.1. Under this assumption, for a single component i, not only the class i is favoured, but all the remaining classes are discouraged by the prior. This encouraging-discouraging behavior is unique to $\mathrm{p}^{\cup}_{\alpha}(\boldsymbol{\pi})$ with $0 < \alpha \leq 1$, while only the encouraging behavior for class i is achieved with $\mathrm{p}^{*}_{\alpha}(\boldsymbol{\pi})$ under $\alpha > 0$. Importantly, $\mathrm{p}^{*}_{\alpha}(\boldsymbol{\pi})$ and $\mathrm{p}^{*}_{\cup}(\boldsymbol{\pi})$ are identical in the special case of $\alpha = 1$, which corresponds to the uniform prior.

Additionally, by applying the Dirichlet decomposition (Sethuraman, 1994, Lemma 3.2), we can see that the mixture $\mathrm{p}^{\cup}_{\alpha}(\boldsymbol{\pi})$ is the symmetric Dirichlet distribution:

$$\mathrm{p}^{\cup}_{\alpha}(\boldsymbol{\pi}) = \mathtt{Dir}(\alpha \cdot \mathbf{1}_C)$$

Throughout the rest of this section, we focus on $\mathrm{p}^{*}_{\alpha}(\boldsymbol{\pi})$ for $\alpha > 0$, and we will discuss at the end the case of $\mathrm{p}^{\cup}_{\alpha}(\boldsymbol{\pi})$.

5.2.3 Conflictual Deep Ensemble

The optimization of the ELBO in the case of the prior p^{*}_{α} is non-trivial due to the presence of multiple modes, which have a significant influence in the critical regime of limited data.

In the absence of data

To understand this, consider the extreme case where no data is observed. The posterior then equals the prior, and the ELBO (Equation 5.3) reduces to the KL divergence term on $\mathcal{D}'$:

$$\mathrm{ELBO}(q) \approx -\frac{1}{|\mathcal{D}'|} \sum_{x \in \mathcal{D}'} \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{\pi}(x)) \,\|\, \mathrm{p}^{*}_{\alpha}(\boldsymbol{\pi})) \tag{5.7}$$

In this case, the optimization of the ELBO should lead to a proxy $q^{*}(\boldsymbol{\theta})$ that is close to $\mathrm{p}^{*}_{\alpha}(\boldsymbol{\pi})$, regardless of the input x. In order to achieve a maximal epistemic uncertainty, this approximation is of interest only if $q^{*}(\boldsymbol{\theta})$ captures all the modes of the simplex equivalently.

Since the prior is a mixture distribution, a natural way to make the optimization of the ELBO easier is to use an ensemble of C models $\{f_{\boldsymbol{\theta}_i}\}_{1 \leq i \leq C}$, with the goal of ensuring that each model $f_{\boldsymbol{\theta}_i}$ aligns its output with one of the modes of p^{*}_{α}:

$$\forall (i,\, x),\ f_{\boldsymbol{\theta}_i}(x) \approx \boldsymbol{\delta}_i$$

Let us take the analysis a step further by examining the consequences of introducing an ensemble on the expression of the ELBO. In fact, an ensemble can be seen as a Dirac mixture over the parameters:

$$q(\boldsymbol{\theta}) = \frac{1}{C} \sum_i \delta_{\boldsymbol{\theta}_i}(\boldsymbol{\theta})$$

Consequently, this mixture over parameters gives rise to a mixture over the outputs:

$$\forall x,\ q(\boldsymbol{\pi}(x)) = \frac{1}{C} \sum_i \delta_{\boldsymbol{\pi}_i(x)}(\boldsymbol{\pi}(x))$$

Focusing again on Equation 5.7, we can rewrite the KL term thanks to the mixture over the outputs:

$$\begin{aligned}-\mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{\pi}(x)) \,\|\, \mathrm{p}^*_\alpha(\boldsymbol{\pi})) &= \mathbb{H}(q(\boldsymbol{\pi}(x))) + \mathbb{E}_{q(\boldsymbol{\pi}(x))}[\log(\mathrm{p}^*_\alpha(\boldsymbol{\pi}))] \\ &= \mathbb{H}(q(\boldsymbol{\pi}(x))) + \frac{1}{C}\sum_{i=1}^{C} \log(\mathrm{p}^*_\alpha(\boldsymbol{\pi}_i(x)))\end{aligned}$$

The entropy $\mathbb{H}(q(\boldsymbol{\pi}(x)))$ in the KL divergence is perhaps the most challenging term since it is not rigorously defined, as it gives rise to indeterminate forms such as $\delta_0(0)$. Yet, it is possible to isolate the influence of $\delta_0(0)$ (through an algebraic passage to the limit of a Dirac proxy) to recover the standard entropy, denoted $\mathbb{H}(n^x_\pi)$, calculated from the number $n^x_\pi = \sum_i \mathbb{1}_\pi(\pi_i(x))$ of occurrences of π among the set of vectors $\{\pi_i(x)\}$:

$$\begin{aligned}\mathbb{H}(q(\boldsymbol{\pi}(x))) &= -\frac{1}{C}\sum_i \log\left(\frac{1}{C}\sum_j \delta_{\pi_j(x)}(\pi_i(x))\right) \\ &= -\frac{1}{C}\sum_i \log\left(\frac{1}{C} n^x_{\pi_i}\,\delta_0(0)\right) \\ &= -\sum_{\pi\in\{\pi_i(x)\}} \frac{n^x_\pi}{C}\log\left(\frac{n^x_\pi}{C}\right) - \log(\delta_0(0)) \\ &= \mathbb{H}(n^x_\pi) - \log(\delta_0(0))\end{aligned}$$

Therefore, in the absence of data, the ELBO can be formulated as the following:

$$\mathrm{ELBO}(q) \approx \frac{1}{|\mathcal{D}'|}\sum_{x\in\mathcal{D}'}\left(\mathbb{H}(n^x_\pi) - \log(\delta_0(0)) + \frac{1}{C}\sum_{i=1}^{C}\log(\mathrm{p}^*_\alpha(\boldsymbol{\pi}_i(x)))\right)$$

Disregarding the indeterminate form $\log(\delta_0(0))$, which can be treated as an "infinite constant", and temporarily setting aside the entropy term $\mathbb{H}(n^x_\pi)$, we observe that the ELBO is maximized if, for all i, the density $\mathrm{p}^*_\alpha(\boldsymbol{\pi}_i(x))$ is maximal. Assuming the prior verifies the property of *class immiscibility*, this amounts to:

$$\forall i,\ \exists j,\ \boldsymbol{\pi}_i(x) = \boldsymbol{\delta}_j$$

However, if we instead only look at the entropy term $\mathbb{H}(n^x_\pi)$ in the ELBO, we observe that it is maximized when the distributions $\pi_i(x)$ are all pairwise disjoint. By combining both constraints, we conclude that the ELBO will be maximized if:

$$\forall i,\ \boldsymbol{\pi}_i(x) = \boldsymbol{\delta}_i$$

We formally recover the intuitively evident result: each component of the mixture $q(\boldsymbol{\pi}(x))$ should converge toward a unique and distinct vertex of the simplex so that:

$$\forall x,\ q(\boldsymbol{\pi}(x)) = \mathrm{p}^*_\alpha(\boldsymbol{\pi})$$

In presence of data

If the maximization of the ELBO for an ensemble can be solved exactly in the absence of data, the general case where the dataset $\mathcal{D}$ is not empty is more complex. Let us first rewrite the ELBO (Equation 5.3) as a sum of C independent terms, each depending on a single model of the ensemble,

setting $\gamma = \frac{1}{|\mathcal{D}|+|\mathcal{D}'|}$,

$$\begin{aligned}
\text{ELBO}(q) &= \sum_{(x,y)\in\mathcal{D}} \mathbb{E}_{q(\boldsymbol{\pi}(x))}[\log\left(\mathrm{p}(y \mid \boldsymbol{\pi}(x))\right)] - \gamma \sum_{x\in\mathcal{D}\cup\mathcal{D}'} \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{\pi}(x)) \,\|\, \mathrm{p}^*_\alpha(\boldsymbol{\pi})) \\
&= \sum_{(x,y)\in\mathcal{D}} \frac{1}{C}\sum_{i=1}^{C} \log\left(\mathrm{p}(y \mid \boldsymbol{\pi}_i(x))\right) + \gamma \sum_{x\in\mathcal{D}\cup\mathcal{D}'} \mathbb{H}(q(\boldsymbol{\pi}(x))) + \frac{1}{C}\sum_{i=1}^{C} \log\left(\mathrm{p}^*_\alpha(\boldsymbol{\pi}_i(x))\right) \\
&= \frac{1}{C}\sum_{i=1}^{C} \mathcal{L}^{(i)}_{\mathcal{D},\mathcal{D}'}(\boldsymbol{\theta}_i) + \gamma \sum_{x\in\mathcal{D}\cup\mathcal{D}'} \mathbb{H}(q(\boldsymbol{\pi}(x)))
\end{aligned}$$

where

$$\mathcal{L}^{(i)}_{\mathcal{D},\mathcal{D}'}(\boldsymbol{\theta}_i) = \sum_{(x,y)\in\mathcal{D}} \log\left(\mathrm{p}(y \mid \boldsymbol{\pi}_i(x))\right) + \gamma \log\left(\mathrm{p}^*_\alpha(\boldsymbol{\pi}_i(x))\right) + \gamma \sum_{x\in\mathcal{D}'} \log\left(\mathrm{p}^*_\alpha(\boldsymbol{\pi}_i(x))\right) \tag{5.8}$$

can be interpreted as the log posterior for the model $\boldsymbol{\theta}_i$. The entropies $\sum_{x\in\mathcal{D}\cup\mathcal{D}'} \mathbb{H}(q(\boldsymbol{\pi}(x)))$ is a coupling term but, as we will soon see, we will eventually manage to keep the distributions $\boldsymbol{\pi}_i(x)$ away from each other, so that this term will be constant and can be ignored.

While this decomposition in independent terms allows to train every model separately, the resulting ensemble will likely produce unsatisfactory results, with degraded epistemic uncertainty. The reason is that the posteriors $\mathcal{L}^{(i)}_{\mathcal{D},\mathcal{D}'}$ have all the same expression whatever the model $\boldsymbol{\theta}_i$. Therefore, the presence of multiple modes in the prior, and thus in the posterior at least in the low-data regime, can lead the ELBO optimization to converge to a sub-optimal solution, where several models of the ensemble capture the same modes of the posterior for some inputs. This coalescence of output distributions in turn produces lower epistemic uncertainty, as seen previously.

However, it is possible to break this symmetry problem by taking advantage of the structure chosen for the prior in a way to guarantee that each model in the ensemble will specialize on the mode arbitrarily assigned to it. To see this, let us first distinguish two opposing regimes, the low-data regime and the large-data regime, with a smooth transition from one to the other, as illustrated on Figure 5.2. In the former, the regularization term of the ELBO dominates so the output distributions $\boldsymbol{\pi}_i(x)$ of the optimal ensemble are expected to remain close to the Dirac distributions $\boldsymbol{\delta}_i$, but begin to deviate from them. In the latter, these same distributions converge toward the common asymptotic distribution $\boldsymbol{\pi}^*(x) = \lim_{|\mathcal{D}|\to+\infty} \boldsymbol{\pi}_{MLE}(x)$. In this regime the prior has no influence and our model behaves as a standard deep ensemble. In the low-data regime however, the prior still has some influence on the ELBO but, we expect the optimal distributions $\boldsymbol{\pi}_i(x)$ to follow separate paths, in a way that the value of the prior $\mathrm{p}(\boldsymbol{\pi}_i(x))$ of model i is mostly determined by the component of the mixture that is closest to $\boldsymbol{\pi}_i(x)$, *i.e.* the one for which the mass is concentrated around $\boldsymbol{\delta}_i$. This allows to formulate the following approximation (whatever the choice of the prior, whether p^*_α or $\mathrm{p}^\cup_\alpha$):

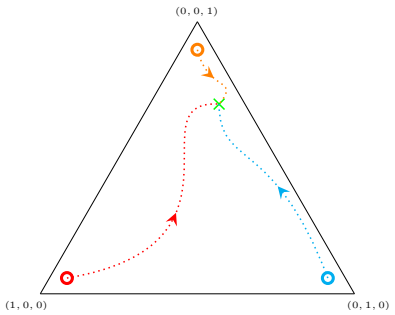


FIGURE 5.2: Illustration of the convergence of optimal output distributions $\{\boldsymbol{\pi}_i(x)\}$ toward $\boldsymbol{\pi}_{MLE}(x)$ (X) as the number of training samples increases.

$$\mathrm{p}(\boldsymbol{\pi}_i(x)) = \frac{1}{C}\sum_{j=1}^{C} \mathrm{p}_j(\pi_i(x)) \approx \frac{1}{C}\mathrm{p}_i(\pi_i(x))$$

where π_i denotes the i^{th} component of the mixture. The logarithm of the prior is then:

- In the case of prior p^*_α:

$$\begin{aligned}\log\left(\mathrm{p}^*_\alpha(\boldsymbol{\pi}_i(x))\right) &\approx \log\left(\frac{1}{C}\,\frac{\Gamma(\alpha+C)}{\Gamma(\alpha+1)}\,\boldsymbol{\pi}_i^{(i)}(x)^{\alpha}\right)\\ &= \alpha\cdot\log\left(\boldsymbol{\pi}_i^{(i)}(x)\right) + \text{ cst.}\end{aligned} \tag{5.9}$$

- In the case of prior $\mathrm{p}^{\cup}_\alpha$:

$$\begin{aligned}\log\left(\mathrm{p}^{\cup}_\alpha(\boldsymbol{\pi}_i(x))\right) &\approx \log\left(\frac{1}{C}\,\frac{\Gamma(C\,\alpha+1)}{\Gamma(\alpha)^{C-1}\,\Gamma(\alpha+1)}\,\boldsymbol{\pi}_i^{(i)}(x)\prod_{j=1}^{C}\boldsymbol{\pi}_i^{(j)}(x)^{\alpha-1}\right)\\ &= \alpha\log\left(\boldsymbol{\pi}_i^{(i)}(x)\right) + (\alpha-1)\sum_{j\neq i}\log\left(\boldsymbol{\pi}_i^{(j)}(x)\right) + \text{ cst.}\end{aligned} \tag{5.10}$$

Injecting these expressions in the log posterior (Equation 5.8) we obtain a specific log posterior for each model $\boldsymbol{\theta}_i$ that is the opposite (negation) of the Conflictual loss as introduced in Section 5.1.2:

- In the case of prior p^*_α, setting $\lambda = \alpha\gamma = \frac{\alpha}{C\,(|\mathcal{D}|+|\mathcal{D}'|)}$

$$\mathcal{L}^{*,i}_{\mathcal{D},\mathcal{D}'}(\boldsymbol{\theta}_i) = \sum_{(x,y)\in\mathcal{D}}\log\left(\mathrm{p}(y\mid\boldsymbol{\pi}_i(x))\right) + \lambda\log\left(\boldsymbol{\pi}_i^{(i)}(x)\right) \tag{5.11}$$

- In the case of prior $\mathrm{p}^{\cup}_\alpha$, setting $\lambda = \gamma = \frac{1}{C\,(|\mathcal{D}|+|\mathcal{D}'|)}$.

$$\mathcal{L}^{\cup,i}_{\mathcal{D},\mathcal{D}'}(\boldsymbol{\theta}_i) = \sum_{(x,y)\in\mathcal{D}}\log(\mathrm{p}(y\mid x,\boldsymbol{\theta})) + \lambda\left(\log\left(\pi_i^{(i)}(x)\right) + (\alpha-1)\sum_{j=1}^{C}\log\left(\pi_i^{(j)}(x)\right)\right) \tag{5.12}$$

However, the Conflictual loss as used in practice does not strictly match the theoretical expression of Equation 5.11 and Equation 5.12, in the sense that the coefficient λ is held constant regardless of the size of $\mathcal{D}$ and $\mathcal{D}'$. This choice, though primarily empirical, is justified by the distinctive behavior of deep networks, which tend to make highly localized predictions. If λ were scaled with the size of the dataset, the influence of the conflictual prior would diminish, especially in low-density regions where its effect is most desired. To prevent this, λ is fixed to deliberately boost epistemic uncertainty in areas of low data density.

While for p^*_α it was possible to merge γ and α due to the choice of the prior and the approximation in Equation 5.9, it is not possible to merge the two in Equation 5.12. We hence consider λ as the regularization coefficient and α will control the shape of the prior $\mathrm{p}^{\cup}_\alpha$. When α is set to 1 in Equation 5.12, we get the same formulation as in Equation 5.11 with an adjusted regularization coefficient. As a final point, and throughout the rest of this thesis, when Conflictual loss is mentioned without specifying the value of α, we are referring to Equation 5.11. However, if $0 < \alpha \leq 1$ is stated, we are pointing to Equation 5.12.

Remark *(Number of models in Conflictual DE).* While the aforementioned discussion focused on Conflictual DE requiring C models, it is possible to consider an ensemble of Conflictual DEs rather than a single Conflictual DE. Equivalently, intead of having only one specialized model per class, one could consider k models for each class, making the count of the base models to $k \cdot C$ in the Conflictual DE.

5.3 Conflictual Loss and Fundamental Principles

The motivation behind Conflictual DE is to fix the issues of uncalibrated epistemic uncertainty for practical BNNs, as it forces ideally diverse predictions when the model is uncertain about a given input. For fairness, the same experiments as those detailed in Chapter 4 will be performed with Conflictual DE, and we will assess if the training with Conflictual loss leads to a calibrated epistemic uncertainty, as defined by the two fundamental principles. In all the reported results, we focus on the uniform prior ($\alpha = 1$) and we set the hyperparameter λ empirically to 0.05. This special case of the Conflictual loss was first introduced and tested in Fellaji et al. (2024), and it was shown to solve the paradoxes of the observed uncalibrated epistemic uncertainty. We build upon this work by incorporating additional experiments to further validate our approach.

5.3.1 A dual-dimensional exploration

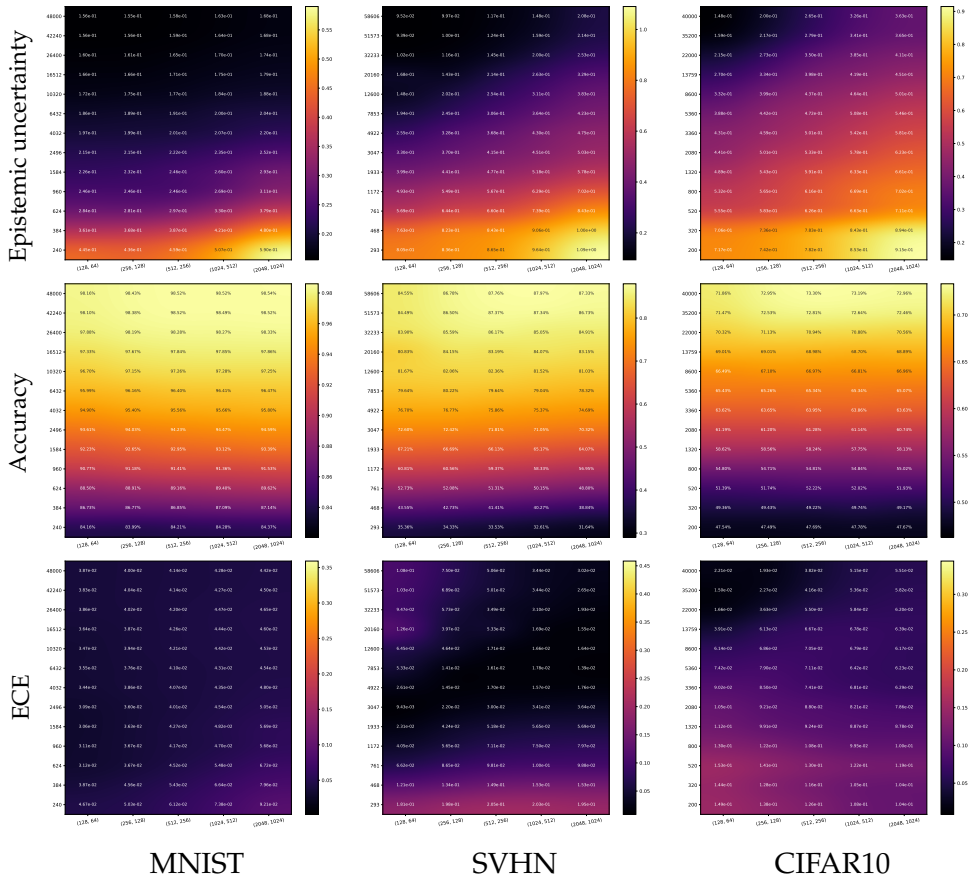


FIGURE 5.3: Results with Conflictual DE. The scale is set per heatmap in the case of the mean of epistemic uncertainty (first row) whereas the scale of the Accuracy (second row) and ECE (third row) is similar to Figure B.1 and Figure B.3, respectively.

The first set of experiments consists of exploring the data-related and model-related principles for the three previously tested datasets (MNIST, SVHN and CIFAR10), as detailed in Section 4.3.2, and we report the results in Figure 5.3. By analyzing the mean of epistemic uncertainty (on the first row), we can visually see that the heatmaps for all the datasets are close to the ideal distribution illustrated in Figure 4.3, with the peak of epistemic uncertainty occurs for the larger model trained with the smallest training set. Therefore, Conflictual DE leads to a calibrated epistemic uncertainty.

Moreover, by quantifying the evolution of epistemic uncertainty in accordance with the process shown in Table 4.1, we can confirm that the two principles are almost always verified (Table 5.1).

While the results might indicate a lack of perfect epistemic calibration, these discrepancies are a direct consequence of the strict count when aggregating[18] the heatmaps, and any potential changes are likely negligible given their minimal significance: the values slightly increase (respectively decrease) when aggregating for the data-related (respectively model-related) principle. As a result, the DE benefits from the diversity imposed by the Conflictual loss. Furthermore, larger models seem to induce greater diversity within the submodels of the DE, resulting from the per-class specialization of the loss function.

		Conflictual DE
Data-related principle	MNIST	100%
	SVHN	97%
	CIFAR10	98%
Model-related principle	MNIST	96%
	SVHN	98%
	CIFAR10	100%

TABLE 5.1: Quantitative summary of epistemic uncertainty heatmaps in Figure 5.3 (similar to Table 4.1).

Now that we made sure Conflictual loss lead to a calibrated epistemic uncertainty, it remains to study whether if it has a negative impact on the performance of the model and on the model calibration (ECE). For the accuracy, it is clear that Conflictual loss does not reduce the model performance, as we can observe in the second row of Figure 5.3 that the accuracy heatmaps are comparable to those shown in Figure B.1. More interestingly, regarding ECE (last row of Figure 5.3), not only the model calibration is not affected by the Conflictual loss, it is better than the tested models from the literature in the sense that the values of the ECE are consistently low and coherent over the different datasets. Finally, it is worth noting that although better calibration is reached on MNIST with MC-Dropout and DE, the results are worse for SVHN and CIFAR10 while with Conflictual DE, the results are not far from these two models on MNIST and significantly better for SVHN and CIFAR10.

5.3.2 Expanding the macro perspective

To gain a better understanding of the evolution of epistemic uncertainty as measured by Conflictual DE, we extend the macro analysis we explored for the tested models from the literature in Section 4.4.1. Rather than tracking the mean on the test set of CIFAR10, we plot the histograms of epistemic uncertainty on the entire test set for different values for the size of the training set and model complexity, and report our findings in Figure 5.4.

On one hand, we can notice the shrinkage of the epistemic uncertainty distributions as we increase the size of the training set and/or decrease the model complexity, with all the distributions being lower bounded by zero. These distributions further illustrate the calibration of epistemic uncertainty that was argued based on the mean on the test set, with the most epistemically uncertain model being the largest that was trained on the smallest training set. On the other hand, epistemic uncertainty takes larger values, as illustrated by the support of the histograms, compared to the models analyzed in Figure 4.5, especially in the low data regime. This is in line with the data-related principle as Conflictual loss leads to a calibration epistemic uncertainty, contrary to the previously analyzed models. Indeed, as the models are trained on more samples, the prior has less influence and thus the support becomes closer to the one for MC-Dropout and DE.

[18]The aggregation consists of computing the percentage of non-positive first differences on the y-axes for the data-related principle, and the percentage of non-negative first differences on the x-axes for the model-related principle.

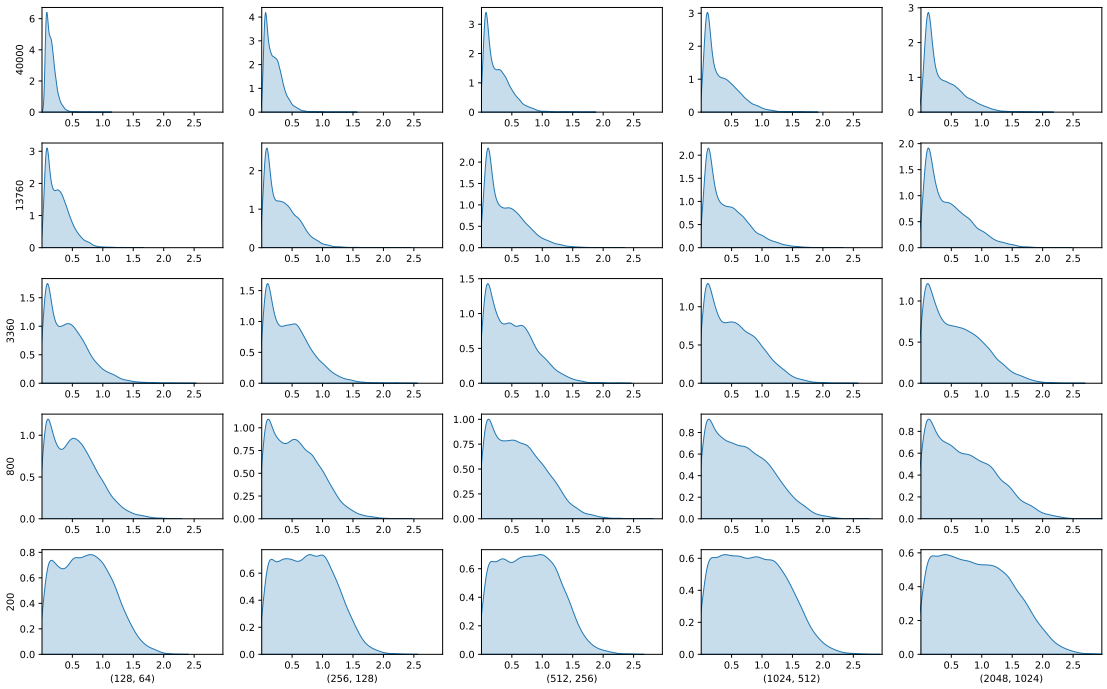


FIGURE 5.4: Histograms of epistemic uncertainty on the test set of CIFAR10 with Conflictual DE. The limits of the x-axes are the same on each figure.

5.3.3 Focusing on specific inputs

We shift now the focus from a macro to a micro analysis by examining specific inputs. As discussed in Section 4.4.2, the selection is done in order to maximize or minimize epistemic uncertainty for all the models with all the configurations on the size of training set and the model complexity. We use the same selected inputs shown in Figure 4.6.

For the samples yielding low epistemic uncertainty (Figures 4.6a and 4.6c), the softmax probabilities are near 1 for the ground-truth and 0 otherwise, and this is true regardless of the size of the training set and the model complexity (Figures 5.5 and 5.7). Additionally, we can notice for both images that, in all the configurations, the second-highest value of the logits is always for the class on which the model is specialized on, with the highest being the actual true label. The only exception is when these two classes match, in which case the highest logit is rightfully the correct class with a higher amplitude with an increased gap to the second-highest logit[19]. Furthermore, a general trend is for the logits to collapse as the model becomes more complex, except for the models trained on the entire CIFAR10 training set.

By looking at the predictions for the samples with high epistemic uncertainty (Figures 4.6b and 4.6d), we can safely conclude that, when in doubt, it is the specialized class associated to the submodel that dominates its prediction (Figures 5.6 and 5.8). Especially in the low data-regime, this dominance is more visible in the softmax-probabilities with an almost maximum total uncertainty (since the BMA is uniform). As the models are trained on more data, we will either have the same observations as for the samples with low epistemic uncertainty if the model learns to correctly predict the input (Figure 5.6), or the BMA will still be close to a uniform prediction with diverse predictions for the submodels when the model fails to learn to correctly classify the input (Figure 5.8).

[19]Being an order-preserving transformation, these observations are true after applying softmax to the logits and are only less visible for softmax-probabilities due to the scale.

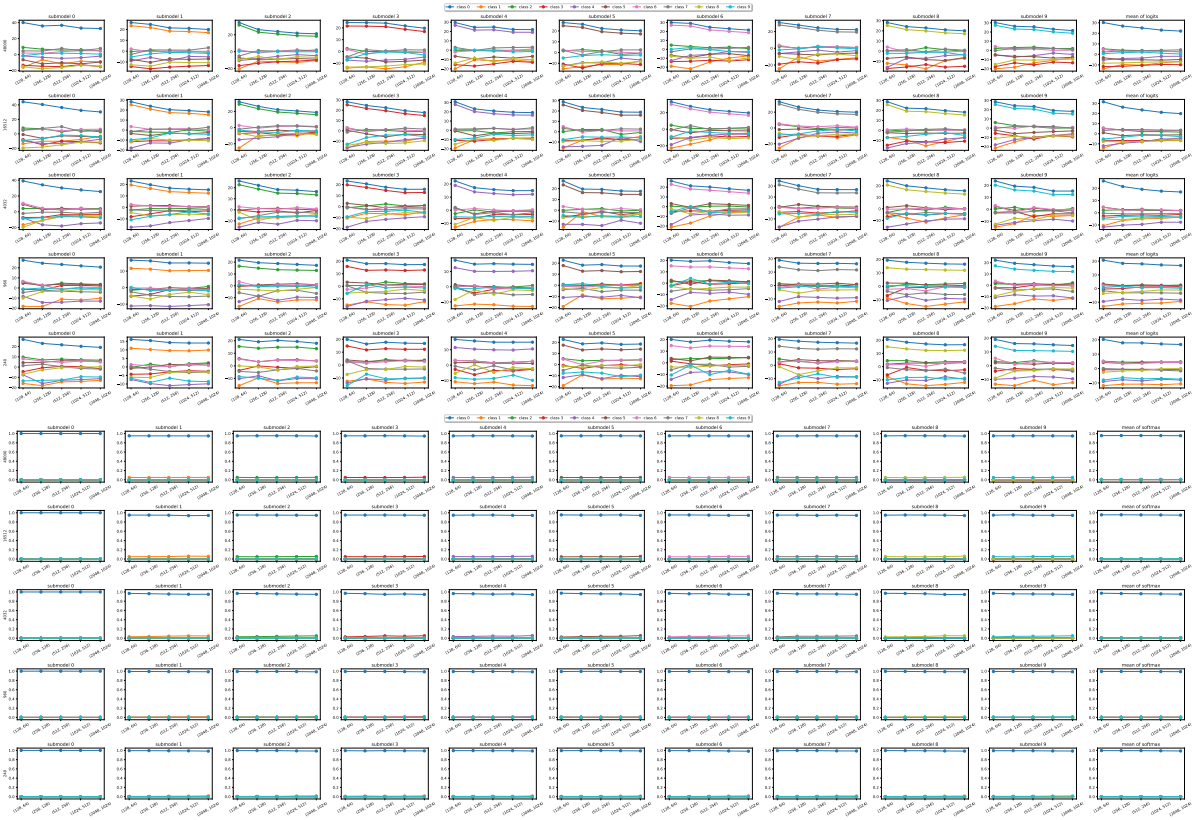


FIGURE 5.5: Predictions by Conflictual DE for Figure 4.6a (the true label is 0). Logits on the top and Softmax on the bottom. See Figure C.10 for a higher-resolution version.

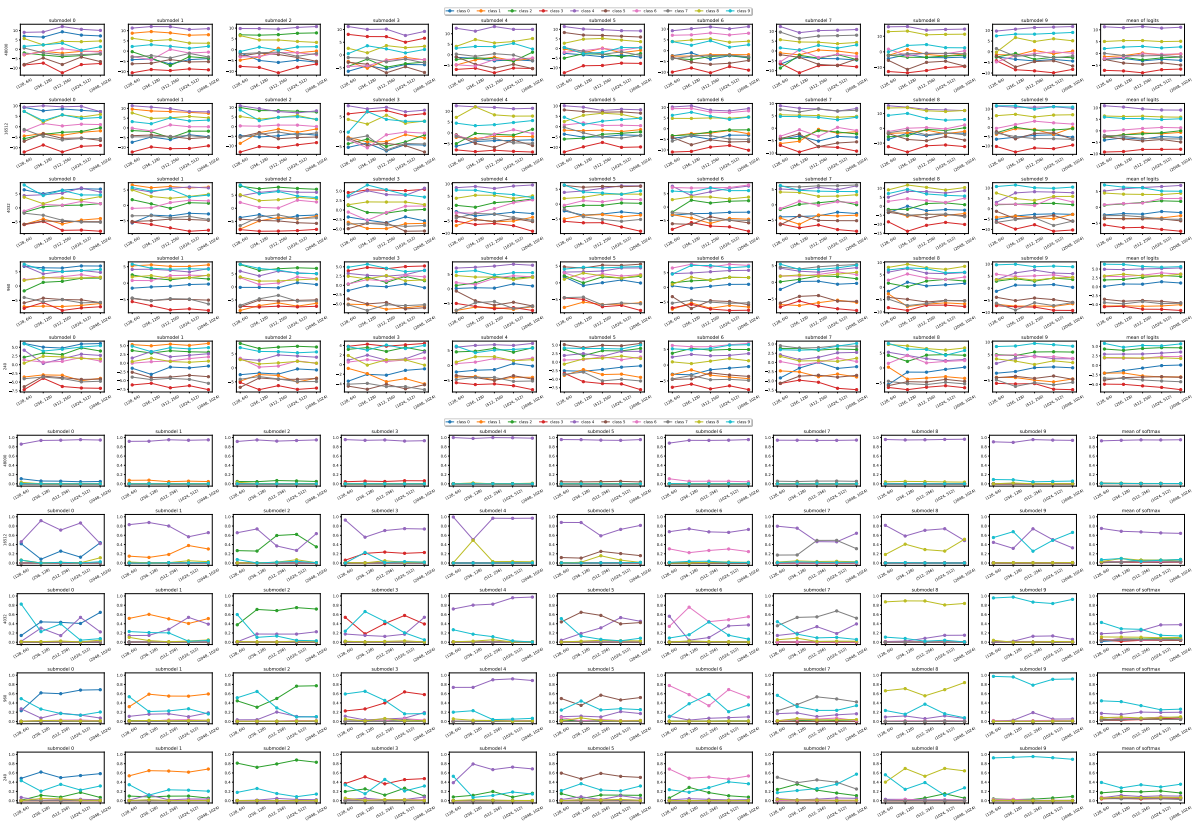


FIGURE 5.6: Predictions by Conflictual DE for Figure 4.6b (the true label is 4). Logits on the top and Softmax on the bottom. See Figure C.11 for a higher-resolution version.

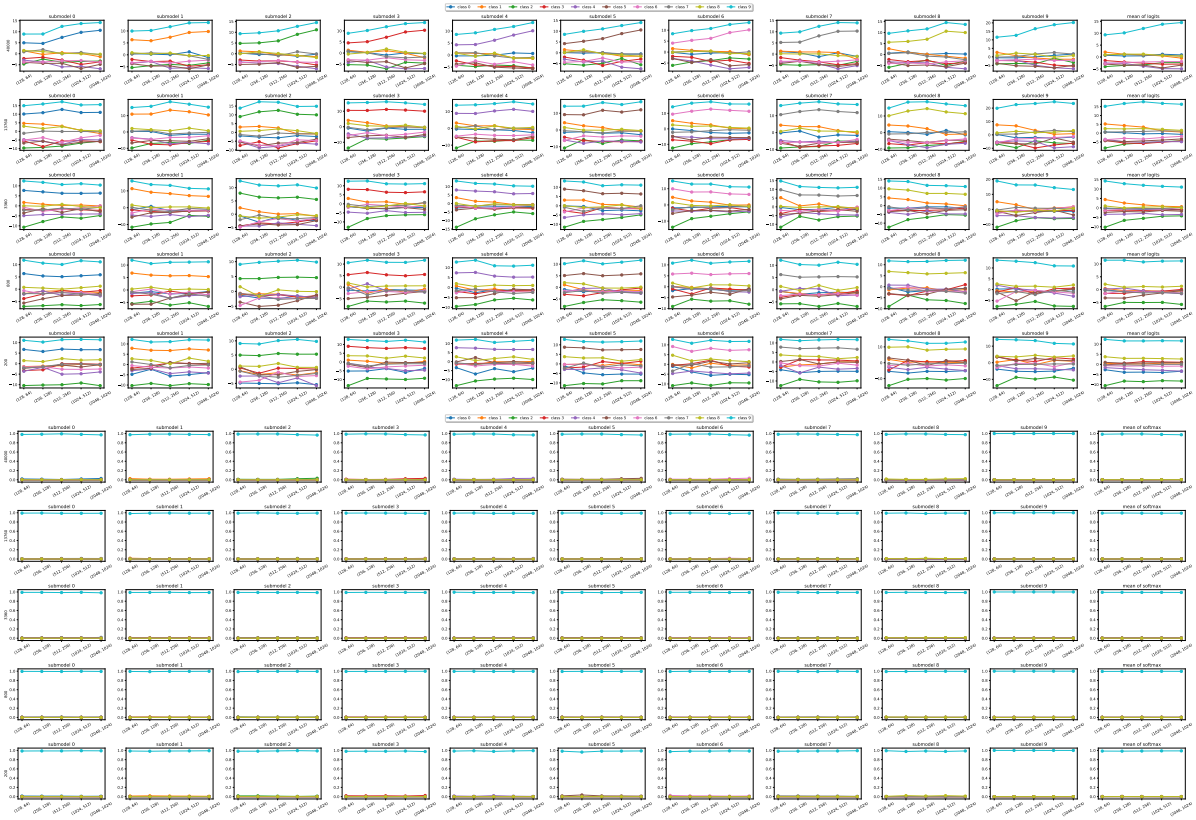


FIGURE 5.7: Predictions by Conflictual DE for Figure 4.6c (the true label is 9). Logits on the top and Softmax on the bottom. See Figure C.12 for a higher-resolution version.

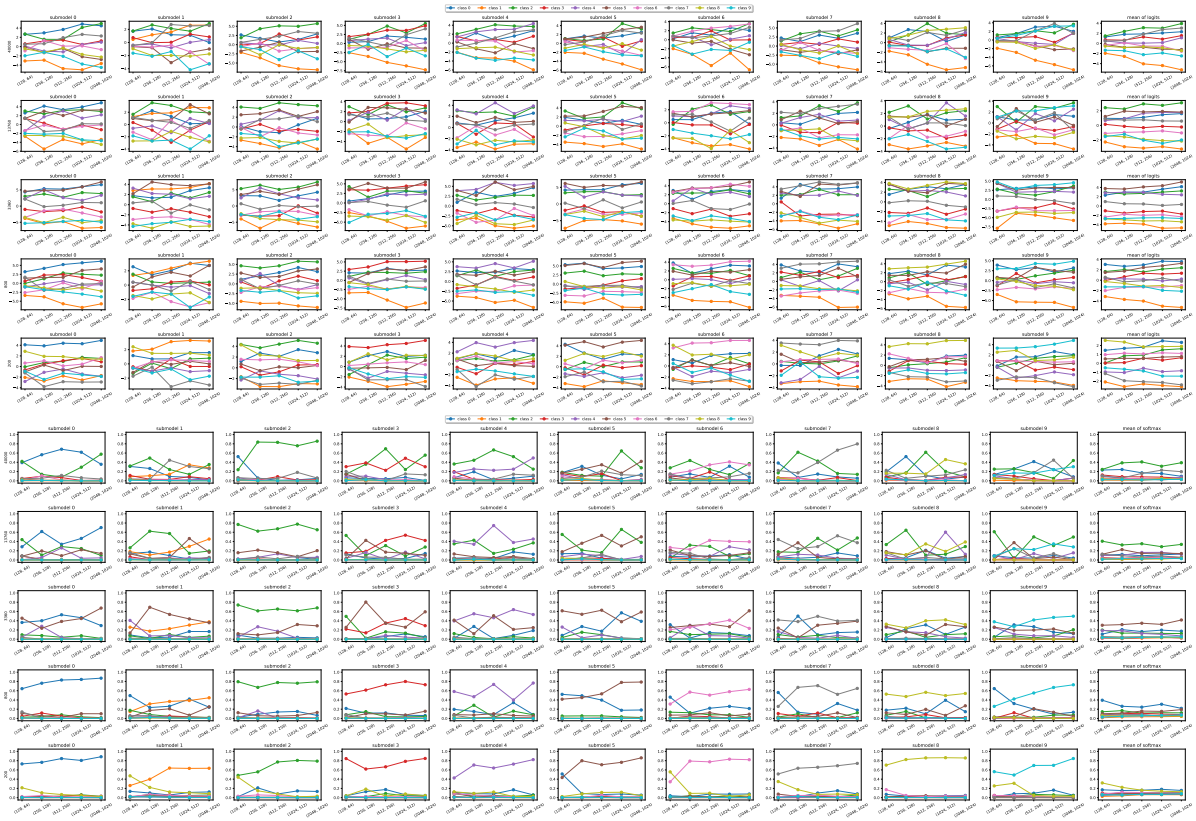
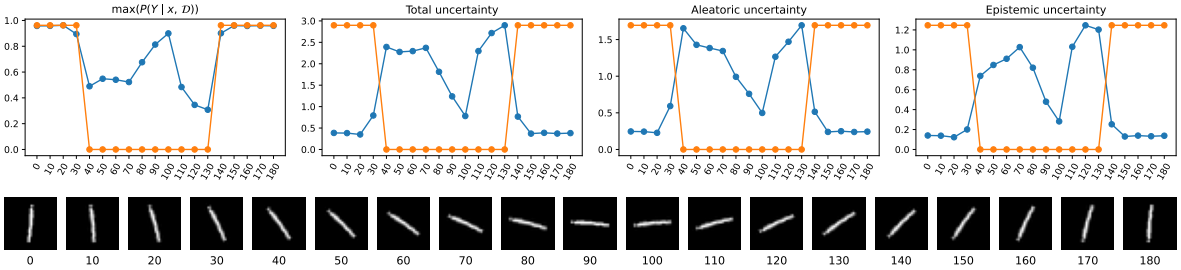


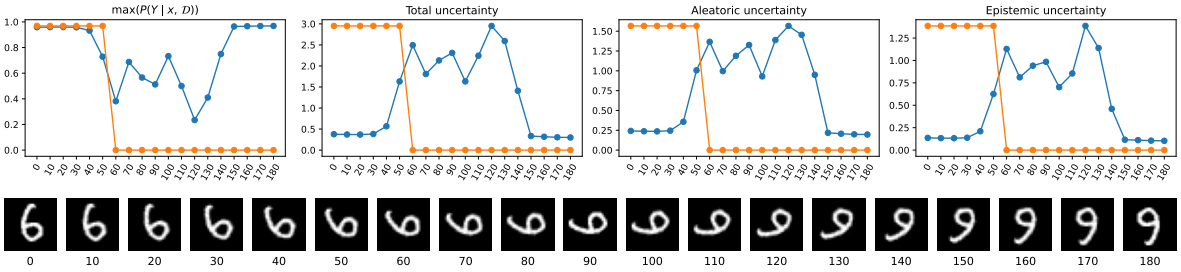
FIGURE 5.8: Predictions by Conflictual DE for Figure 4.6d (the true label is 7). Logits on the top and Softmax on the bottom. See Figure C.13 for a higher-resolution version.

5.3.4 Evaluating aleatoric uncertainty

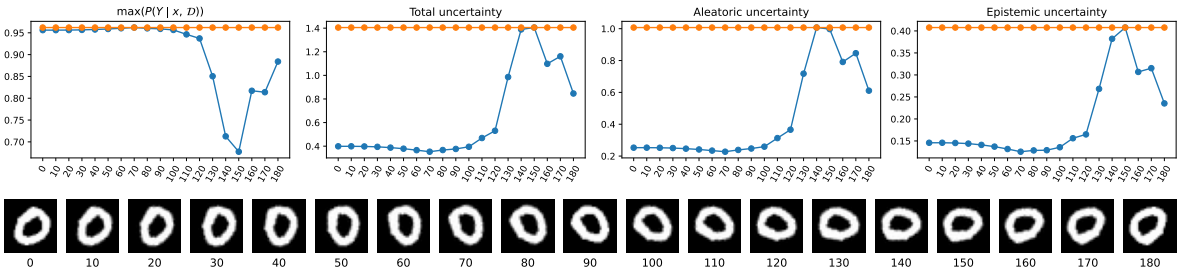
Finally, we would like to continue the analysis carried in Section 4.5, and see the effect of rotating inputs and of using ambiguous images on uncertainties. As discussed in Section 5.3.1, Conflictual DE are calibrated from an aleatoric perspective when considering ECE (Figure 5.3). Therefore, we expect Conflictual DE to reliably represent aleatoric uncertainty in these two scenarios.



(A) **1-to-1**



(B) **6-to-9**



(C) **All-0**

FIGURE 5.9: Rotating selected images from MNIST for Conflictual DE. The orange curve indicates whether the predictions match the target.

Rotating samples. In all the examples shown in Figure 5.9, aleatoric uncertainty is generally higher than epistemic uncertainty. Especially, the gap is higher when observing large rotations with **All-0**, further advocating for the cause of the "inconsistent" classifications being due to the noise introduced by rotating the digit "0". From an uncertainty perspective, we expect **All-0** to have low epistemic uncertainty since the rotation consistently produces similar images. Although the case of **All-0** with Conflictual DE is the closest to DE, the curves are different for **1-to-1** and **6-to-9**. As Conflictual loss encourages the model to have diverse predictions when unable to correctly predict the output, we expect epistemic uncertainty to increase if given an input that does not belong to any class: the "1" and the "6" rotated by 80° are hardly recognizable digits. As a result, epistemic uncertainty is significantly higher compared to the previously tested models. Moreover, even though the maximum of the BMA is quite confident, the uncertainties are not negligible as it is the case with MC-Dropout and DE. On a side note, the 100° rotation of the "1" lead to confident

BMA and a drop in uncertainties, yet this is not specific to Conflictual DE as we have the same observation in Figure 4.13.

Ambiguous-MNIST. The second class of inputs to be tested is the class of ambiguous samples, as initially introduced in Section 4.5.2, and the results are shown in Figure 5.10. We notice that on most of the evaluated inputs, Conflictual DE leads to maximum (or at least fairly high) total uncertainty. This high uncertainty is mainly explained by the epistemic component, and although it was also the case with the tested models in Figure 4.16, it is more pronounced with Conflictual DE. These findings will be further detailed in Section 6.1.6 with a deeper analysis of the Ambiguous-MNIST dataset, to whether these selected examples should be in fact considered as OOD samples as they do not resemble to the training data, or shall we still analyze them from the lens of noisy and heavily corrupted ID samples. Moreover, aleatoric uncertainty for these samples is higher than with unambiguous MNIST samples (around 0.25 in Figure 5.9), as it should, suggesting that the model is also aware of the noise component in the data.

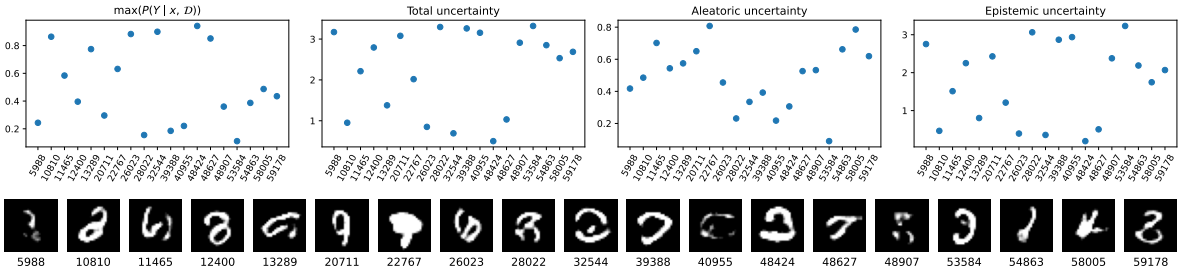


FIGURE 5.10: Ambiguous-MNIST inputs for Conflictual DE.

5.3.5 Conflictual loss for a complex learner

Finally, one might wonder if Conflictual loss favors the diversity even in the context of a more complex base learner. To this end, we adapted the experiment on the deep ensemble of ResNet18 where the hole of epistemic uncertainty was first highlighted (Section 4.3.1). Since the ensemble consists of 10 learners and is trained on CIFAR10, the Conflictual DE version is straightforward[20], and the results are hence comparable.

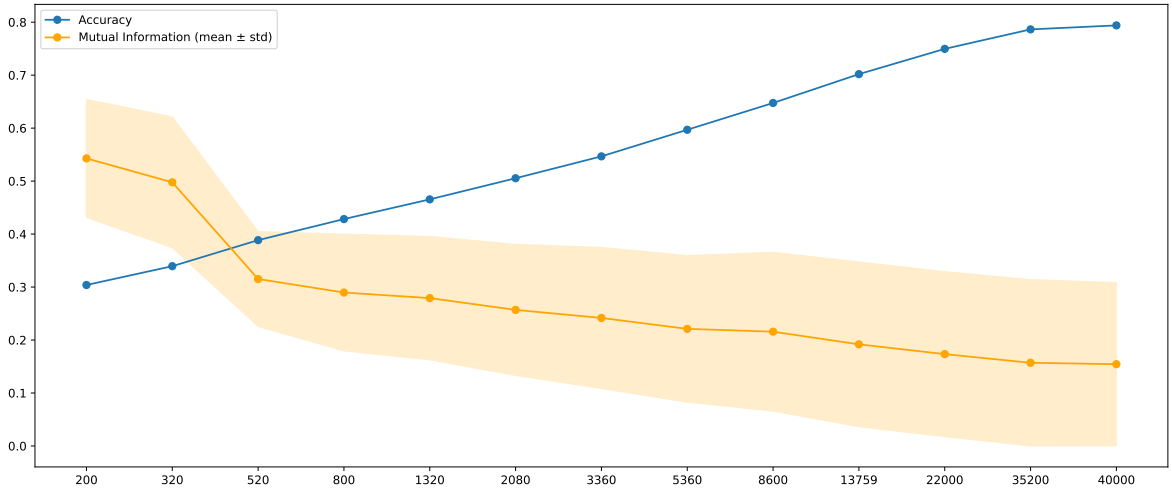


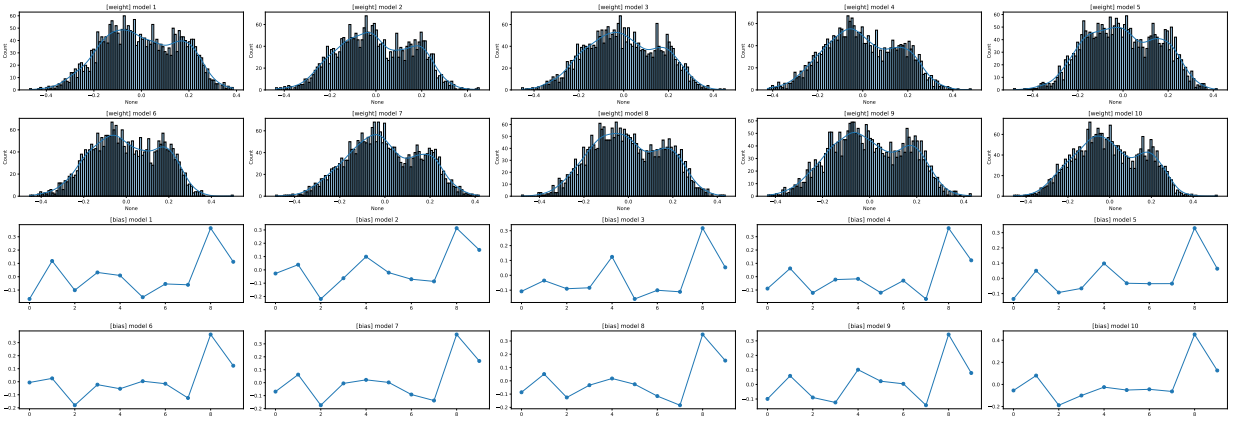
FIGURE 5.11: Similar setup to Figure 4.1, with the DE replaced by Conflictual DE.

[20]We set the hyperparameter λ to 0.01.

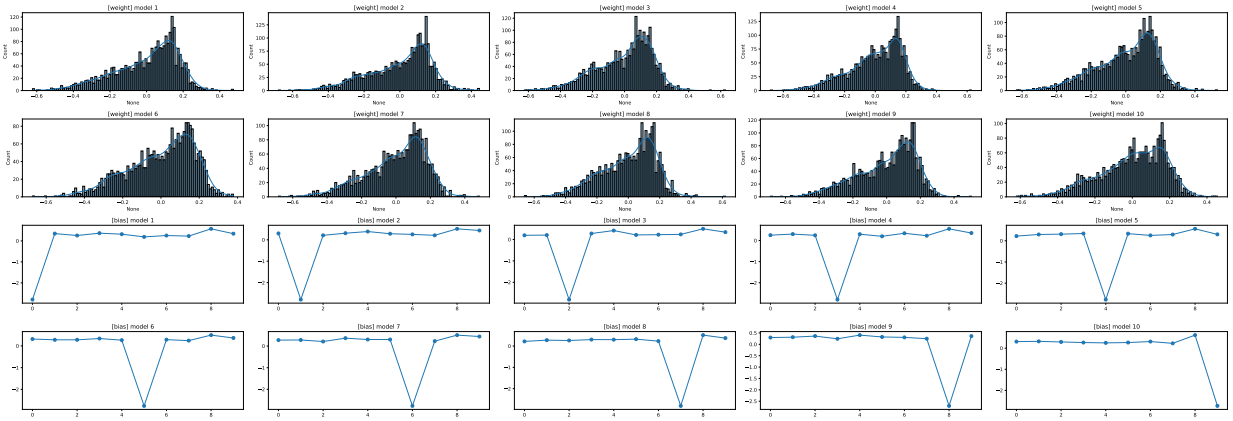
Figure 5.11 presents the results with the Conflictual DE (of ResNet18), which align with the findings from the dual-dimensional analysis of the MLPs (Section 5.3.1). In fact, as we train on additional data, the model becomes more accurate and most importantly, epistemic uncertainty decreases in accordance with the data-related principle. Consequently, Conflictual loss does indeed fix the epistemic uncertainty hole, independent of the model architecture.

5.4 The diversity in Conflictual DE

The idea of a diverse deep ensemble was explored in previous works such as *Deep Anti-Regularized Ensembles* (DARE) (de Mathelin et al., 2023) and *Maximum Weight Entropy* (de Mathelin et al., 2025) where the notion of an "anti-regularization" is examined. In essence, training a DE with the cross-entropy loss results in small values of the weights making the outputs of the ensemble closely aligned. In order to mitigate that, anti-regularization aims at increasing the amplitude of the model's parameters. As a result, these large weights of the models lead to diverse outputs in the case of DE. Several methods exist to promote higher-magnitude parameters. For instance, a regularization is added during training in the case of DARE to favor large values for the parameters while keeping the loss within an acceptable range. In contrast, Maximum Weight Entropy starts from a standard pretrained network and fits a distribution over its weights by maximizing entropy while maintaining performance on the training data.



(A) DE



(B) Conflictual DE

FIGURE 5.12: Histograms of the weights and the biases for the last Linear layer in the models. In each subfigure, the first two rows are for the weights while the last two are for the biases.

Inspired by the justification of the anti-regularization, we explore the weights-space of Conflictual DE for a better understanding of the sources of the outputs' diversity, and how the regularization manifests itself in the weights once the models are trained. To this end, and under the same training configuration, we study the distribution of the parameters for a DE and a Conflictual DE layer-wise. Given the nature of the MLP models, the idea is to look at the histograms of both the weights and the biases in the three Linear layers for the ten submodels once trained.

We start by examining the histograms of the parameters of the last layer (mapping to the outputs) in Figure 5.12. For DE (Figure 5.12a), the distributions of the weights in the submodels are to some extent similar from an aggregated perspective, and are centered around zero while the biases exhibit resembling trends. On the other hand, the distributions in the case of Conflictual DE stand out for two main reasons compared to DE (Figure 5.12b). First, the distributions of the weights are positively skewed with the peak around 0.15, and their support is larger than the case of DE. Second, and perhaps the most noticeable difference, is that the bias in each submodel depends on the class that the submodel is specialized in. The large negative bias for the associated class in the submodel suggests that this bias helps counterbalance the submodel's specialization on this class, preventing overly confident predictions. This hypothesis is empirically supported in Section 5.3.3, where the logits for this class were not excessively high.

Surprisingly, this difference in the parameters-space between DE and Conflictual DE is only visible for the last layer. The skewed weights distribution and the class-dependent biases do not appear in the first two linear layers, making the case that the diversity of Conflictual DE and the impact of the Conflictual loss are largely attributed to the final classification layer.

5.5 Conflictual Model

Perhaps one of the limitations of Conflictual DE, as formalized in Section 5.2.3, is the necessity of having at least C submodels, each favoring a distinct class. For the tested datasets, having 10 submodels was still manageable, however, for more complex tasks in terms of the number of classes, this could be infeasible from a computational and/or storage perspectives. Based on our findings in Section 5.4, we propose *Conflictual Layer* that takes advantage of the diversity created in the last layer of the Conflictual DE.

In Section 2.2.2, we discussed different optimized ensembling strategies, in particular, the use of weight sharing in TreeNet (Lee et al., 2015) for an optimized ensemble (Figure 2.2). We thus introduce a Conflictual model combining Conflictual DE and TreeNet. The first adaptation is to have a model where all the blocks/layers are shared expect for the last one, which we refer to as Conflictual Layer. It consists of an ensemble of linear layers that takes the outputs of the common backbone. The "diversity-encouraging" loss of TreeNet will be replaced by Conflictual loss, hence enforcing diversity through conflict. This enables maximum reduction of the parameter count, making the approach applicable to datasets with large number of classes.

As we are dealing with a single model in the case of a Conflictual model, the training is sequential. To this end, the proposed loss will be the averaged Conflictual loss computed from each element in the Conflictual layer:

$$\mathcal{L}_{\text{CL}}(\boldsymbol{\theta}, \mathcal{D}) = \frac{1}{k \cdot C} \sum_{i=1}^{k} \sum_{j=1}^{C} \mathcal{L}_{\text{CL}}^{(j)} \left((\boldsymbol{\theta}_b, \omega_{i \cdot j}), \mathcal{D} \right) \tag{5.13}$$

with $\boldsymbol{\theta}_b$ being the parameters of the shared backbone, and ω_i the parameters of the component i of the Conflictual layer such that $i \in \{1, \ldots, k \cdot C\}$. Furthermore, the use of $k \cdot C$ components in the Conflictual Layer is inline with the remark in Section 5.2.3.

To some extent, the backbone allows embedding the data in a lower dimensional space, or equivalently, computing general features that will be used to train the Conflictual layer. Hence, the learning process consists of better projecting the data by the backbone, and producing calibrated epistemic uncertainty through the Conflictual layer.

Remark *(The use of features from pretrained models).* The reported results on CIFAR10 were based on training MLPs for which the inputs were computed using a pretrained ResNet, hence embedded in a lower dimensional space. Our initial motivation behind this is to simplify study the calibration of epistemic uncertainty, especially the model-related principle, without retraining the backbone for features extraction.

Histograms of Conflictual layer. Under the same experimental setting as in Section 5.4, we explore the weights of the trained Conflictual Model: a single MLP where the last layer is replaced by Conflictual layer. When plotting the histograms for the Conflictual layer, as illustrated in Figure 5.13, we observe that they follow similar trends to those of Conflictual DE as the weights are positively skewed and the biases are dependent on the specialized class. This observation is not specific to an MLP trained on MNIST that it generalizes to other model architectures trained on different datasets, as can be seen in Figure D.1 for example in the case a Conflictual convolutional model trained on MNIST.

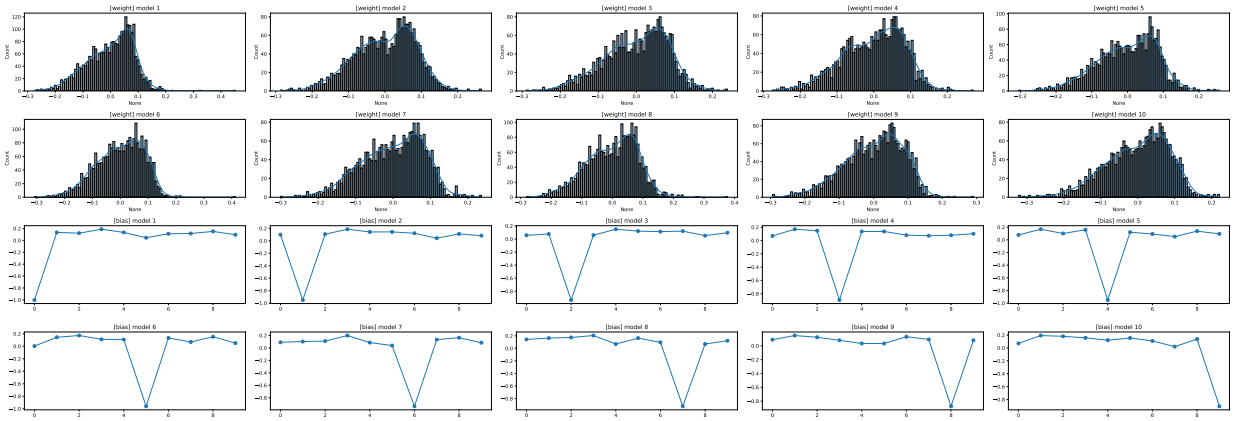


FIGURE 5.13: Histograms of the weights and the biases of a Conflictual MLP Model with the last layer being a Conflictual Layer (for $\lambda = 0.05$).

A dual-dimensional exploration. The similarities with Conflictual DE of the Conflictual Model persist also when evaluating the two fundamental principles in the heatmaps, as illustrated in Figure 5.14. By using a Conflictual layer, epistemic uncertainty is overall calibrated and verifies the data-related and model-related principles. The cases when these principles where not verified (as aggregated in Table 5.2) are generally for insignificant differences, and they were reported for the purpose of full disclosure and following the same level of rigor as in Table 4.1 and Table 5.1. Furthermore, Conflictual layer leads to a calibrated aleatoric uncertainty, as measured by the ECE, similar to the observations with Conflictual DE. Finally, the resulting models do not show drops in performance as measured by the accuracy, maintaining their predictive performance.

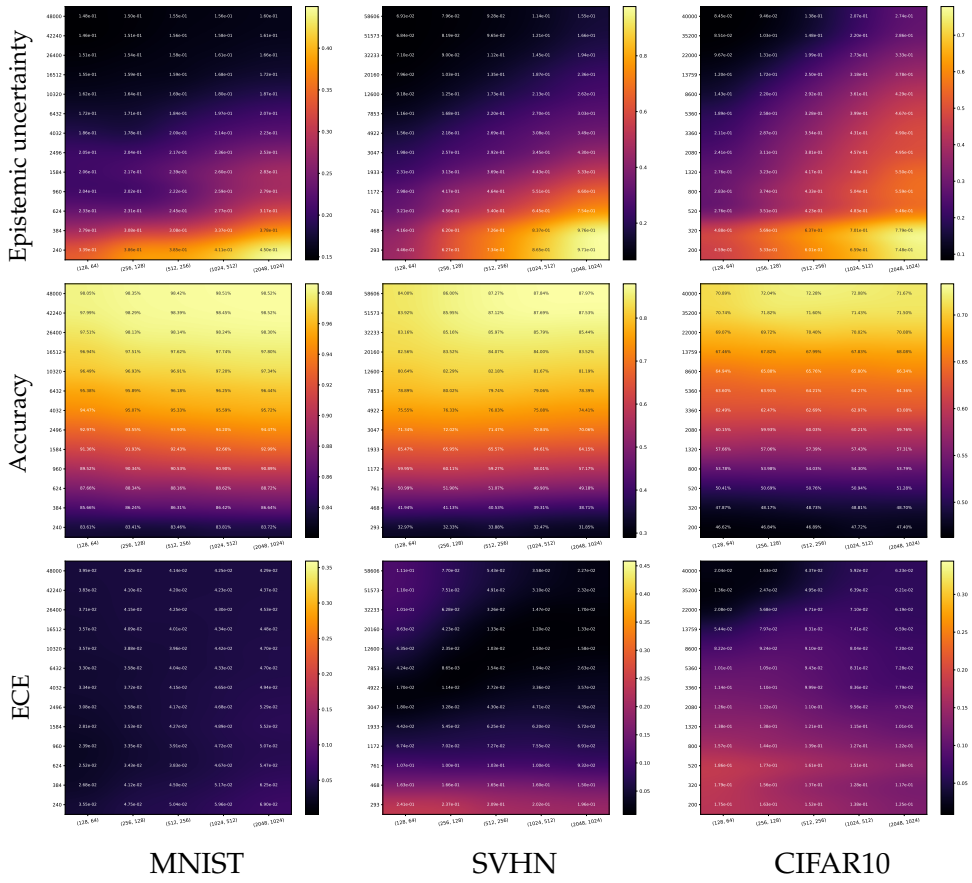


FIGURE 5.14: Results with Conflictual Model. The scale is set per heatmap in the case of the mean of epistemic uncertainty (first row) whereas the scale of the Accuracy (second row) and ECE (third row) is similar to Figure B.1 and Figure B.3, respectively.

		Conflictual Model
Data-related principle	MNIST	90%
	SVHN	97%
	CIFAR10	84%
Model-related principle	MNIST	88%
	SVHN	100%
	CIFAR10	100%

TABLE 5.2: Quantitative summary of the epistemic uncertainty heatmaps in Figure 5.14 (similar to Table 5.1).

As a result, Conflictual Model offers calibrated epistemic uncertainty thanks to the use of the Conflictual loss adapted to a multi-heads model while providing faster training and inference times along with reduced storage requirements.

5.6 Conclusions

With the prior distribution having a significant impact on the epistemic uncertainty calibration, setting an uninformative prior that was rewritten to favor diversity, fixes the pitfalls of the uncalibrated epistemic uncertainty of the commonly used models. By applying this prior to a DE,

we showed that Conflictual DE has a calibrated epistemic uncertainty, as defined through the fundamental principles. When exploring the weight space of the trained Conflictual DE, we noticed the diversity is mainly manifested in the last classification layer, both for the weights and the biases. This gives raise to an optimized version of the Conflictual DE, that was referred to a Conflictual Model, which consists of a multi-heads model with conflictual heads, thanks to the Conflictual loss. Under the same experimental setup, Conflictual Model is shown to lead as well to a calibrated epistemic uncertainty and the same trends as with Conflictual DE. In the next chapter, these models will undergo extensive evaluation across a diverse set of challenging tasks.

CHAPTER 6

LEVERAGING EPISTEMIC UNCERTAINTY IN PRACTICE

"When knowledge becomes practical, it elevates human society."
Al-Farabi

Epistemic uncertainty is often used and assessed in different tasks, namely for OOD detection. In addition, it is considered to be the ideal metric in the context of active learning to select samples to be labelled. In the literature, the validity of any epistemic uncertainty measure is often studied based on how they perform on these respective tasks. Although this test is suboptimal as one should rather study the verification of the data-related and model-related principles, we study in this chapter the influence of a calibrated epistemic uncertainty on these applications. Nonetheless, measuring epistemic uncertainty is only possible for a subset of models, particularly BNNs as discussed in Chapter 2, and unfeasible in the case of standard models. Hence, we will start by exploring how OOD detection is performed with standard models and also discuss the distinct situation of BNNs. We will conduct further testing on a variety of models and benchmark their performance in the task of OOD detection. Moreover, the second part of this chapter is on active learning, and more precisely Bayesian active learning where a brief literature review is presented, followed by some experiments on MNIST and CIFAR10.

6.1 Identifying the Incorrect and the Unknown

Achieving perfect (*i.e.* Bayes optimal) accuracy is desirable, though it is an ideal goal that may be difficult or even impossible to fully attain. As a result, there will always be inputs that the model will misclassify, which are either noisy, ambiguous or just hard-to-classify inputs. Therefore, we expect the model not only to be uncertain about these faulty predictions, but also to indicate uncertainty when encountered. The distinction was illustrated in Figure 4.2 in the case of a DE of ResNets trained on CIFAR10, where we notice differences between these two categories (*i.e.* correctly and misclassified inputs) from an uncertainty point of view, and will be further investigated in this section.

Additionally, DL models are generally trained and evaluated under the assumption of a closed-world for which all the samples used are sampled from the same dataset $\mathcal{D}$. However, in practice, it is likely to have an input sampled from a different distribution. These samples are referred to as *out-of-distribution* (OOD)[21] samples, emphasizing their differences from the *in-distribution*

(ID) samples (Hendrycks and Gimpel, 2017). It remains thus important for the model, especially in safety-critical applications, to detect these samples and to correctly distinguish ID and OOD samples. Likewise, owing to the overconfidence of DL models (Szegedy et al., 2015; Guo et al., 2017; Sensoy et al., 2018), a reliable OOD detection becomes more critical.

While both misclassification and OOD detection can be considered as part of the more general task of outlier detection (Hawkins, 1980; Brodley and Friedl, 1996; Knox and Ng, 1998; Hodge and Austin, 2004), we make the following nuanced definition between the two. On one hand, OOD detection identifies samples that deviate from the distribution of training data, often involving models that are trained on a specific dataset and need to recognize when new data is too different from that set. Outlier detection, on the other hand, focuses on identifying data points that are far from the rest of the data in a given dataset, regardless of whether the data is part of the same distribution or not. To the best of our knowledge, misclassification and OOD detections were first discussed in the context of neural networks in the work of Hendrycks and Gimpel (2017); Kendall and Gal (2017). For recent surveys on OOD detection in machine learning, we refer to Yang et al. (2022, 2024).

Consider, for example, a model trained on a binary classification dataset of cats and dogs. When the model makes an error on a test input, we must be able, thanks to the uncertainties, to capture and assess this failed prediction. For instance, the misclassification of an input could be due to a noisy image, or for an image of an animal that has features-like both cats and dogs, hence caused by a high aleatoric uncertainty. In addition, in the case where the model is trained on only a subset of the available training dataset, its epistemic uncertainty implies that the model could misclassify some valid ID images. Furthermore, if we feed the model with an unrelated input (an image of a tree, a tv or a car for example), the model should identify it as an OOD input.

6.1.1 Score-based approaches for standard DL

As detailed in Yang et al. (2022), there exist different methodologies for OOD detection that could be grouped in three main categories: post-hoc methods, training-time regularization and training with OOD exposure. The latter two methods change the training pipeline in order to favour the model capability to detect OOD examples either by using specific loss functions or by training the model with ID and OOD samples, as used in the Prior Networks (Malinin and Gales, 2018) for example.

For several reasons, we mainly focus on post-hoc methods, especially score-based approaches with no additional "training". First, Yang et al. (2022) showed that post-hoc methods usually have better OOD detection compared to the training related methods. Second, exposing the model to OOD samples during training seems suboptimal to enhance OOD detection. As a matter of fact, the choice of the OOD dataset to sample from is not universal: ID dataset is unique compared to the infinite choice for OOD dataset, a good detection on one dataset does not ensure the detection for all OOD dataset. Finally, we believe that the model should ideally be self-aware and able to detect the misclassified and OOD samples, notably in the case of BNNs. For a safe deployment, models should not be optimized only to maximize accuracy on the ID test samples, but also to identify incorrect and unknown straight out of the box. Hence, we will discuss score-based approaches, mostly relying on softmax probabilities and uncertainties.

MSP. The first proposed baseline is the *Maximum Softmax Probability* (MSP) (Hendrycks and Gimpel, 2017) which relies on the maximum value of the softmax probabilities vector to decide whether an input is either misclassified or an OOD. The main argument is that MSP is higher in the case of correctly classified inputs compared to misclassified and OOD samples (or not). However,

[21] Also referred to in the literature as *out-of-data* (Kendall and Gal, 2017).

as demonstrated in Nguyen et al. (2015), DL models are easily deceived and can lead to high MSP for human-unrecognizable inputs. Furthermore, MSP considers only the dominant class which could be suboptimal in multiclassification tasks, and does not offer a distinction between aleatoric and epistemic uncertainties.

Energy. Considering the critics of MSP, Liu et al. (2020b) advocated for the desirability of the energy score for OOD detection as it better aligns with the probability density of the inputs. This score, as illustrated in Equation 6.1, consists of the LogSumExp function applied to the logits $f_\theta(x) = \left[f_\theta^{(1)}(x), \ldots, f_\theta^{(C)}(x)\right]$. As the energy is pushed down for a model trained with negative log-likelihood, if an input has high energy, it is likely that it was not sampled from the same distribution as the training datapoints, and thus can be considered as OOD. With MSP on the other hand, the previous implication is not guaranteed. What matters to us is the use of energy at inference time to detect OOD samples, hence, the *temperature* T is set to 1 in Liu et al. (2020b), making the energy score parameter-free. It is important to highlight that with the energy score, the information in the logits is preserved contrary to the use of softmax probabilities. Finally, and from a practical perspective, when we are dealing with DE or MC-Dropout, the reported energy will be the average of the energies computed for each submodel.

$$E(x; \theta) = -T \cdot \log \left(\sum_{i=1}^{C} \exp \left(\frac{f_\theta^{(i)}(x)}{T} \right) \right) \tag{6.1}$$

Entropy. The entropy of the softmax probabilities is yet another strong baseline for misclassification and OOD detections (Lakshminarayanan et al., 2017; Smith and Gal, 2018; Ovadia et al., 2019; Valdenegro-Toro, 2021; Kirsch et al., 2021). In particular, Ovadia et al. (2019) argued that predictive uncertainty, as measured by entropy, is significant when there is a distributional shift. Nonetheless, given the fact that entropy is maximal for a uniform distribution and near zero for an overconfident prediction, one could argue that it has the same pitfalls as MSP: the predictive entropy will be low for the same inconsistent inputs as in Nguyen et al. (2015), and it does not offer a distinction between aleatoric and epistemic uncertainties.

6.1.2 Utilizing established foundations

Although there exist additional scores for standard DL models, our focus will be mainly on the aforementioned scores. For example, Mucsányi et al. (2024) tested in their benchmarks different methods and found that Mahalanobis distance (Lee et al., 2018b) performs the best for OOD detection, and the worst for misclassification detection. As Mahalanobis distance relies on OOD samples during training to learn a latent mixture of Gaussians[22], its effectiveness is therefore logical for OOD detection. Additionally, the use of OOD samples for training the OOD classifier raises multiple questions, namely, on the choice of the OOD dataset.

Prior networks (Malinin and Gales, 2018) is one yet another example that uses OOD samples during training to enhance OOD detection. In all these special loss functions, as the model has access to a subset of OOD examples, there is no guarantee that it will successfully detect a different subset of OOD samples. The choice of an OOD dataset to be used in training is subjective and not unique and thus could harm the generalization of the model, and more broadly the OOD detection. Kirsch et al. (2021) raised similar issue and discussed its effect on uncertainties: having OOD samples in the training dataset will affect the calibration of epistemic uncertainty as the latter

[22]The outputs of the penultimate layer of the pretrained DL model is used to learn the latent mixture of Gaussians, then the score measures the distance of the test input to the closest class-conditional Gaussian distribution.

will be shifted into aleatoric uncertainty, as the model handles unfamiliar patterns in the OOD dataset as inherent data randomness.

The empirical conclusions of Mucsányi et al. (2024) are that the superiority of a specific score over the others for a given task and dataset does not hold true when changing the task and/or the dataset. Thus, they claim that since there are various task groups, a single uncertainty estimator cannot be universally applied. We disagree with the previous conclusion as it seems as if we are throwing the baby out with the bathwater. As discussed in Section 2.3.4, the use of mutual information as a measure of epistemic uncertainty was also criticized due to the observed calibration paradoxes. Yet, as demonstrated in the previous chapters, these paradoxes were mainly caused by the lack of epistemic calibration.

6.1.3 Detection with BNNs

The previous section focused mainly on score-based methods for misclassification and OOD detection in the case of standard DL models. Given that our work is centered on Bayesian models, we will discuss in the following the specific scores we obtain by employing these models, more precisely, uncertainty-based scores.

Epistemic uncertainty. It is argued that OOD detection should solely rely on epistemic uncertainty (Kendall and Gal, 2017; Malinin and Gales, 2018; Amersfoort et al., 2020; Wang et al., 2021; D'Angelo and Henning, 2022). As epistemic uncertainty is mostly related to model disagreement, the models sampled from the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$ should agree on ID samples and disagree on OOD samples. Thus, epistemic uncertainty is considered the ideal measure for OOD detection. It is important to highlight that a total disagreement is a stronger form of total uncertainty: each sampled model from the posterior distribution explains the data in a different (and perhaps confident) way, leading to a uniform predictive distribution.

The use of epistemic uncertainty to identify unknown samples has been the subject of criticism in a number of papers in the literature. For instance, Henning et al. (2021) questioned the common use of epistemic uncertainty for this identification due to some empirical insignificant differences with non-Bayesian models: the detection based on epistemic uncertainty for some BNN was only marginally superior to these models. However, some explanations were offered in an updated version of the paper (D'Angelo and Henning, 2022), most importantly, the choice of the prior distribution. In case of model or prior misspecification, OOD detection based on epistemic uncertainty could be erroneous. From our perspective, this is directly linked to the calibration of epistemic uncertainty, which depends on the choice of the prior distribution. Therefore, when epistemic uncertainty is calibrated for a given BNN, we expect epistemic uncertainty to be the ideal score for OOD detection.

> **Remark** *(Dirichlet models).* Since epistemic uncertainty can be further decomposed into model and distributional uncertainties for EDL (Section 2.4.3), Malinin and Gales (2018) advocated for the use of the latter for OOD detection. In our experiments, epistemic uncertainty will refer to the distributional uncertainty in the case of these models.

Aleatoric and total uncertainties. In the previous section, we discussed the use of entropy and MSP as scores for OOD and misclassification detections. One can view both of them as measures of total uncertainty (Malinin and Gales, 2018), or of aleatoric uncertainty when we are dealing with standard models. Since misclassification is mostly due to noisy or ambiguous inputs, aleatoric uncertainty is more adequate from a theoretical perspective for misclassification detection. In the case of these inputs, each sampled model from the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$ will ideally

be uncertain about its prediction and have a uniform softmax probabilities. In practice, total uncertainty is more appropriate for detecting misclassified samples as its encodes both aleatoric and epistemic uncertainties, and hence marginally better than aleatoric uncertainty. When it is possible to measure the various sources of uncertainties, one should not rely on the (predictive) entropy as a score for OOD detection as it is "inherently inappropriate" (Kirsch et al., 2021) since a high value could result from an ambiguous input (high aleatoric uncertainty), and thus not an OOD sample. For OOD detection, total uncertainty is suboptimal from a theoretical perspective as it does not offer a proper distinction between OOD samples and (misclassified) ambiguous inputs (Kirsch et al., 2021). Therefore, OOD detection prevails over misclassification detection which must be then identified as ID samples (see Algorithm 1).

6.1.4 The simple case of Two Moons dataset

To illustrate the evolution of the different aforementioned scores that are relevant to misclassification and OOD detection, we will start by analyzing the classification dataset *two moons* [23], which consists of a binary classification of two interleaving half circles.

Inspired by the experimental setups in Liu et al. (2020a); de Jong et al. (2024), we train the same model architecture, an MLP with residual connections, under different approaches: MC-Dropout, DE, EDL, Conflictual DE and Conflictual Model. All these models were trained on an increasing number of datapoints in the training set (32, 64 and 128). In each case, once the model is trained, we will evaluate different scores on the entire two-dimensional grid, namely epistemic uncertainty (Figure 6.1), aleatoric uncertainty (Figure 6.2), energy (Figure 6.3) and predictive uncertainty (Figure 6.4). In each figure, we report the evolution of the score for each model and with the increasing size of the training set (on the y-axis). Additionally, we will plot the decision boundary defined as the BMA at 0.5.

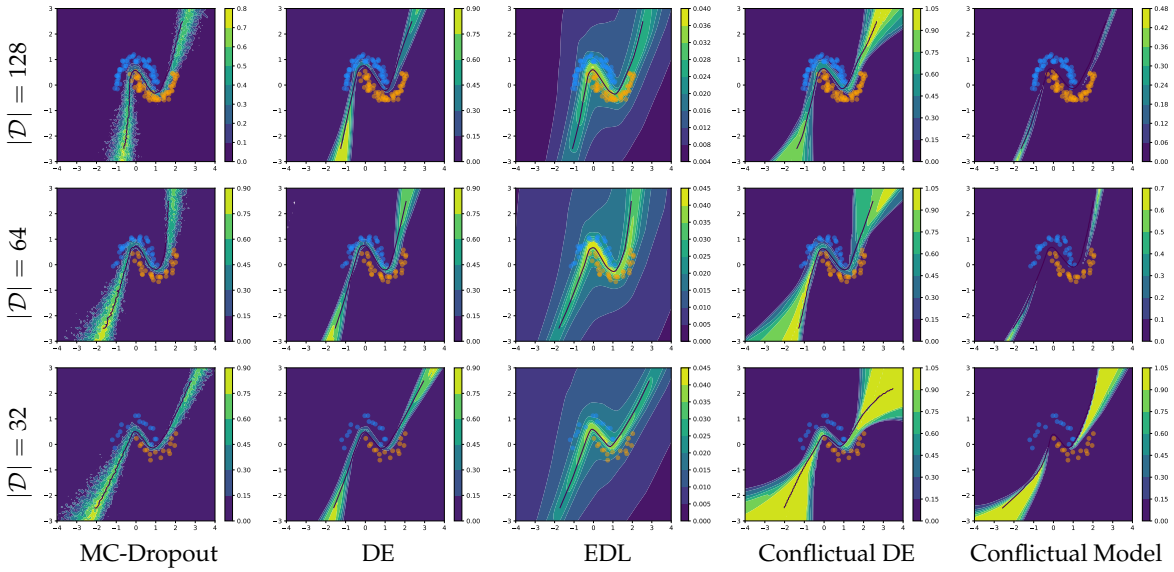


FIGURE 6.1: Epistemic Uncertainty. Each subplot uses a different colormap scale.

Epistemic uncertainty (Figure 6.1)**.** For the two-moons classification task, high epistemic uncertainty is expected to be around the decision boundary, further from the training dataset and slightly between the two clusters (especially in the low-data regime). By acquiring more data, the model becomes more certain about the decision boundary. By analyzing the results, MC-Dropout seems

[23] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html

the least inline with this expectation: although the decision boundary changes as the model is trained on more samples, these changes are not included in the epistemic uncertainty of the initial model trained on 32 datapoints. The same observation holds true for DE, especially in the top right zone. Furthermore, the width of the region with high epistemic uncertainty is more or less the same around the decision boundary in both models, regardless of size of the training set. In the case of EDL, epistemic uncertainty (more precisely, distributional uncertainty) is negligible, and it decreases as we move away from the decision boundary. Moreover, the use of Conflictual Loss leads to expected change progression of epistemic uncertainty on the grid: all the decision boundaries are within the epistemic uncertainty intervals of the model trained with the smallest training set ($|\mathcal{D}| = 32$). Finally, and except for EDL, epistemic uncertainty is lower between the two moons, and this is more pronounced for Conflictual Model. With epistemic uncertainty being the ideal measure for OOD detection, Conflictual DE and Conflictual Model are the ones who, by far, meet expectations, even though their epistemic uncertainty will rapidly drop to zero as one moves away from the training set.

Aleatoric uncertainty (Figure 6.2)**.** As the two clusters do not overlap, aleatoric uncertainty is expected to be high on the decision boundaries, and in between the two clusters (especially in the high-data regime). Generally speaking, aleatoric uncertainty is concentrated on the classification boundary, and it fades the further we move from the training data. Additionally, we notice that it increases between the two clusters as we train on more samples. With EDL, it is clear that aleatoric uncertainty is the dominant source of uncertainty.

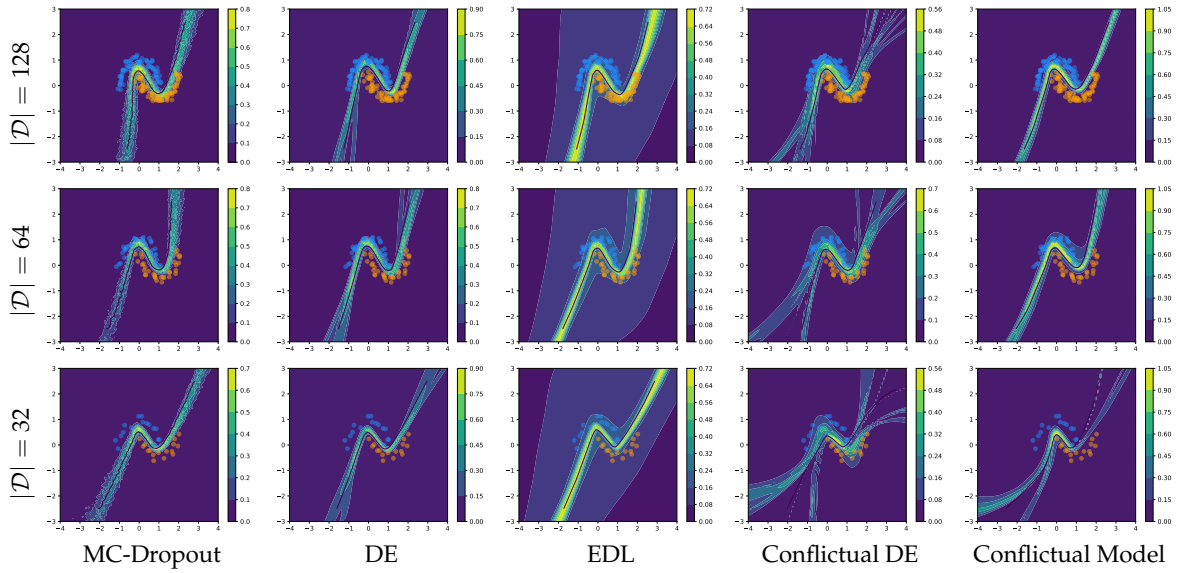


FIGURE 6.2: Aleatoric Uncertainty. Each subplot uses a different colormap scale.

Energy (Figure 6.3)**.** It is argued that energy is a good identifier for OOD detection, however, the results on the two moons dataset show that energy is more suitable to detect ambiguous samples, that are on the decision boundary, rather than OOD samples. While it appears that the energy achieves it maximum near the decision boundary and decreases as we move away from it, with more influence on the distance to the two clusters in the case of Conflictual DE, the energy score remains highly correlated with aleatoric uncertainty. When looking at the data on a different scale for the colormaps (by plotting the exponential of the energy rather the energy score, as illustrated in Figure E.1), we observe trends akin to those observed for aleatoric uncertainty, except for EDL where the energy is concentrated on the classification boundary inside the two moons.

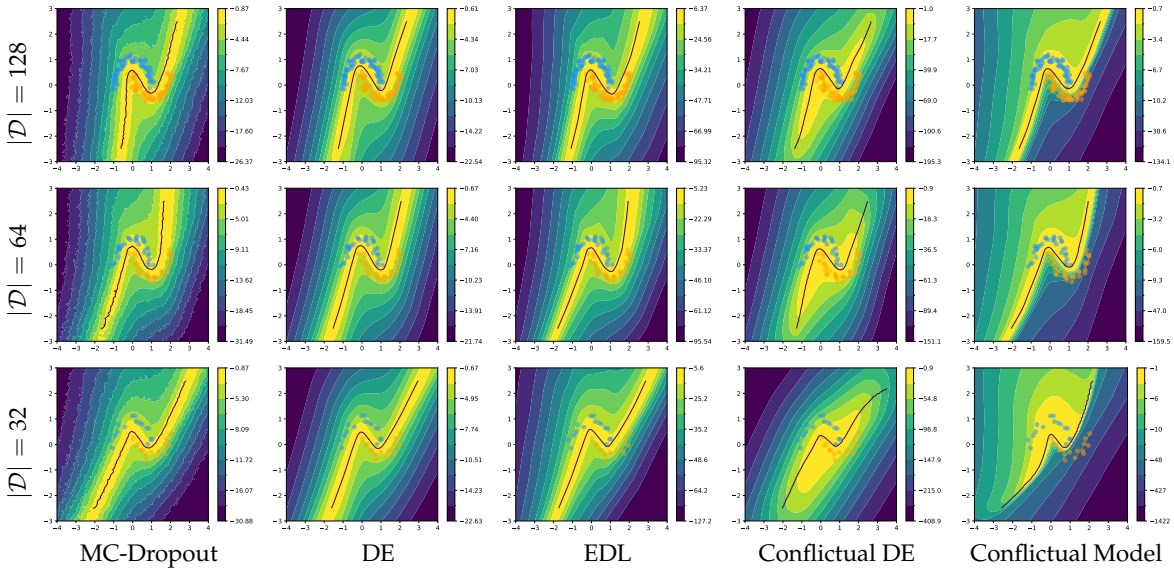


FIGURE 6.3: Energy. Each subplot uses a different colormap scale.

Predictive uncertainty (Figure 6.4). Being a measure of total uncertainty, predictive uncertainty should allow the detection of both unknown and incorrect samples, without explicitly distinguishing which is which. Computed on the models trained on the two moons dataset, we can clearly see that predictive uncertainty is high when epistemic uncertainty is high and also within the classification boundary near the training datapoints. Some work, such as DUQ (Amersfoort et al., 2020) and SNGP (Liu et al., 2020a), advocate for a measure of predictive uncertainty that is distance-aware, in the sense that the samples that are further from the training manifold, should be detected as OOD, according to some distance metric. For the two moons dataset and under this assumption, predictive uncertainty is expected to be low around the two clusters and high elsewhere. This is close to a GP with the RBF kernel, or even an SVM (Cortes and Vapnik, 1995) with the RBF kernel. While in general the decision boundary for a standard DL model is a continuous sinusoidal curve (Amersfoort et al., 2020; Valdenegro-Toro, 2021; de Jong et al., 2024), we believe that previous argument is complex to generalize in the context of DL. Finally, the MSP is reported in Figure E.2, and it is overall correlated visually with the predictive uncertainty.

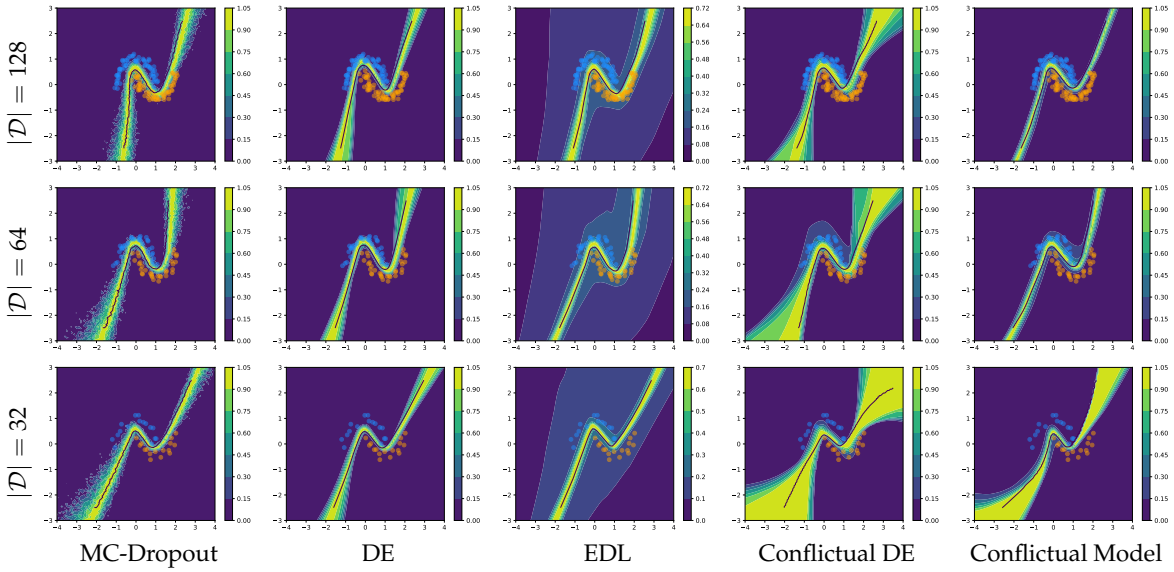


FIGURE 6.4: Predictive Uncertainty. Each subplot uses a different colormap scale.

6.1.5 A dual-dimensional exploration

In this section, we will further develop the dual-dimensional analysis initiated in Chapter 4 and Chapter 5 with a focus on misclassification and OOD detections. Thus, these tasks will be evaluated on the models trained in these chapters. To this end, FashionMNIST is used as an OOD dataset for the models trained on MNIST, whereas we use SVHN (respectively CIFAR10) as an OOD dataset for the models trained on CIFAR10 (respectively SVHN). Although restrictive due to the infinite number of choices for OOD datasets, we will adhere to this benchmark owing to its widespread application in the literature (Lee et al., 2018a; Ren et al., 2019; Wang and Aitchison, 2021; Everett et al., 2022; Nguyen et al., 2022; D'Angelo and Henning, 2022; Yang et al., 2022). To quantify the capabilities of the detection, we will rely on the AUC-ROC metric for a binary classification to distinguish between the correctly classified (negative class) and the misclassified (positive class) inputs, or between the ID (negative class) and OOD (positive class) samples.

Misclassification detection. We can see in Figure 6.5 that aleatoric uncertainty is a reliable metric to identify misclassified samples. Perhaps from a practical perspective total uncertainty (Figure B.4) could perform slightly better than aleatoric uncertainty. Regardless of the model and the dataset, the AUC-ROC tends to increase as the model is trained on more samples. The use of epistemic uncertainty to detect the incorrect samples is inadequate, even in practice, as shown in Figure B.5, and hence further advocating for the shortcomings of using it for this task. It leads to inferior detection compared to aleatoric and total uncertainties, and could assign high (respectively low) values to correctly classified (respectively misclassified) inputs, similar to MC-Dropout with LS.

OOD detection. In the same spirit, we evaluate the abilities of these models to detect unknown samples. The empirical findings support the theoretical discussion in the previous section. In fact, total uncertainty is not valid overall for OOD detection as illustrated in Figure B.6 and even worst if one relies on aleatoric uncertainty Figure B.7. Furthermore, as the contradictions do not appear for MNIST, it could explain the justifications of the use of total or aleatoric uncertainties for this identification. However, in CIFAR10, and more critical in SVHN, these uncertainties confuse ID and OOD samples. On the other hand, epistemic uncertainty provides a better separation of ID and OOD samples as demonstrated in Figure 6.6, except for MC-Dropout with LS and EDL. For the former, we believe that the collapse is attributed to this model being the least epistemically calibrated (as outlined in Table 4.1). For the latter, we notice that the patterns in the SVHN heatmaps are similar across the three uncertainties, and a comprehensive understanding has yet to be established. Overall, while DE (LS) seems to achieve the best OOD detection based on epistemic uncertainty, Conflictual DE closely follows as the second-best.

Algorithm 1: OOD and misclassification detection

Inputs: Sample x, Model $\mathcal{M}$ trained on $\mathcal{D}$, Thresholds τ_{OOD}, τ_{MIS}.

```
/* OOD detection - Epistemic uncertainty */
```
if $\mathcal{U}^{e}_{\mathcal{D},\mathcal{M}}(x) > \tau_{OOD}$ **then**
  return *OOD*
end

```
/* Misclassification detection - Predictive uncertainty */
```
if $\mathcal{U}^{t}_{\mathcal{D},\mathcal{M}}(x) > \tau_{MIS}$ **then**
  return *misclassified ID*
else
  return *correctly classified ID*
end

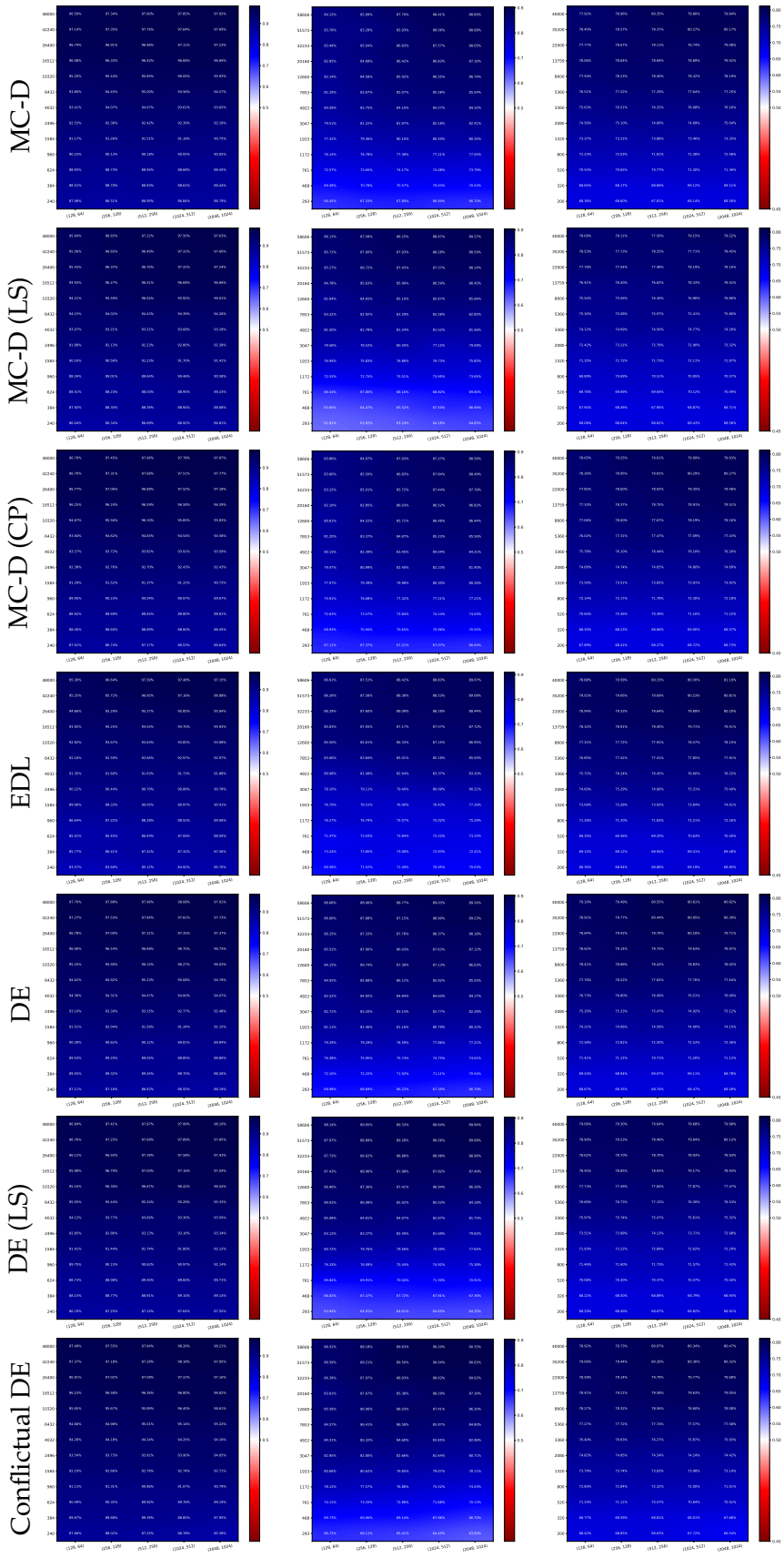


FIGURE 6.5: Misclassification AUC-ROC based on aleatoric uncertainty, with the same representation as in Figure 4.4. Heatmaps normalized per dataset (column).

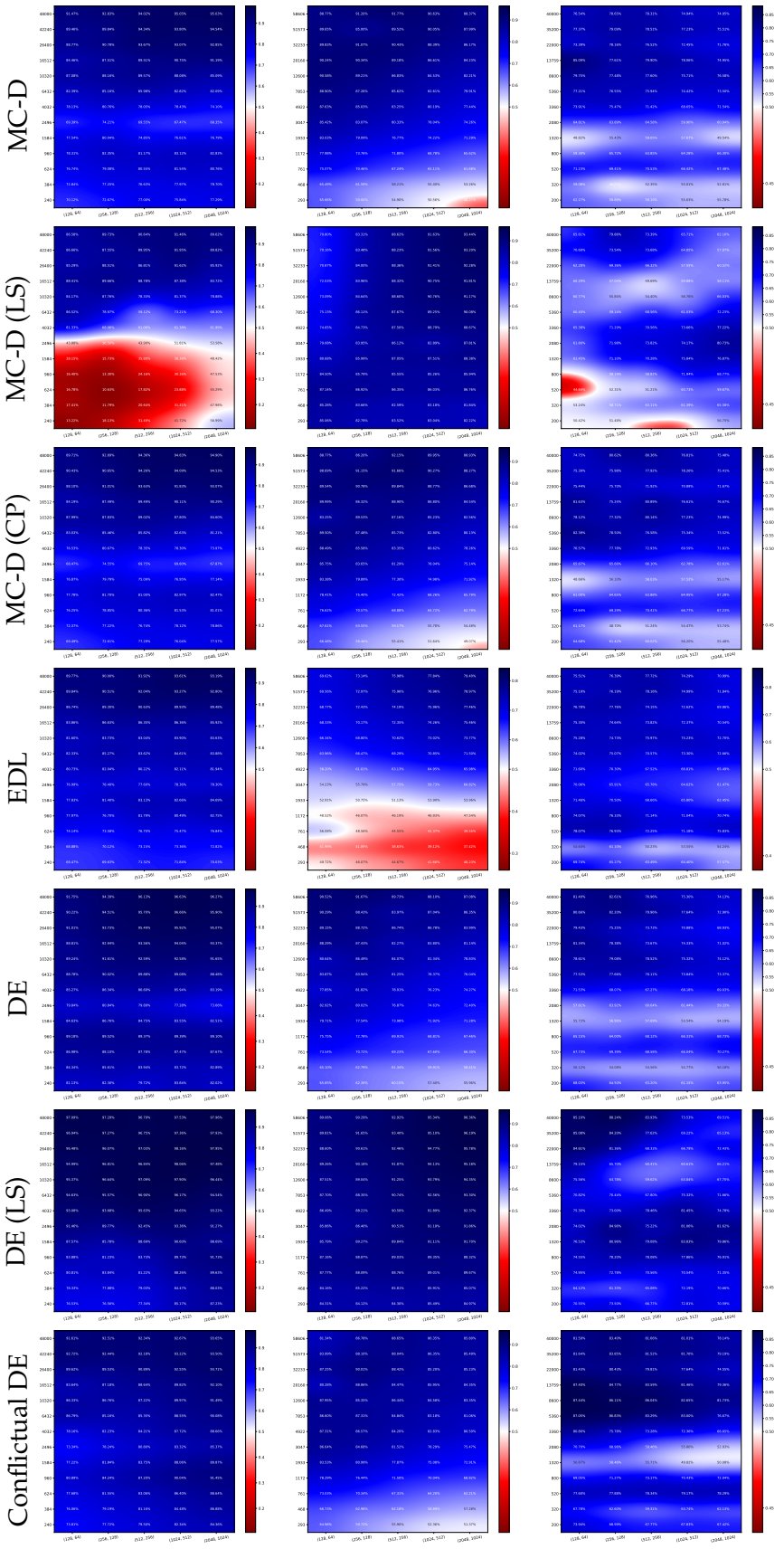


FIGURE 6.6: OOD AUC-ROC based on epistemic uncertainty, with the same representation as in Figure 4.4. Heatmaps normalized per dataset (column).

Detection protocol. Based on the previous discussion, it is possible to address the detection of the misclassified and OOD samples in an iterative approach. Given a model $\mathcal{M}$ trained on $\mathcal{D}$, the goal is to classify a (test) sample x as OOD, misclassified ID or correctly classified ID. First, we start by measuring epistemic uncertainty $\mathcal{U}^{e}_{\mathcal{D},\mathcal{M}}(x)$ and compare it to a threshold τ_{OOD} to determine whether x is an OOD sample. The threshold τ_{OOD} is fixed based on the validation ID samples. If x has a low epistemic uncertainty, it is thus likely an ID sample, and we can classify it as misclassified or correctly classified by comparing its predictive uncertainty $\mathcal{U}^{t}_{\mathcal{D},\mathcal{M}}(x)$ to a predefined threshold τ_{MIS} based as well on the validation dataset. The approach is summarized in Algorithm 1.

6.1.6 Ambiguous-MNIST

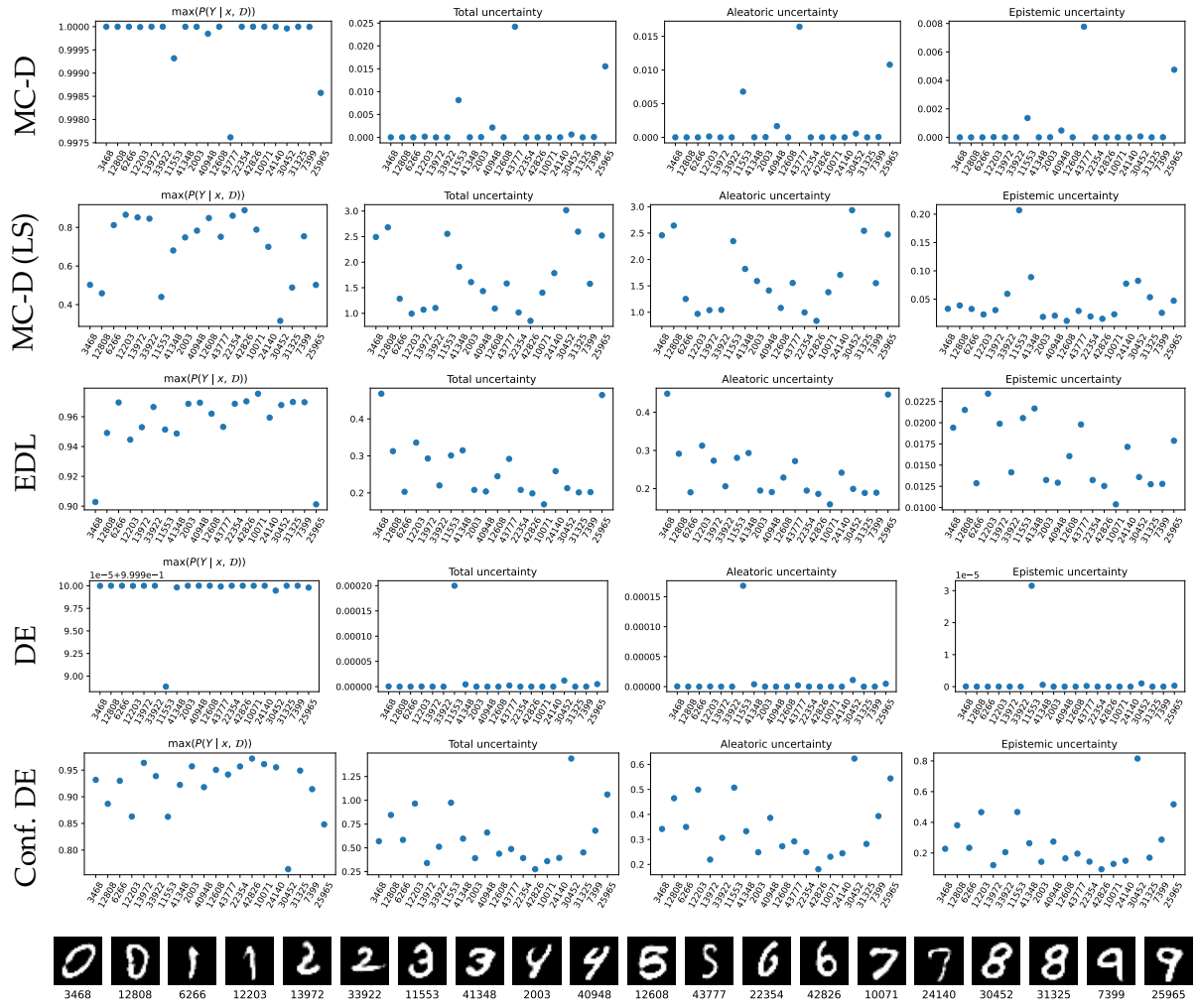


FIGURE 6.7: Results on "clean" inputs from Ambiguous-MNIST. The integers under the images represent their indices in the dataset. The y-axes are scaled differently in each subplot.

In Figure 4.16 and Figure 5.10, we highlighted some contradictions related to the assumption that aleatoric uncertainty should be high for Ambiguous-MNIST (Kirsch et al., 2021). While Ambiguous-MNIST, as introduced by Mukhoti and Kirsch (2023), aims at evaluating the calibration of aleatoric uncertainty as its samples are noisy MNIST images, by using the entire introduced set, we believe that the selected samples in our evaluation are closer to OOD samples than to noisy ID samples, thus partially explaining the uncertainty being mostly epistemic. We expect high aleatoric uncertainty for noisy samples or images where class separability is hard. However, as aforementioned, the tested images were picked for their difficulty in being recognized as human,

raising the question whereas we can consider these specific images as noisy images or perhaps OOD samples.

This hypothesis is confirmed in Figure 6.7 as the models are tested on samples from Ambiguous-MNIST that look more like MNIST data: these images are recognizable by human and appear to have a negligible amount of noise. Although MC-Dropout, EDL and DE are confident in their predictions, which are similar to ID samples, we notice a slight increase of aleatoric uncertainty with Conflictual DE with confident predictions, as can be seen on $\max\left(\mathrm{p}(Y \mid x, \mathcal{D})\right)$. However, the results with MC-Dropout with LS are the most surprising on the tested unambiguous inputs. In fact, LS makes the model incorrectly underconfident and artificially increases aleatoric uncertainty. Finally, all the models are correctly predicting all the digits.

Hence, we are a bit skeptical about the use of Ambiguous-MNIST to assess aleatoric uncertainty as the noise in the generating process makes some inputs completely unrecognizable, such as the ones used in Figure 5.10 which do not resemble the training set, or completely similar to the ones in MNIST, as the ones shown in Figure 6.7. After a careful examination of the dataset, we noticed redundancy in the dataset such that each image is repeated 10 times: for a given i, all the images associated to the indices $\{10 \times i + j\}_{j \in \{0,\dots,9\}}$ are very similar, if not identical. This raises the question on the adaptivity of these samples to assess aleatoric uncertainty, as it is the goal with Ambiguous-MNIST dataset. It is important to mention that the tested samples are not in any case a special case of the images in Ambiguous-MNIST but are rather fairly represented in the dataset.

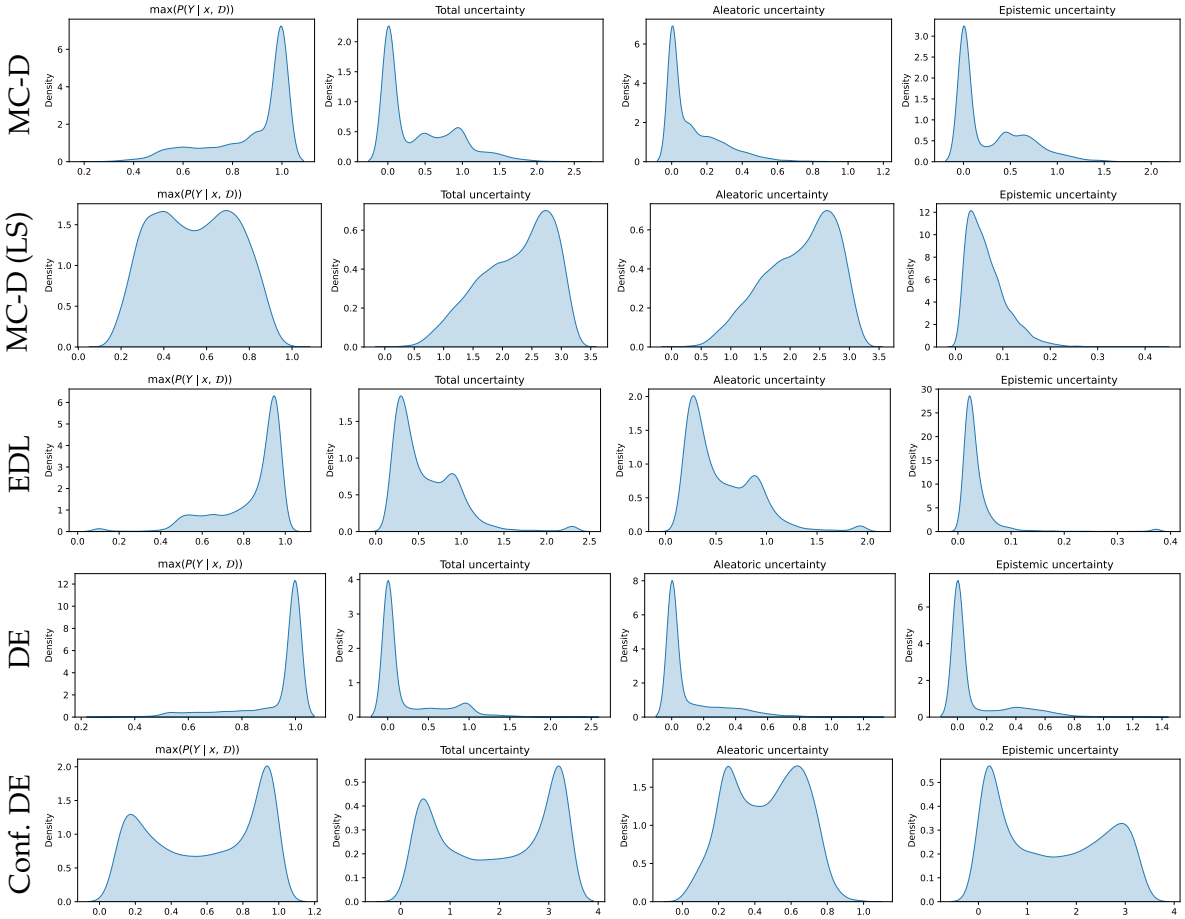


FIGURE 6.8: Histograms of different scores on the non-redundant subset of the test set of Ambiguous-MNIST, with the same models as in Figure 6.7. The limits on the x-axes and y-axes are not normalized.

We take one final look at Ambiguous-MNIST on its entirety. With the same models as those used in the micro analysis, we visualize the histograms of different scores on the non-redundant subset of Ambiguous-MNIST, which is determined by selecting only the images corresponding to indices that are multiples of 10. The results in Figure 6.8 confirm our previous discussion: Ambiguous-MNIST consists of easy-to-classify noisy samples and of OOD-like samples. For each model, we notice that at least one score shows two distinct peaks associated to these two clusters of examples. Besides, Conflictual DE is the only model allowing a full distinction of these two modes. The results with DE are the least desirable as the model is confident in its predictions, and it gives low uncertainties. It is important to mention that the dual modality is unique to Ambiguous-MNIST and does not occur with the test set of MNIST for which all the models are significantly confident, as illustrated in Figure 6.9, even with Conflictual DE.

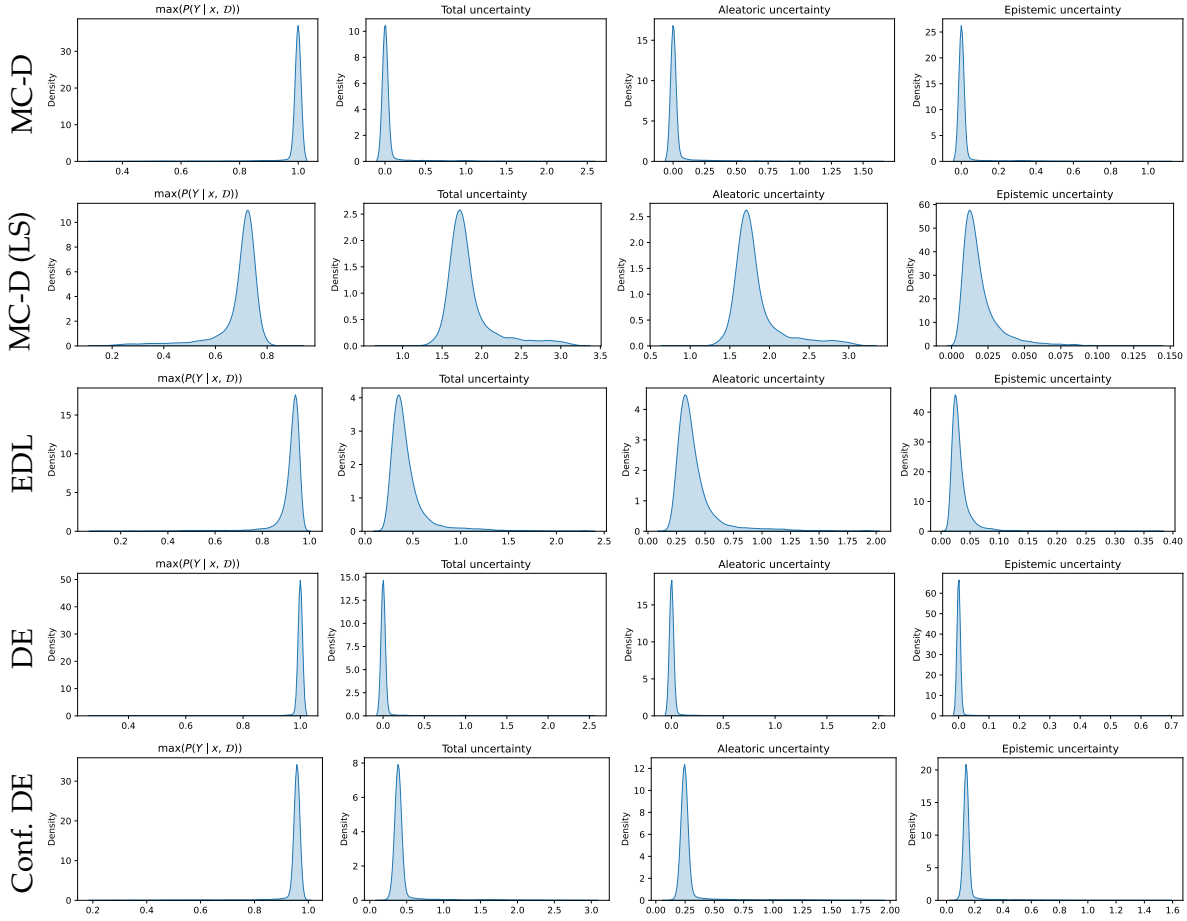


FIGURE 6.9: Histograms of different scores on the test set of MNIST with the same models as in Figure 6.8. The limits on the x-axes and y-axes are not normalized.

6.2 Active Learning

Up to this point, we dealt with classical classification tasks: given a model and a training dataset, we are interested in the performance on the test set. Even when we evaluated the effect of the size of the training set as part of the data-related principle, all the splits were deterministic and defined beforehand. In active learning (AL) (Settles, 2012) however, the (labelled) training data evolves as new samples are added from a large unlabeled pool of data after being labelled by an oracle. The use of an oracle in active learning is driven by the high cost of labelling, which demands considerable time, monetary resources, and expert knowledge. AL is thus a strategy where a

model selectively queries the most informative data points for labelling, aiming to achieve high performance with minimal labeled data. Each step of the active learning loop consists of training the model on the available labelled samples, and selecting a batch of k unlabeled datapoints which are then labelled by the oracle. In the terminology of AL, k is referred to as *query size* or *acquisition size*, and the selection method of the queries is referred to as *query method* or *acquisition function*.

In addition to the known challenges associated with the model selection and the training pipeline, the AL loop brings even more constraints, making the task extremely challenging. Arguably, the most impactful component is the choice of the query method as it defines how the pool of the unlabeled samples is explored. Ideally, the selected queries should be the most informative and yield the highest performance gain compared to any other possible batch of datapoints. Acquisition function could either be more specific for single-acquisitions or more general for multi-acquisitions. We will discuss in the following some heuristics for the acquisition function that are widely used in AL and Bayesian AL.

6.2.1 Single-acquisitions

BALD. (Bayesian Active Learning by Disagreement) (Houlsby et al., 2011) relies on epistemic uncertainty of the model, as computed using the mutual information, and applied to Gaussian processes. As introduced by Houlsby et al. (2011), BALD selects only the sample for which the model disagrees the most. Additionally, we can interpret this choice as exploring new areas in the inputs space. These unexplored areas could be considered as OOD at the current step of the AL loop, thus the use of epistemic uncertainty is ideal (Hüllermeier and Waegeman, 2021). Moreover, Gal et al. (2017) combines MC-Dropout with BALD, allowing efficient AL for high dimensional inputs, and making its use practical for any model with Dropout layers. Its efficiency was tested in (Atighehchian et al., 2020) on realistic datasets and shown to be a strong baseline. At each step of the loop, Gal et al. (2017) suggest resetting the model and training from scratch to solely quantify the performance of the acquisition function and the informativeness of the selected batch of data. We will use the same experimental protocol in our experiments (Section 6.2.4).

DEAL. Despite their limited popularity, EDL models can also be applied in framework of active learning, such as DEAL (Deep Evidential Active Learning) (Hemmer et al., 2020). Although Malinin and Gales (2018) suggest that distributional uncertainty can be an appropriate acquisition function, DEAL is based on the minimal margin measure (Wang et al., 2017) as it provides better empirical results. In a nutshell, the minimal margin measure consists of computing the difference between the probabilities of the first and second most probable classes, and it can be seen as a proxy for aleatoric uncertainty (the higher the difference, the lower the uncertainty). Compared to BALD and BatchBALD (see Section 6.2.3), the starting labelled set and the query size is much higher in the experiments conducted in (Hemmer et al., 2020).

6.2.2 Pitfalls and Challenges of Active Learning

Due to flawed methodologies and to the lack of standardized protocols, Lüth et al. (2023) demonstrate the limitations and difficulties of effectively evaluating AL methods from a deployment perspective. More precisely, they identified five pitfalls that contribute to the inconsistencies in the evaluation: (1) lack of evaluated data distribution settings, (2) lack of evaluated starting budgets, (3) lack of evaluated query sizes, (4) neglection of the classifier configuration, and (5) neglection of alternative training paradigms. We will take a closer look at the second and the third pitfalls.

Starting budgets. When starting the AL loop with a small labelled set, the batch selection may underperform, leading to misleading conclusions about the robustness of the acquisition function.

This phenomenon is often referred to as *Cold start problem* (Maltz and Ehrlich, 1995; Schein et al., 2002; Houlsby et al., 2014; Konyushkova et al., 2017; Gao et al., 2020), and is mostly relevant when the task is complex and/or the data exhibits significant variability. Semi and self supervised techniques could make efficient use of the large pool of unlabeled data and elevate the cold start problem, we refer to Lüth et al. (2023) and references therein.

Query size. Compared to Houlsby et al. (2011), the reported results in (Gal et al., 2017) use a query size $k > 1$ where the queries are the top-k samples with the highest acquisition score. The choice of the query size is quite tough and arduous. On one side, with a larger batch of queries, DL models experience more substantial benefits in terms of learning and performance. Additionally, the cost associated to the AL loop will reduce as fewer trainings of the model are required. On the other, when the query size is large, the queries could end up being selected from the same unexplored region and thus adding a suboptimal batch that contains redundancy. Imagine the simple example of CIFAR10 and assume that all the images in Figure 6.10 are in the unlabeled pool. If at an iteration of the AL loop one of the images in Figure 6.10 has the highest epistemic uncertainty, it is likely that epistemic uncertainty for the remaining images in the same figure is also high, leading to their selection, and hence the batch will be of redundant samples. Furthermore, in a realistic setup, labelling a large batch might be infeasible, as it is challenging to secure an adequate number of experts, and the potentially high costs involved in the labelling. The subfield of selecting an informative batch rather than a single input is referred to as *batch active learning*.

FIGURE 6.10: Similar images from the train set of CIFAR10 with the same label (1). The integer under each image represents its index in the train set of CIFAR10.

6.2.3 Batch Active Learning

BatchBALD. By selecting the top-k samples with BALD, their informative redundancy is not taken into account. The use of the top-k operator is equivalent to selecting the k samples that maximize the sum of the score (epistemic uncertainty in this case). Kirsch et al. (2019) propose BatchBALD which is more suited for batch AL as it relies on the joint mutual information rather than the sum of individual mutual information scores. With the consideration of the joint mutual information in BatchBALD, the dependency of the queries is acknowledged, given rise to an efficient and informative batch. It is essential to recognize that in the reported results in (Kirsch et al., 2019), as the query size increases, the performance with both BALD and BatchBALD declines, especially with the former.

BADGE. Another strong baseline for batch AL is BADGE (Batch Active learning by Diverse Gradient Embeddings) (Ash et al., 2020). The goal of BADGE is similar to BatchBALD in the

sense that they both aim at selecting the most informative and distinct queries. BADGE uses the gradient vector (embeddings) of the loss with respect to parameters in the final layer as a proxy for uncertainty, and k-means++ (Arthur and Vassilvitskii, 2006) to capture diversity in the batch. To compute the gradient, Ash et al. (2020) choose to rely on the predicted class by the model, even though it could be erroneous. This choice is justified as they show that this gradient's norm lower bounds the gradient norm computed by any other label. Furthermore, the seeding algorithm k-means++ ensures diversity in the batch, as it approximates the k-means objective function in expectation

Stochastic Batch Acquisition. Both BatchBALD and BADGE allow a diverse and informative batch selection, yet they are computationally expensive, limiting the query size. Kirsch et al. (2023) introduce randomness into the scores, namely BALD, for a stochastic selection, thanks to Gumbel noise and the Gumbel-max trick (Gumbel, 1954; Maddison et al., 2015; Huijben et al., 2022). As the ranking of unlabeled samples changes over the iterations of the AL loop, and under the assumption that the future scores are a perturbed version of the current scores, the use of a stochastic acquisition function will counterbalance the imprecision of the score. The randomness introduced by the Gumbel noise means that we are not simply selecting the largest values of the original noisy scores, but rather selecting the scores that would be ranked the highest if we account for the additional randomness. Three acquisitions were defined in (Kirsch et al., 2023): Soft-Rank, Softmax and Power, where the Gumbel noise is added to the log rank of the scores, the scores and the log scores respectively. These stochastic acquisitions outperform BALD, and have comparable results to complex acquisition functions such as BatchBALD and BADGE, with no computational overhead.

Remark *(Query size)*. Lüth et al. (2023) highlighted that, in theoretical (mainly information-theoretical) papers, a small query size is often used, contrary to practical papers which have additionally a large starting budget.

6.2.4 Experiments

In this section, we would like to test the effect of a calibrated epistemic uncertainty on the Bayesian active learning loop. To this end, we will investigate the performance at each step from various perspectives, in contrast to the literature, where results are typically reported based solely on accuracy. These metrics provide complementary information about the effectiveness of the batch acquisition and the used Bayesian model. Some of the additional metrics that we will analyze in the context of active learning are: Brier score, ECE, misclassification based on epistemic and predictive uncertainties.

By using the same convolutional base model, we assess the performance of active learning on a simple task (MNIST) and a more complex one (CIFAR10). The goal is to further illustrate the impact and the informativeness of the selected batch, and we expect the differences to be more pronounced in the case of CIFAR10 due to its difficulty. Moreover, we test two acquisition functions: epistemic uncertainty and predictive uncertainty. While the former is considered the ideal measure for Bayesian active learning, the use of the latter is of interest since it is an upper bound for epistemic uncertainty and given the absence of aleatoric uncertainty in these two datasets.

Experimental setup. After making sure the same labeled and balanced pool will be used to train all the models (20 for MNIST, 200 for CIFAR10), we set the batch sizes to 10 and 50, respectively, and train until reaching a labeled pool of 500 and 1500 samples, respectively. Furthermore, at the beginning of each step, the model is randomly initialized, discarding the trained weights

from the previous step. Gal et al. (2017) motivate resetting the model in order to emphasize the informativeness of the selected batch and reduce the impact of the previously trained model. We test MC-Dropout, DE and Conflictual DE with two values for α: 0.5 and 1. We note that the experiment was also conducted using smaller batch sizes for comparison (5 for MNIST and 20 for CIFAR10). Given that this did not lead to performance gains, the results reported are based on the larger batch sizes with each model being trained with 5 different random seeds for reliability. We report the mean and the standard deviation in the plots (Figures 6.11, 6.12, 6.13 and 6.13).

Expectations. At the end of each step in the AL loop, we expect the trained model to yield better performance on the test set. As the model is trained on additional informative datapoints, its ability to generalize is expected to improve with each iteration of the loop. Therefore, the accuracy is likely to increase while Brier score and ECE are supposed ideally to decrease. Moreover, since the batch selection aims at finding the most uncertain samples in the unlabeled pool, a direct consequence of the improved generalization is the progressively better capacity at distinguishing correctly and incorrectly classified samples. Hence, the AUC score for misclassification based either on epistemic or predictive uncertainties will likely increase.

Active learning applied to MNIST

Under the experimental setup highlighted previously, we aggregate the results on the test sets for the models trained on MNIST relying on epistemic uncertainty (Figure 6.11) and predictive uncertainty (Figure 6.12) as acquisition functions. Overall, the results show considerable variability for both acquisition functions, particularly when fewer than 300 datapoints are labeled. This variability is more pronounced with MC-Dropout, likely due to the inherent randomness of the dropout masks.

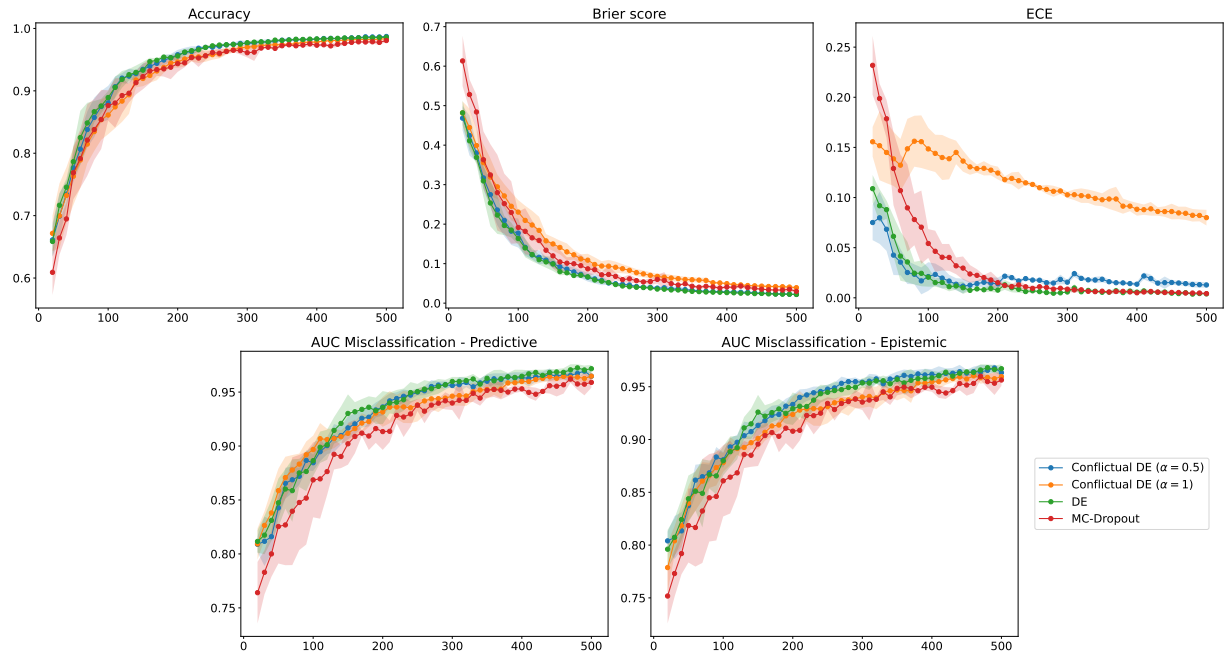


FIGURE 6.11: The evolution of different metrics at each iteration of the AL loop for a selection based on epistemic uncertainty for MNIST. The results are reported for 5 different runs.

Epistemic selection. When exploring the unlabeled pool based on epistemic uncertainty (Figure 6.11), we notice that Conflictual DE with $\alpha = 0.5$ performs similarly to DE. Moreover, the

metrics behave overall as expected, with a few exceptions for Conflictual DE with $\alpha = 1$. In fact, the most striking observation is perhaps the ECE score as it decreases slowly compared to the other compared models, and the accuracy is lower than with $\alpha = 0.5$ or DE. Yet, at the first iterations, the AUC score for misclassification detection based on predictive uncertainty is relatively better than the tested models.

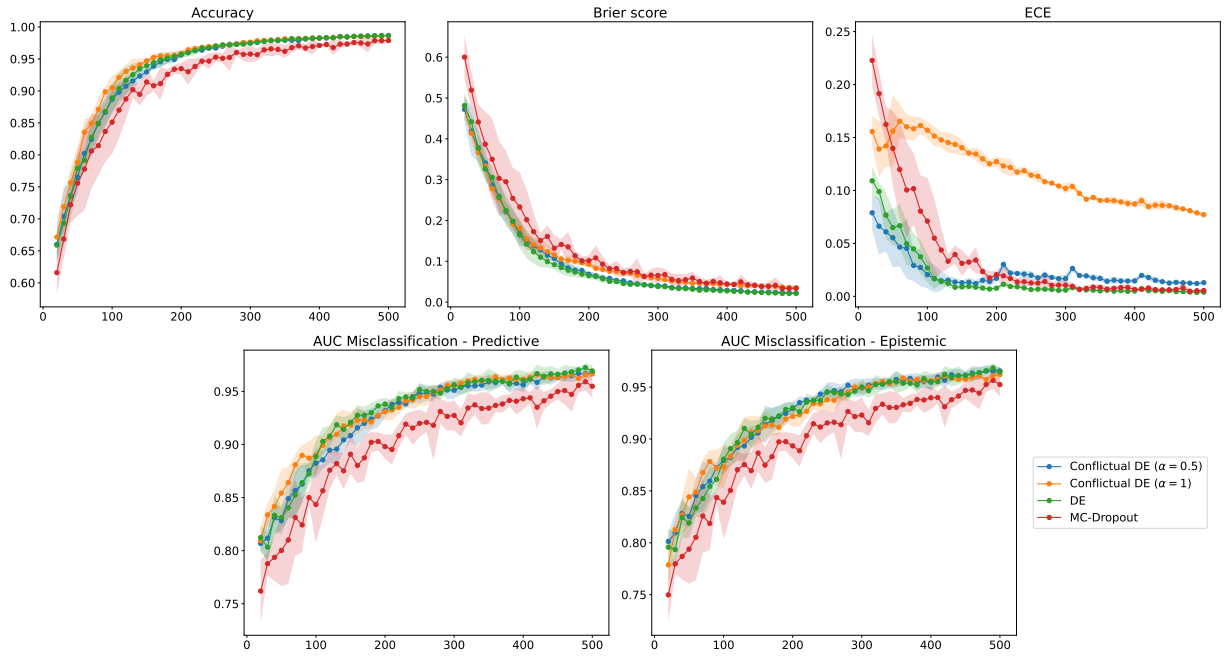


FIGURE 6.12: The evolution of different metrics at each iteration of the AL loop for a selection based on predictive uncertainty for MNIST. The results are reported for 5 different runs.

Predictive selection. The use of predictive uncertainty for Conflictual DE with $\alpha = 1$ is shown to yield the best accuracy curve (Figure 6.12) for all the models and the two acquisition functions, in addition to an improvement of Brier score for this model, while the other metrics are comparable to those obtained in the case of epistemic uncertainty based acquisition. As with using epistemic uncertainty for batch selection, the AUC score for misclassification based on epistemic uncertainty is highly correlated with that based on predictive uncertainty.

Active learning applied to CIFAR10

Due to the relative simplicity of the MNIST classification task, for which we achieve easily a 95% accuracy with only 200 to 300 datapoints, we extend the comparison to a more complex task, CIFAR10, to further illustrate the impact of the selection. Similar to MNIST, we start by evaluating the selection based on epistemic uncertainty (Figure 6.13) and then based on predictive uncertainty (Figure 6.14). Furthermore, comparable to the MNIST curves, the variability in regard to the different random seeds is high, especially for AUC scores.

Epistemic selection. When analyzing the use of epistemic uncertainty as an acquisition function, we notice that at least one of the tested Conflictual DEs is the best for the different tracked metrics. However, the ECE for Conflictual DE with $\alpha = 1$ exhibits unexpected behavior being not a decreasing function. While we do not have a clear understanding of this phenomenon, it could result from the hyperparameter λ (Equation 5.12) being a tradeoff between epistemic and aleatoric calibration, especially since it is around the same value as in the case of MNIST. This

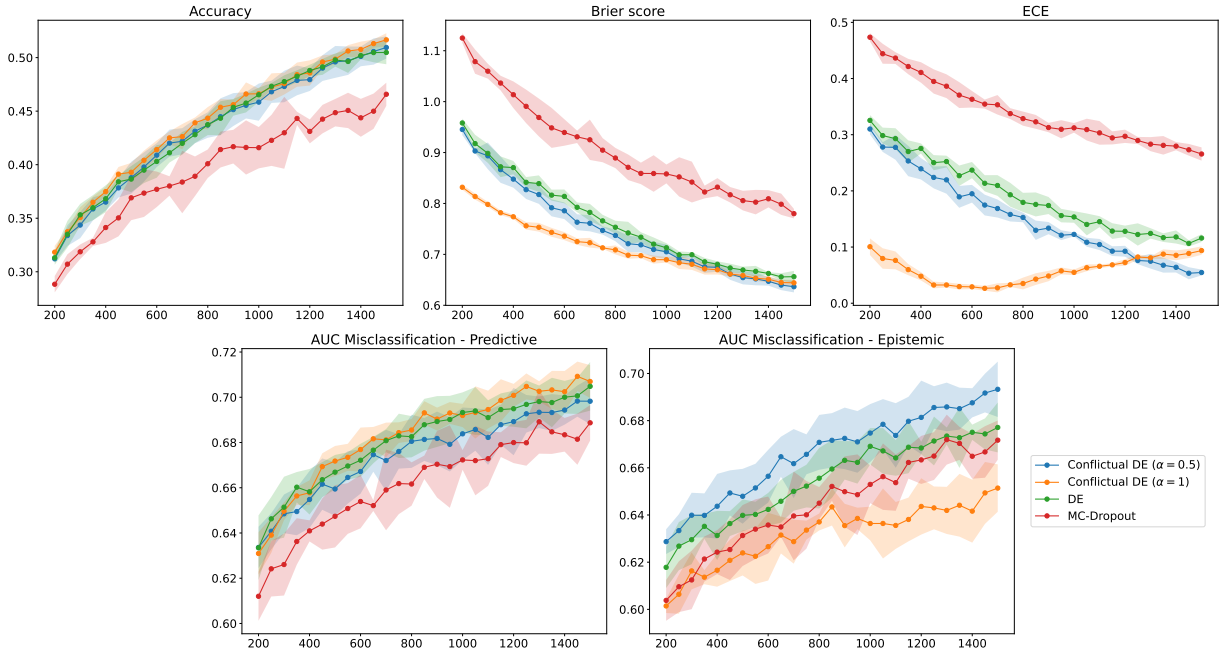


FIGURE 6.13: The evolution of different metrics at each iteration of the AL loop for a selection based on epistemic uncertainty for CIFAR10. The results are reported for 5 different runs.

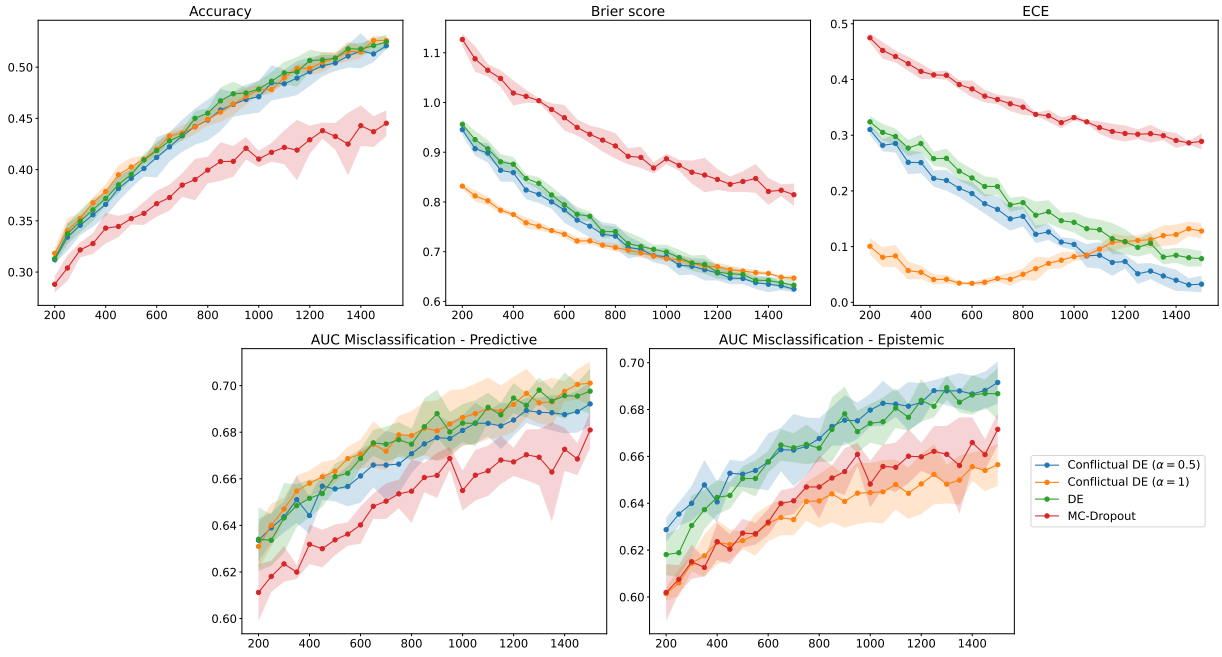


FIGURE 6.14: The evolution of different metrics at each iteration of the AL loop for a selection based on predictive uncertainty for CIFAR10. The results are reported for 5 different runs.

hyperparameter may establish a lower bound for the ECE score, as BMA is encouraged to assign non-zero probabilities to all classes which can lead to higher ECE due to reduced confidence even when predictions are correct.

Predictive selection. Compared to the selection based on epistemic uncertainty, the results with predictive uncertainty (Figure 6.14) are quite comparable with the same ranking overall across all the metrics. The major difference perhaps is that we achieve comparable accuracy slightly faster with predictive uncertainty than with epistemic uncertainty. For instance, a 50% accuracy is achieved using 1200 datapoints with the former, whereas it needed 1400 with the latter.

For both the tested datasets, we notice that a selection based on predictive uncertainty slightly outperforms the one based on epistemic uncertainty. This is partially due to the fact that predictive uncertainty is an upper bound of epistemic uncertainty, and these datasets have negligible aleatoric uncertainty. Additionally, when comparing the curves of the AUC scores for misclassification in the case of Conflictual DE, we observe the superiority of $\alpha = 1$ when relying on predictive uncertainty and of $\alpha = 0.5$ when the detection is based on epistemic uncertainty. As aforementioned, as $\alpha \leq 1$ approaches zero, the DE is more diverse leading to a higher epistemic uncertainty, making epistemic uncertainty dominant in the total uncertainty. Finally, with no surprises, MC-Dropout is the least performing model in the previous experiments being the only single model.

6.3 Conclusions

To conclude, the two commonly tested applications for epistemic uncertainty, OOD detection and AL, show the impact and the necessity for a reliable and calibrated measure of epistemic uncertainty. For example, when we analyzed the dataset AmbiguousMNIST (Figure 6.8), only an epistemically calibrated model allowed the distinction of the clear and ambiguous (which are closer to OOD samples than to noisy MNIST samples) examples from an uncertainty perspective. Both on two moons dataset and the two-dimensional analyses, the use of Conflictual loss is shown to be beneficial. On the former, we notice that it results in a better verification of the data-related principle in the ensemble and single model versions. On the latter, Conflictual DE performs similarly, if not better, than plain DE in the three tested datasets across the two dimensions. Prominent results are shown in the active learning experiments, especially on the hardest task CIFAR10. Not only the selection benefited from the calibrated epistemic uncertainty from the performance point of view. Since the tested datasets are noise-free, it would be valuable to explore active learning on more complex and challenging datasets that better reflect real-world applications. On such applications, the benefits of a calibrated epistemic uncertainty are expected to be more noticeable.

CHAPTER 7
CONCLUSIONS AND PERSPECTIVES

"Don't worry about doing research. Just search."
Austin Kleon

7.1 Summary of Contributions

In this thesis, the focus was on epistemic uncertainty as measured by BNNs, and more precisely on its calibration. While the calibration of aleatoric uncertainty is a well-studied and a well-defined problem, the calibration of epistemic uncertainty on the other hand lacks a rigorous definition. One important factor on the difficulty of establishing a clear formalism for the epistemic uncertainty calibration is the subjectivity of choosing the prior distribution in the Bayesian framework, especially given the influence of this choice on the calibration. Furthermore, this dependency makes evaluating the calibration of a given model highly challenging. In the following, we will summarize the main contributions of the thesis.

7.1.1 Theoretical principles

One of the important parts in this thesis is the definition, in Chapter 4, of the two fundamental principles: the *data-related* and the *model-related* principles. These fundamental properties ensure necessary conditions regarding the evolution of epistemic uncertainty. On one hand, the data-related principle formally states the non-increasing assumption of epistemic uncertainty when more data becomes available. Especially, this principle is valid theoretically for epistemic uncertainty as measured by the mutual information. On the other hand, the model-related principle formulates the non-decreasing property of epistemic uncertainty as a function of the "complexity" of the model, framed through the notion of submodels. Importantly, these principles are only true under certain conditions, such as the *i.i.d.* assumption of the samples.

The definition and verification of these principles make the prior distribution absolute, and therefore the study of the epistemic calibration becomes objective. For instance, by adding more samples to the training set, we expect epistemic uncertainty to decrease as the model has acquired more knowledge and is more certain about its parameters, regardless of the prior distribution. Additionally, a BNN with more parameters requires more datapoints to reach the same confidence about the predictions compared to a smaller (submodel) BNN. Moreover, due to the challenge of establishing a ground truth for epistemic uncertainty, it is more practical to focus on relative changes rather than absolute values.

7.1.2 Empirical paradoxes

When evaluating the commonly used BNNs, such as deep ensemble, MC-Dropout and evidential deep learning, we noticed in Chapter 4 that they do not have a calibrated epistemic uncertainty, since they do not verify the fundamental principles. We referred to this phenomenon as the *epistemic uncertainty hole*. This evaluation was possible thanks to a dual-dimensional experiment where both the size of the training set and the model's capacity are scaled. More precisely, the lack of epistemic calibration is mostly noticeable for the model-related principle as epistemic uncertainty collapses for larger models. In addition, we also showed that the epistemic uncertainty hole exists with more performant models, when only the data-related principle is evaluated.

These paradoxes make the measure of epistemic uncertainty unreliable since it fails to adhere to the fundamental principles. As a result of this failure, one could question the use of the mutual information for example as an adequate measure for epistemic uncertainty, and/or the uncertainty disentanglement. In a nutshell, our arguments are that mutual information is a valid measure of epistemic uncertainty as it satisfies the fundamental principles, and that the tested models are epistemically uncalibrated, justifying the observed paradoxes. Therefore, it remains important to explore to what extent it is possible to incentivize the model to be epistemically calibrated.

7.1.3 Proposed solutions

In order to mitigate the lack of epistemic calibration in deep ensembles, as noticed in the tested models, we introduced in Chapter 5 Conflictual loss which leads to Conflictual DE when applied to a deep ensemble. Focusing on the loss function is of interest as it encourages the model during the training to have a calibrated epistemic uncertainty, for which the regularization term can be seen as a prior distribution. The success of the Conflictual loss in restoring the epistemic calibration properties of a DE are ultimately due to combining an uninformative prior and a specialization of the models in the ensemble. The use of an uninformative prior in the output space simplifies the choice of the prior and makes it straightforward as defining a prior distribution in the output space is fairly simpler than in the parameters space. Furthermore, the specialization of the models forces conflict in the deep ensemble, and hence a stronger form of diversity.

Conflictual DE, once trained, exhibited two main behaviors that are associated with samples that are difficult or easy to classify. In the former, each model in the ensemble predicts its specialized class as the top class, whereas in the latter, the ensemble models correctly predict the target class, with the specialized class being the second-highest ranked prediction. By analyzing the specificities of Conflictual DE, we noticed that the diversity is mostly (and visibly) created at the last classification linear layer. Therefore, Conflictual Layer was proposed, with reduced space and time complexity, yet resulting in a calibrated epistemic uncertainty. In addition, the parameters of Conflictual Layer have similar trends as noticed in Conflictual DE.

7.1.4 Practical applications

With epistemic uncertainty being considered the ideal measure in OOD detection and active learning, having a calibrated epistemic uncertainty is a must. Therefore, we evaluate the Conflictual loss in Chapter 6 on these applications. For OOD detection, we observed on the two moons dataset that epistemic uncertainty met expectations only for the models trained with the Conflictual loss. Additionally, when investigating OOD detection in the previously evaluated dual-dimensional experiment, Conflictual DE yields closer results to the best performing model (DE with LS) in this task. However, DE with LS has poor calibration from an aleatoric (ECE) and epistemic perspectives, unlike Conflictual DE. Finally, Conflictual DE was the only model capable of clearly distinguishing MNIST-like samples vs ambiguous samples in the AmbiguousMNIST dataset, from an uncertainty point of view.

Rather than only tracking the accuracy at each step of the AL loop, we included additional metrics for a better understanding of the model performance in a broader sense. By evaluating two acquisition functions (predictive and epistemic uncertainties) on datasets representing different levels of difficulties, one relatively straightforward (MNIST), and the other more challenging (CIFAR10), we affirmed the positive impact of a calibrated epistemic uncertainty on the Bayesian active learning loop. While the accuracy curves show a slight advantage of using Conflictual loss, it is more visible when analyzing the additional metrics. For instance, the model are better calibrated (aleatorically) as illustrated by Brier score and ECE, and the misclassified samples are detected more effectively based on uncertainties.

7.2 Perspectives

Our comprehensive approach offers valuable insights into the calibration of epistemic uncertainty. However, it is evident that the complexity of this aspect extends beyond the scope of this work. While the possibilities are limitless, we will explore potential extensions in this section to further analyze the epistemic calibration. These extensions may include establishing potential links with existing phenomena or suggesting additional tests based on the proposed experiments.

7.2.1 Double-Descent

When looking at the fundamental principles of epistemic uncertainty, one could find similarities with the *double-descent* phenomenon (Belkin et al., 2019; Nakkiran et al., 2019; Lafon and Thomas, 2024). More precisely, the *model-wise double-descent* (Belkin et al., 2019) indicates that the error on the test set, as a function of the model size, first improves, then worsens before improving again. On the other hand, Nakkiran et al. (2019) showed that the same phenomenon exists for the test error as a function of the epoch, and referred to it as *epoch-wise double-descent*. Visual illustrations of the phenomenon can be found in the left plot of Figure 7.1, and we reproduced the same experiment (Figure 7.2a and Figure 7.2b) and reported the performance on the validation dataset rather than the test dataset. Importantly, the double-descent phenomenon do not occur on the train error.

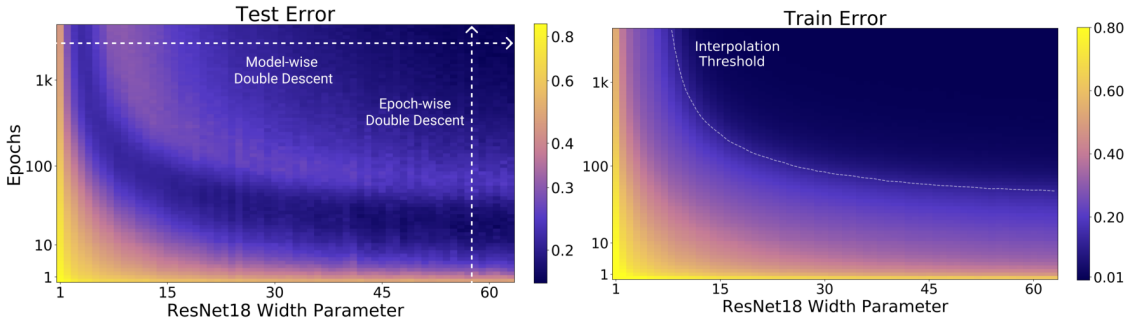
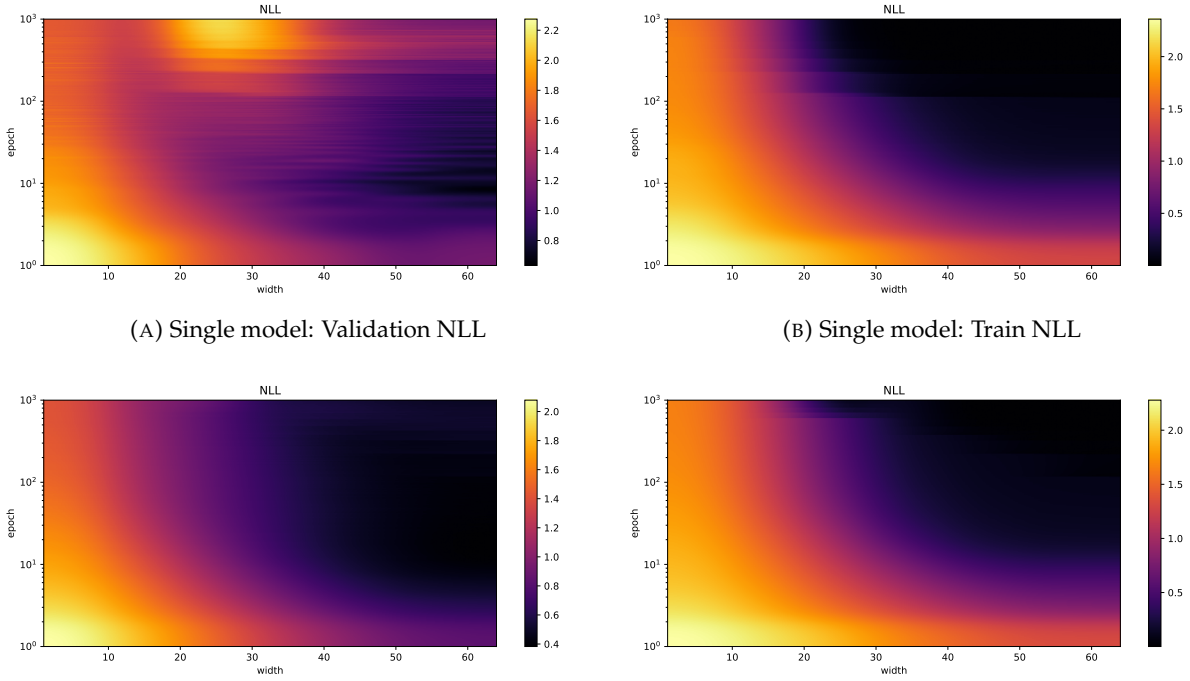


FIGURE 7.1: Illustration of model-wise and epoch-wise double-descents. Credits to Nakkiran et al. (2019).

Although double-descent is more related to the error (NLL), and hence focuses on the level-1, the epistemic uncertainty hole could share similar causing factors with double-descent. For instance, Nakkiran et al. (2021) showed that double-descent can be alleviated if l2-regularization is used and its hyperparameter is carefully finetuned, suggesting the effect of the prior over the weights on double-descent. Moreover, under reasonable marginalization, Wilson and Izmailov (2020) assert and show empirically that double-descent do not occur for BNNs. In agreement, when replacing the single model with an ensemble of ten ResNet, we found that double-descent is not seen with a DE (Figure 7.2c and Figure 7.2d).



(A) Single model: Validation NLL

(B) Single model: Train NLL

(C) DE: Validation NLL

(D) DE: Train NLL

FIGURE 7.2: Illustration of double-descent in the case of a single model (on the first row), and the absence of double-descent with a DE of ten models (on the second row). We use the same experimental setup as in Nakkiran et al. (2019).

7.2.2 Choice of the ideal prior

In Definition 5.1, we formulated four necessary criteria a conflictual output prior should meet. Both the proposed priors satisfy these criteria and have unique behaviors: an encouraging behavior vs an encouraging-discouraging behavior. For the former, the criteria, especially the *class immiscibility*, are most satisfied for larger α. Yet, due to the regularization coefficient that depends on the sizes of the datasets, the choice of α is empirical and indirect. For the latter, while the term α remains in the final expression of the loss function, choosing an ideal value is challenging and ambiguous.

A promising research perspective is to objectively choose the explicit value of α. In this regard, one could take advantage of the field of reference priors and define some sufficient conditions for this value. Furthermore, being directly linked to epistemic uncertainty, the objectively selected prior should lead to calibrated epistemic uncertainty, while providing high performance. For instance, Berger et al. (2015) recommended as a Dirichlet prior, $\texttt{Dir}(1/C \cdot \mathbf{1}_C)$, with this recommended value being inline with the Jeffreys prior in the binary case.

7.2.3 Score-based epistemic calibration

Analyzing the calibration of epistemic uncertainty through fundamental principles may elevate subjectivity in the choice of the prior distribution. However, this approach can be costly and sometimes infeasible. To this end, it is important to investigate whether it is possible to define properties or score-based approaches that objectively evaluate the epistemic calibration for a given trained model. Importantly, the proposed solution must be applicable to any Bayesian neural network, as long as we can measure epistemic uncertainty.

7.2.4 Additional tests

Although extensive research has been conducted in this work on the calibration of epistemic uncertainty across various models, datasets, and tasks, further testing is necessary to ensure the generalization of the epistemic uncertainty hole for epistemically uncalibrated models. This testing will also help determine the role of Conflictual loss in mitigating this issue. For instance, evaluating more complex models is expected to yield similar observations as with the tested models, yet it requires empirical confirmation. Moreover, OOD detection might be examined more extensively by examining a broader range of OOD datasets and avoiding the restriction to a single dataset. The case of active learning might also benefit from additional benchmarks. While the tested datasets consist of noise-free samples, the benefits of acquiring unlabeled batches based on epistemic uncertainty is less visible. Therefore, it is likely to illustrate the benefit of a calibrated epistemic uncertainty on realistic datasets with intrinsic noise.

Appendices

APPENDIX A
USE CASE: BBB WITH DIFFERENT PRIORS

This appendix includes implementation details and additional results for Section 3.2.6.

A.1 Implementation details

The BNN consists of an MLP with two hidden layers (512 and 256), and ReLU activation layers. Each (Bayesian) linear layer is associated with a Gaussian distribution for the posterior variational distribution, whose parameters will be learned during training. The goal of the experiment is show the effect of the prior distribution on the learned (variational) posterior, illustrating the difficulty of setting a prior on the parameters space. Three classical prior distributions were tested: Normal, Laplace and Cauchy. These priors were tested in two regimes consisting of different values for their scales: moderate (0.2) and vague (2). In addition, we also tested the Mixture of Gaussians used in Blundell et al. (2015) and a uniform prior is tested, bringing the total number of tested configurations of the prior to eight.

In each configuration, the model is trained for 150 epochs, with a batch size of 200. We used Adam optimizer with a learning rate of 10^{-3} coupled with the learning rate scheduler *ReduceLROnPlateau*[24] based on the validation NLL, with a factor of 0.5 and patience of 10. Furthermore, 20 MC samples were used to approximate the training loss, compared to 50 for the metrics on the validation and the test datasets.

We only consider the case of MNIST, and the training dataset was partitioned in two stages: initially, 20% of the default train set was set aside to form the validation set, subsequently, a second split of 10% was applied to the remaining data (80%) to obtain a reduced subset of the training samples used for model training, which we refer to is as the train set. For the test set on the other hand, we used the by default test set of MNIST.

In all visualizations of the parameters of the learned Gaussian variational distribution, we distinguish between the mean (μ) and the variance (σ) for both the weights and the biases. Hence, as we have three linear layers, each visualization consists of three rows with four columns each. Due to the high dimensionality of each parameter, we are limited to using aggregated visualizations in the form of histograms.

[24] https://docs.pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

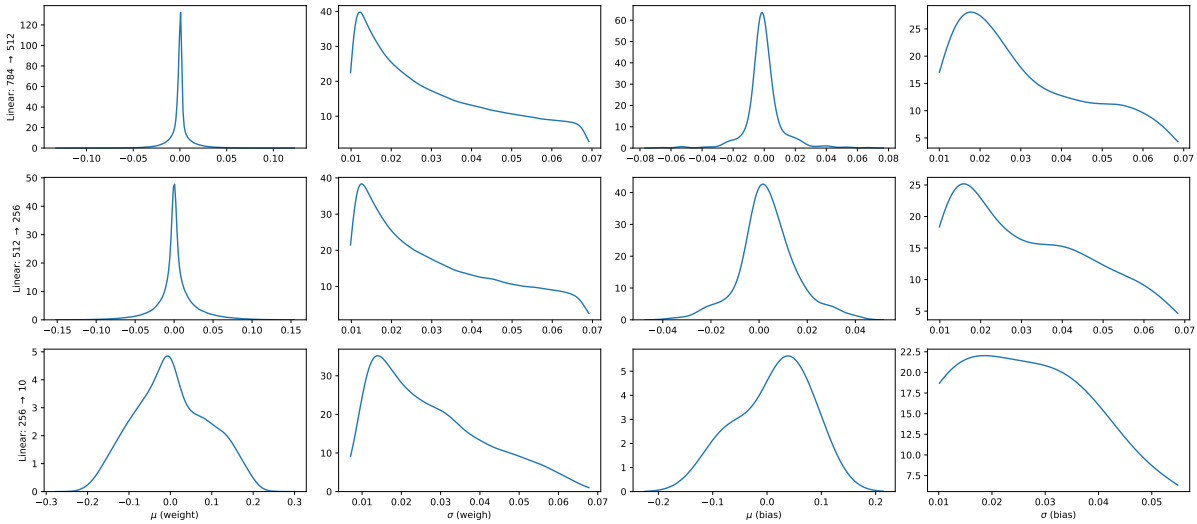


FIGURE A.1: $\mathcal{N}(0, 0.2)$

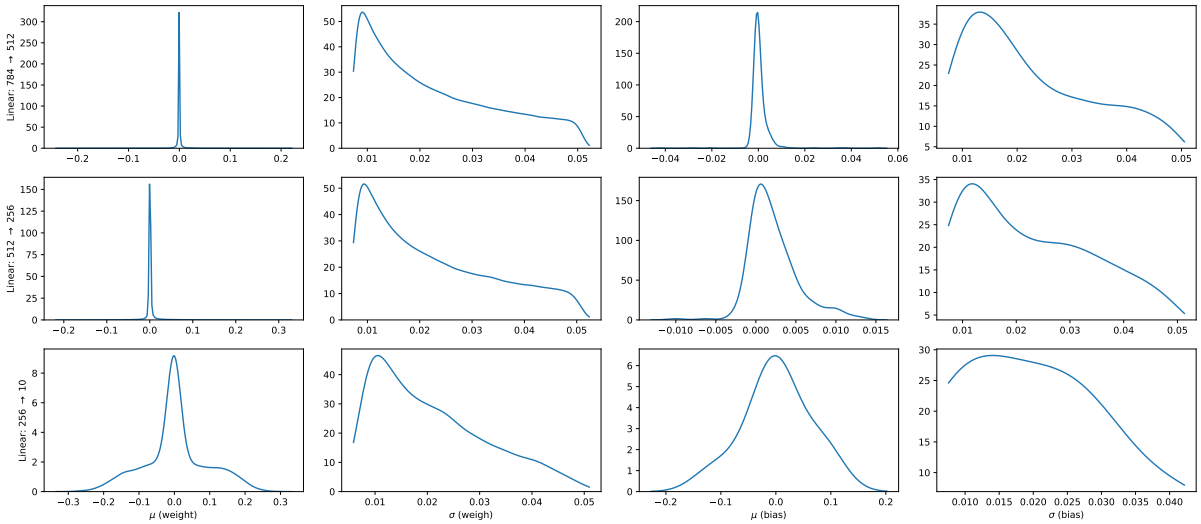


FIGURE A.2: Laplace$(0, 0.2)$

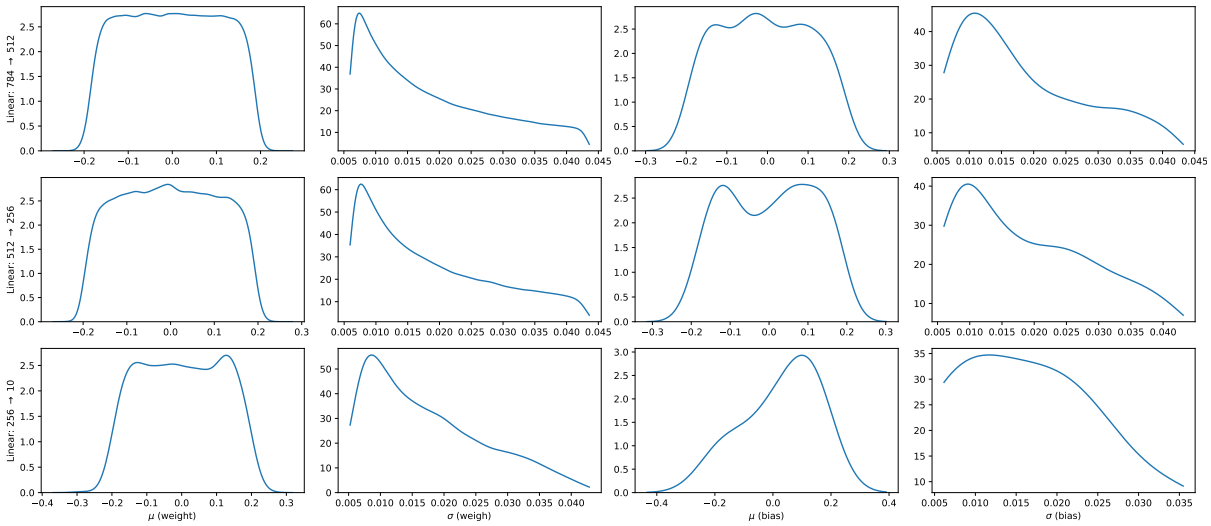


FIGURE A.3: Cauchy$(0, 2)$

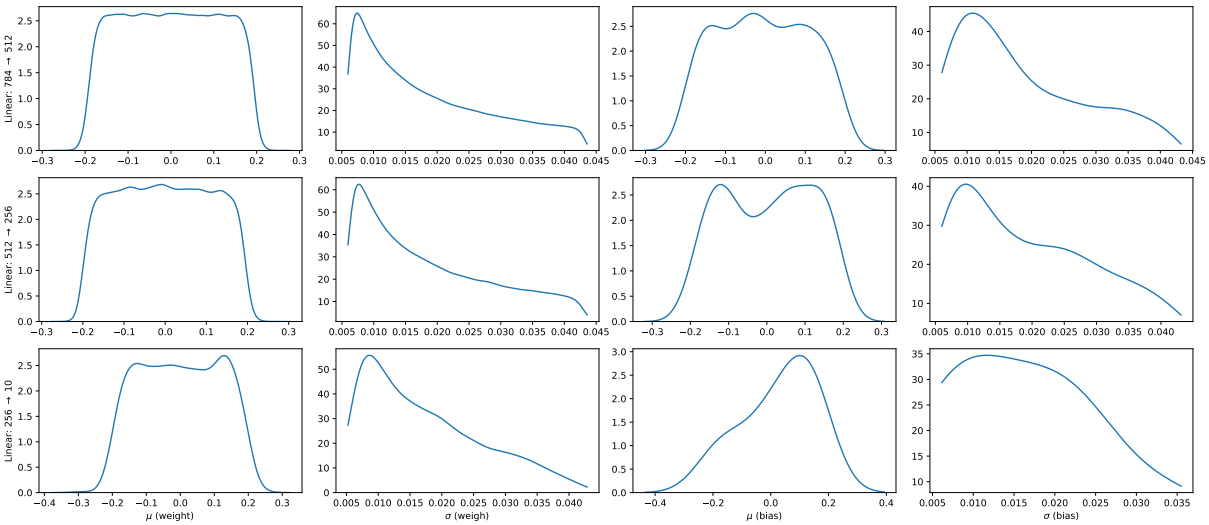


FIGURE A.4: $\mathcal{N}(0, 2)$

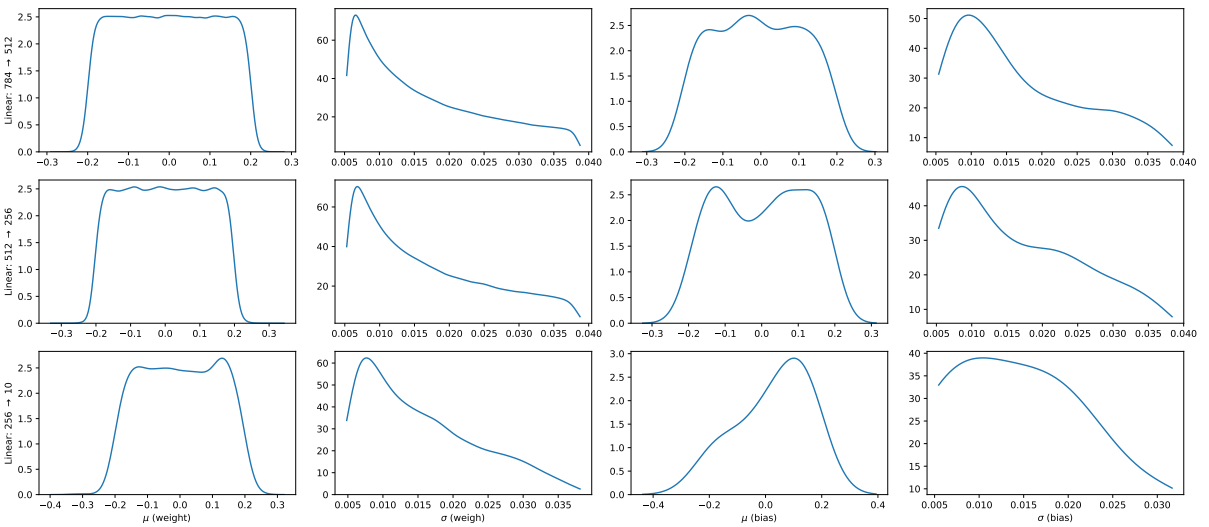


FIGURE A.5: Uniform$(-0.5, 0.5)$

APPENDIX B
DUAL-DIMENSIONAL EXPLORATION

These details are similar to those in our work Fellaji et al. (2024).

B.1 Datasets

As described in the paper, the models were trained on the MNIST and CIFAR10 datasets. A 20% validation split was applied to the training data, after which subsets were selected for training: a total of 48000 samples for MNIST, 58606 for SVHN, and 40000 for CIFAR10. Care was taken to ensure these subsets were class-balanced. For CIFAR10, feature encoding was performed using a pre-trained ResNet34 model. This is functionally equivalent to training a ResNet34 with frozen feature extraction layers, where only the classification head is learned.

B.2 Data transformations

We apply standard normalization to the datasets, using a mean of 0 and a standard deviation of 1. The same transformation is then applied to all test samples, whether they are in-distribution (ID) or out-of-distribution (OOD). Models are trained exclusively on the training samples, with no data augmentation used.

B.3 Training

The models were trained for 500 epochs on MNIST, 600 epochs on SVHN and 700 on CIFAR10. We used the SGD optimizer with weight decay, parameterized with (learning rate, momentum): (0.01, 0.95) for MNIST, (0.02, 0.95) for SVHN, and (0.04, 0.9) for CIFAR10. Each ensemble was trained on a single GPU and the best model (based on the validation loss) was tracked during training and used for early stopping and the learning rate scheduler. A small weight decay was also added, proportional to the size of the training set, to prevent overfitting (10 times lower for MNIST compared to CIFAR10). The parallel execution of the different configurations was possible thanks to GNU Parallel (Tange, 2018).

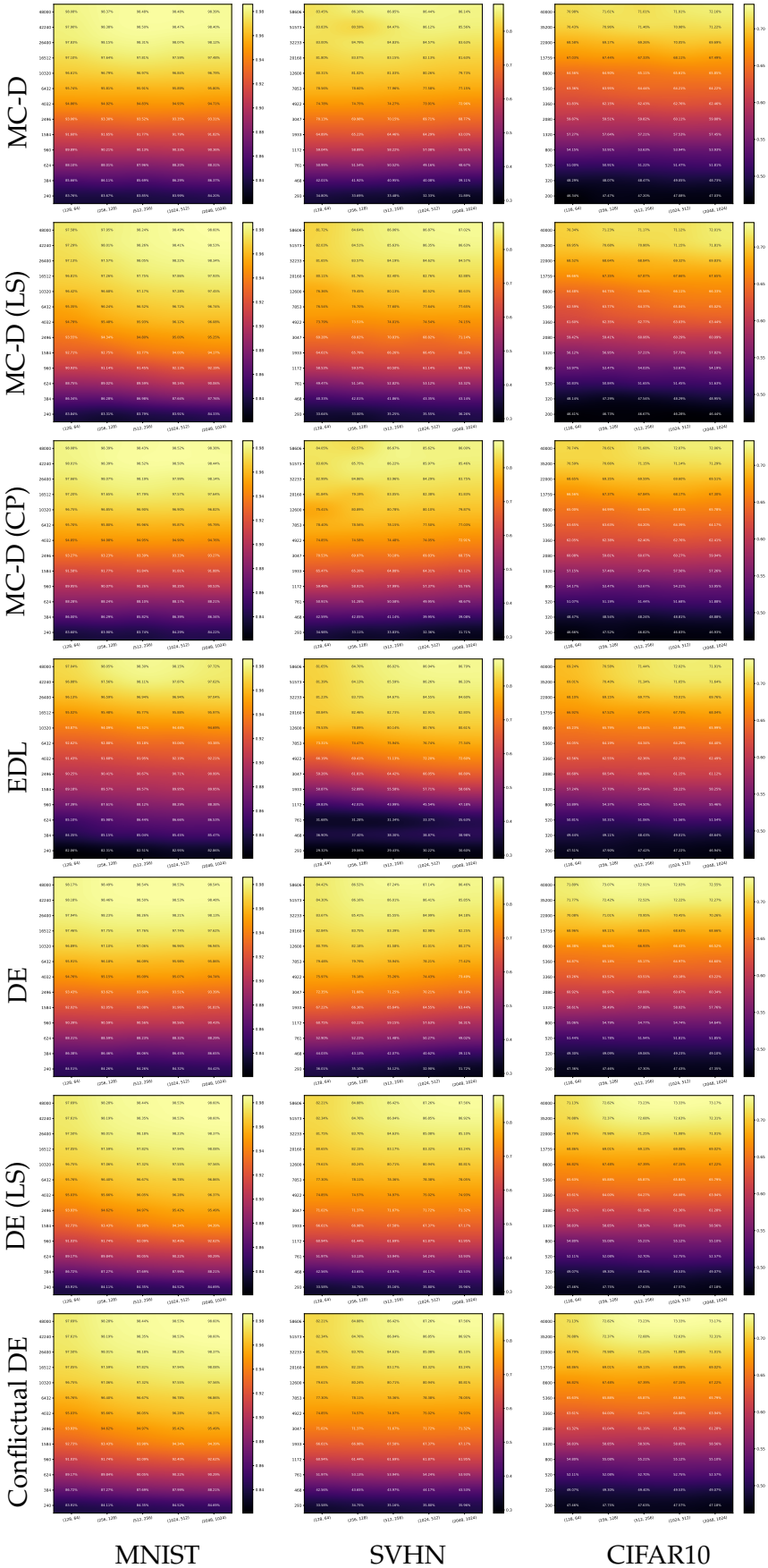


FIGURE B.1: Accuracy. Heatmaps normalized per dataset (column).

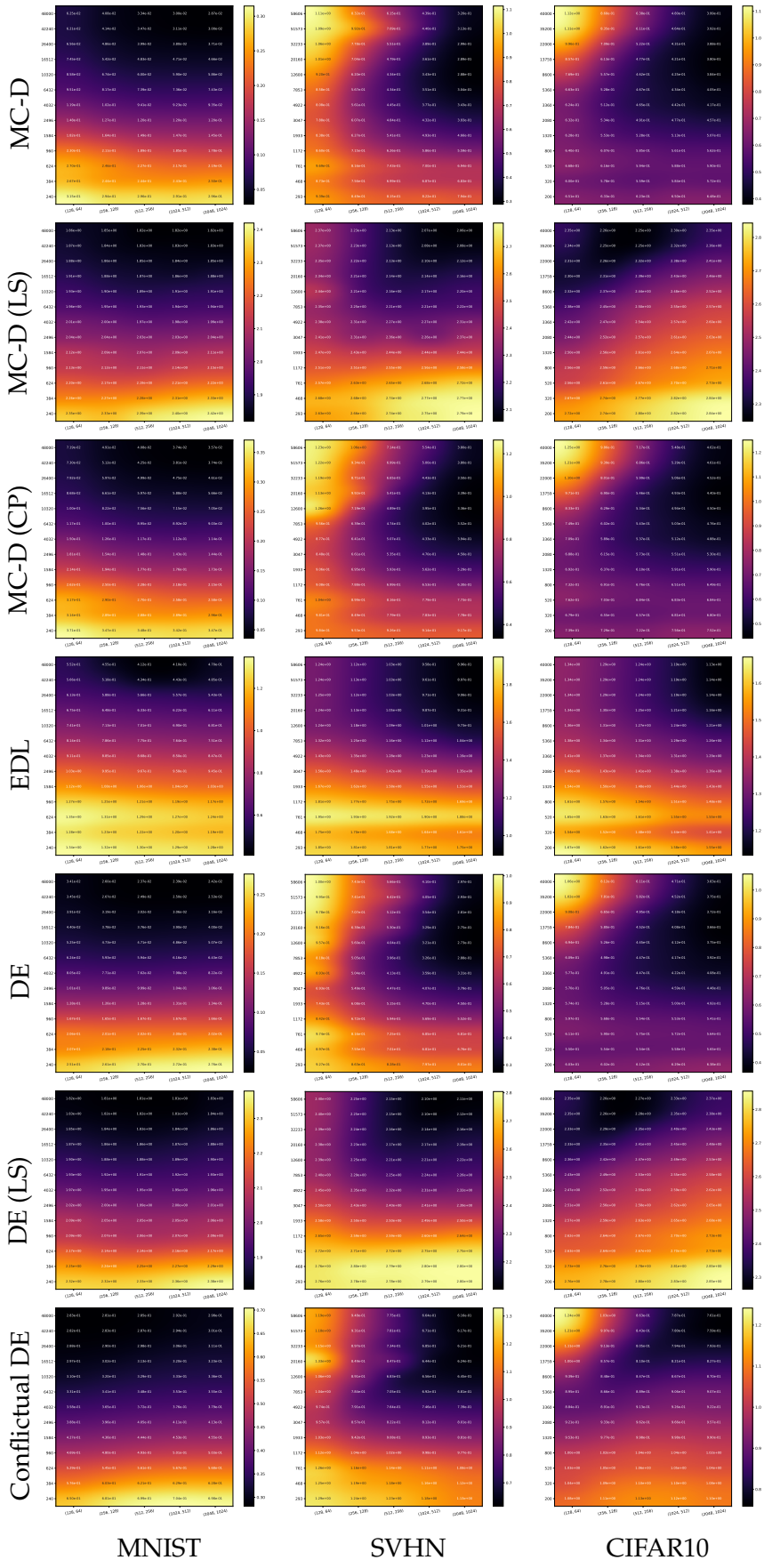


FIGURE B.2: Mean of aleatoric uncertainty on the test set after training the model.

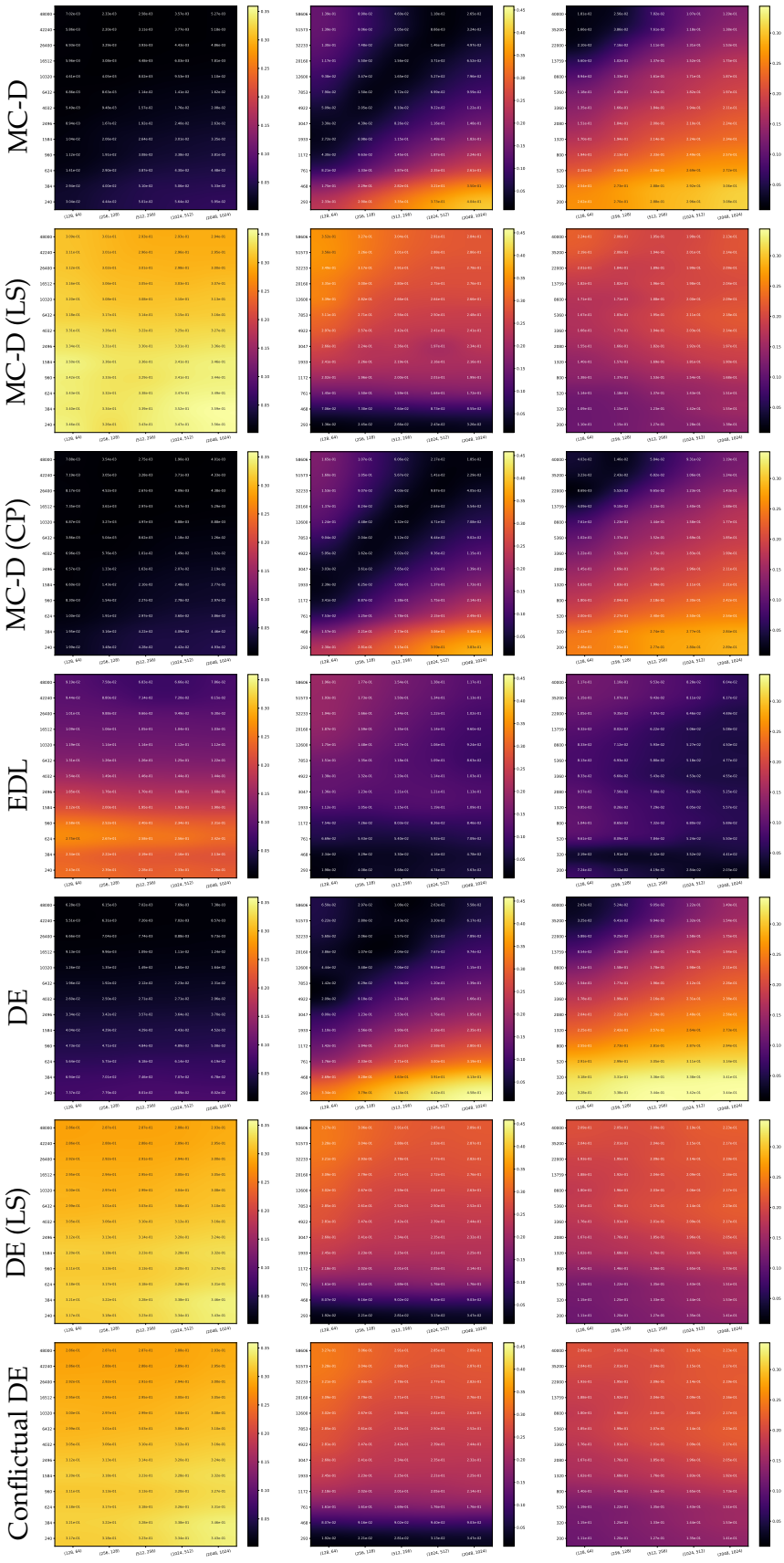


FIGURE B.3: calibration quality of predictive distribution as measured by ECE (lower is better). Heatmaps normalized per dataset (column).

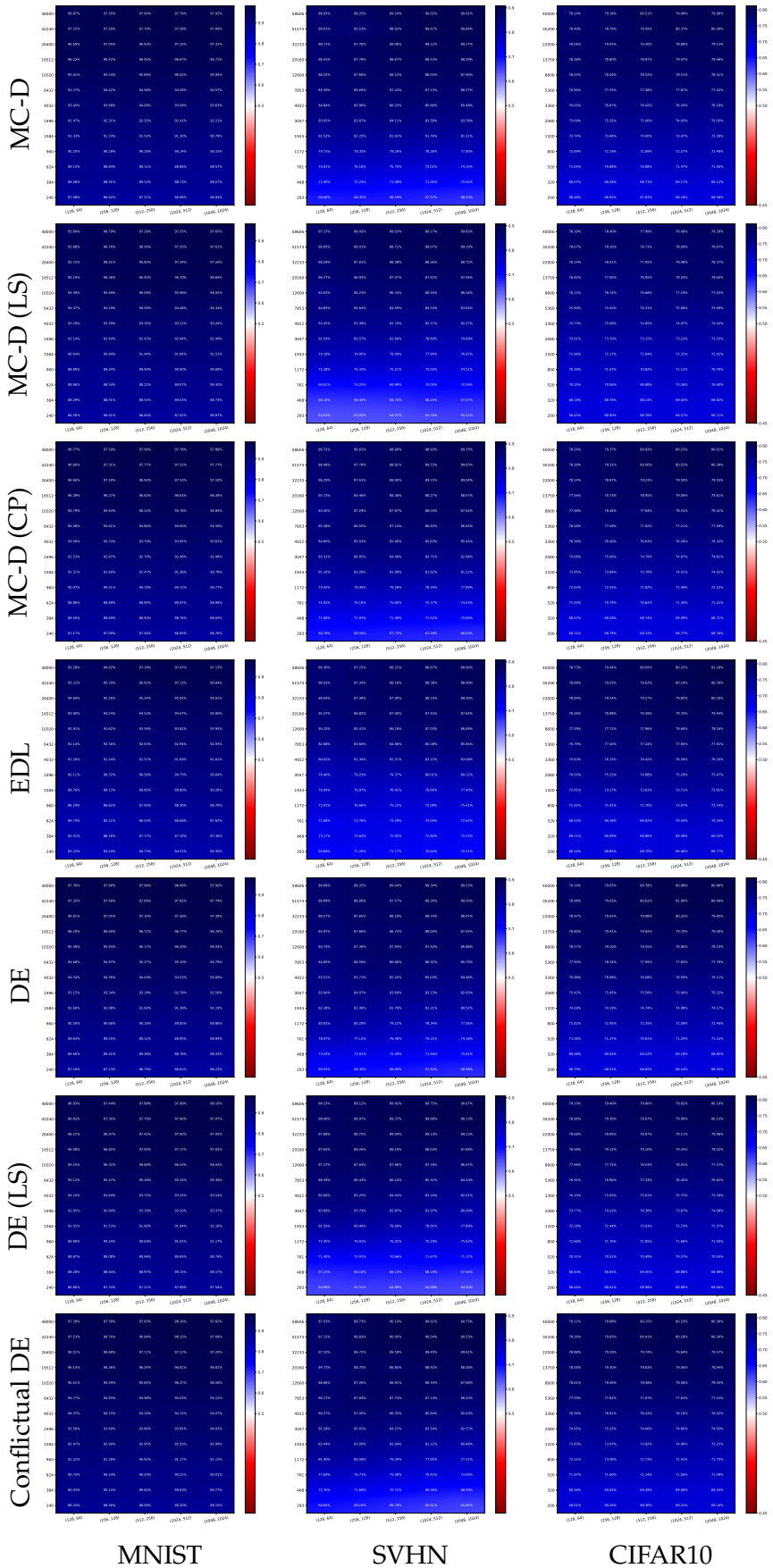


FIGURE B.4: Misclassification AUC-ROC based on total uncertainty.

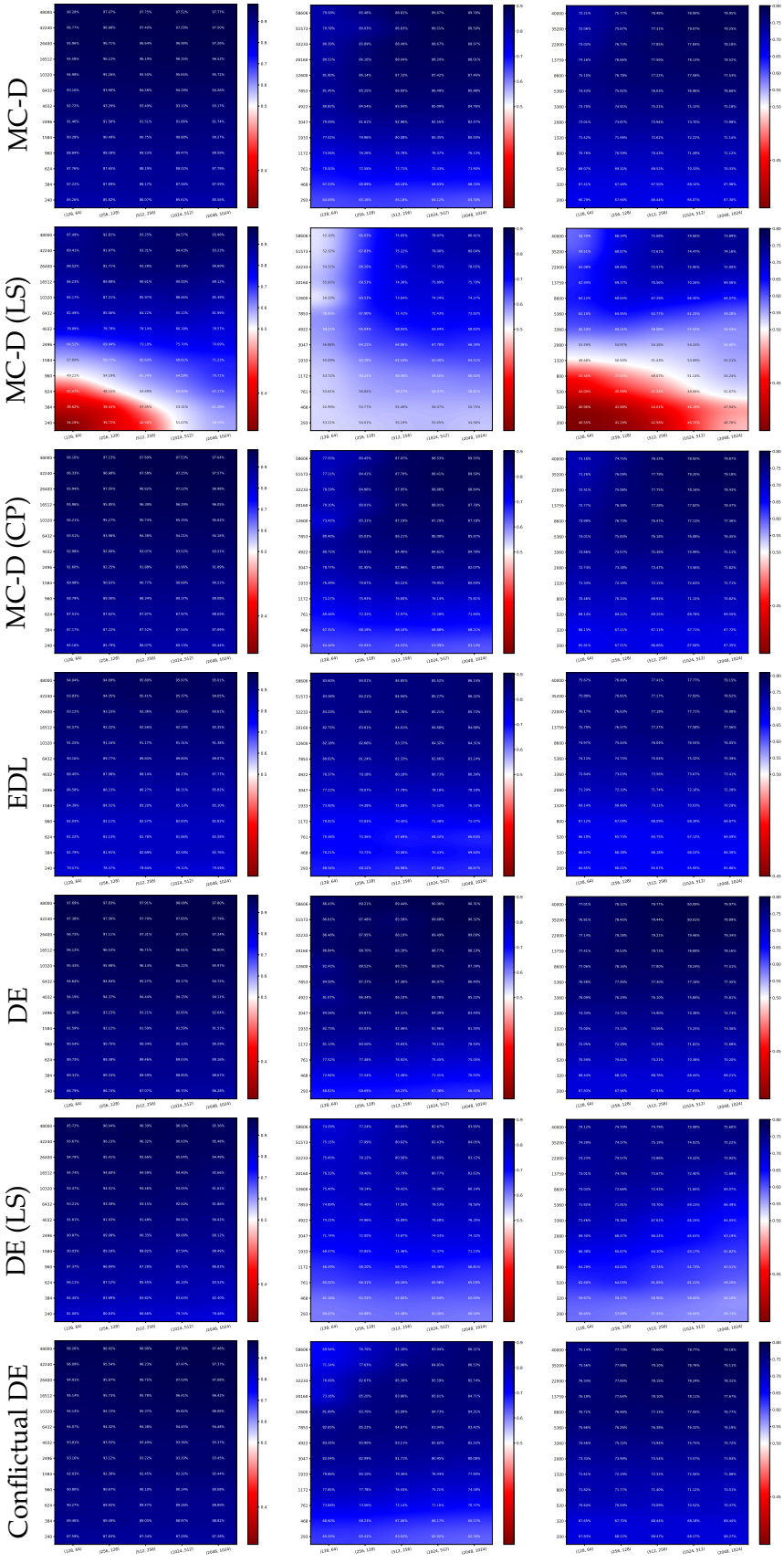


FIGURE B.5: Misclassification AUC-ROC based on epistemic uncertainty.

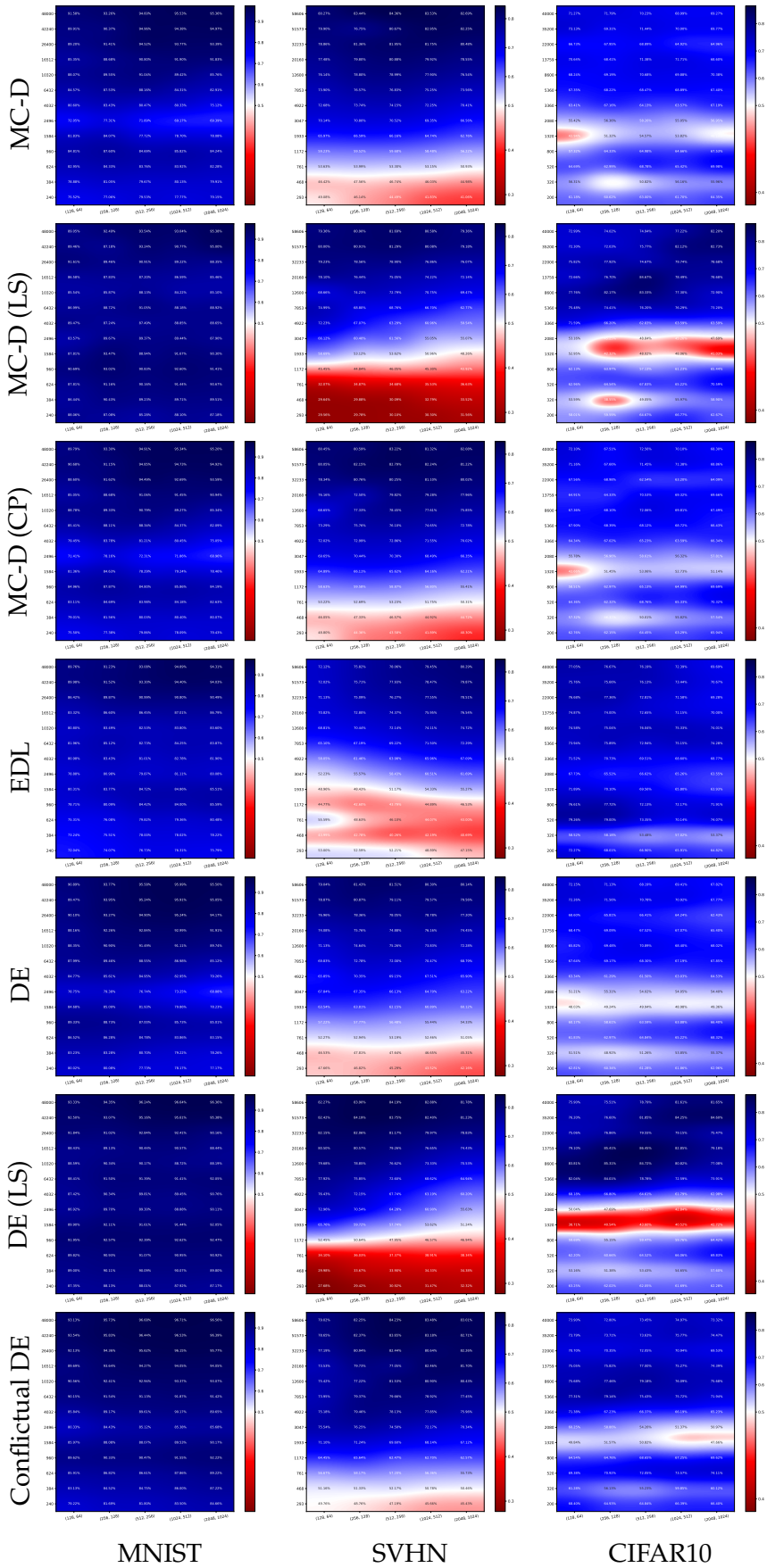


FIGURE B.6: OOD AUC-ROC based on total uncertainty. It is less problematic with Conflictual DE.

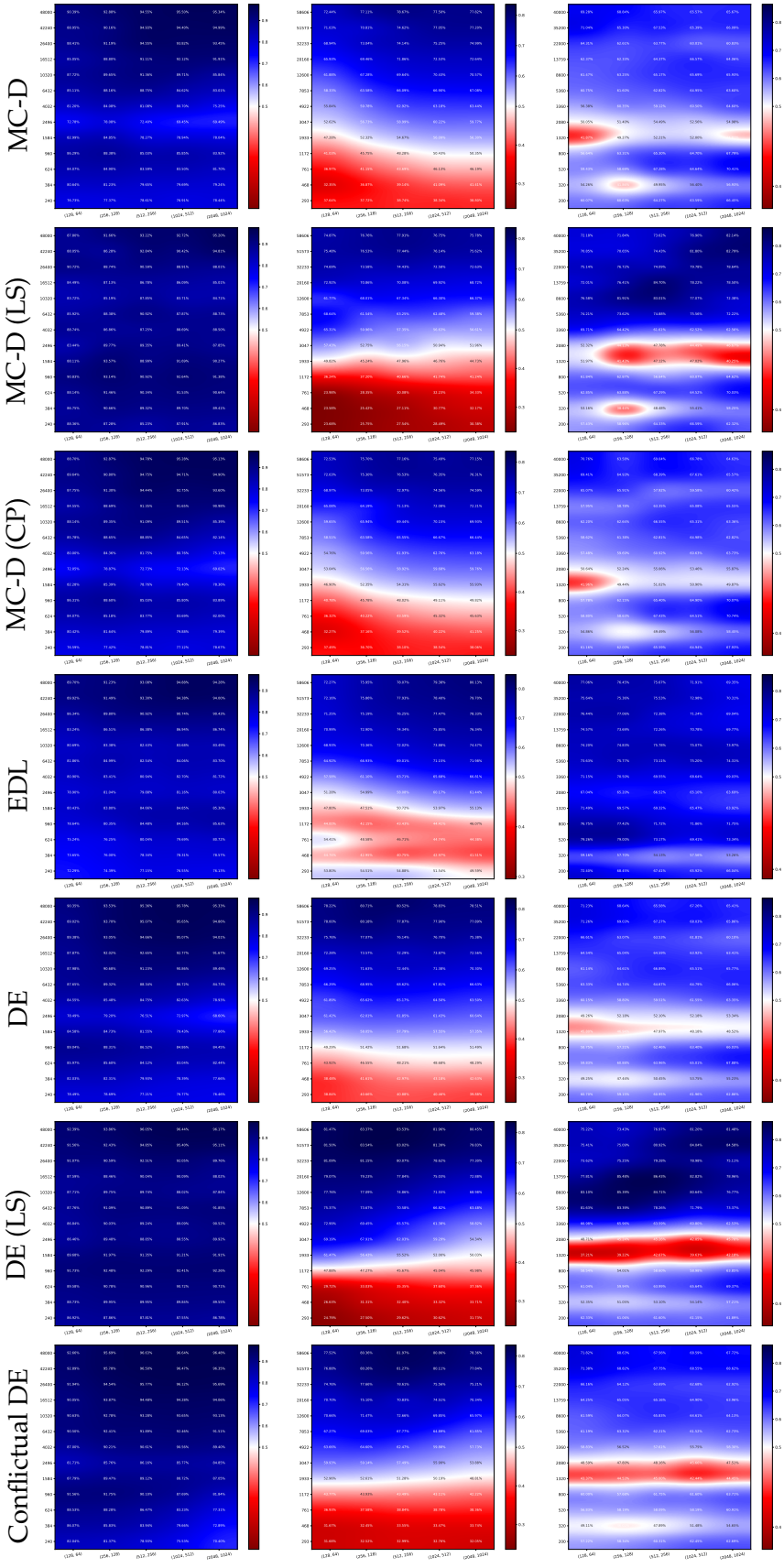


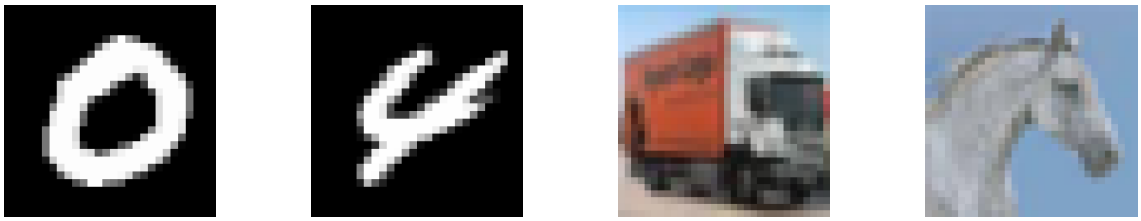
FIGURE B.7: OOD AUC-ROC based on aleatoric uncertainty.

APPENDIX C

FOCUSING ON SPECIFIC INPUTS

In order to gain a better understanding on how the predictions are made, we aim to identify samples that either maximize or minimize epistemic uncertainty. This process can be carried out heuristically across all the tested models, each independently estimating epistemic uncertainty for a shared set of inputs (test set). The top-K samples with the highest or lowest epistemic uncertainty values are selected for each model. The final set of chosen samples is then determined by taking the intersection of these top-K selections across all models, ensuring that the selected samples represent a consensus regarding regions of high or low epistemic uncertainty.

The result of the selection on MNIST and CIFAR10 for the samples that mutually maximize or minimize epistemic uncertainty can be found in Figure C.1, which is the same as Figure 4.6. Three models were tested in total: MC-Dropout, DE and Conflictual DE (Table C.1).



(A) MNIST (6114 – 0) low epistemic uncertainty (Figure 4.6a)

(B) MNIST (6769 – 4) high epistemic uncertainty (Figure 4.6b)

(C) CIFAR10 (8931 – 9) low epistemic uncertainty (Figure 4.6c)

(D) CIFAR10 (6158 – 7) high epistemic uncertainty (Figure 4.6d)

FIGURE C.1: Examples of images resulting in high or low epistemic uncertainty from the test sets of MNIST and CIFAR10. The indices and the true labels are reported under each image (index – label).

	MNIST (6114 – 0)	MNIST (6769 – 4)	CIFAR10 (8931 – 9)	CIFAR10 (6158 – 7)
MC-Dropout	Figure C.2	Figure C.3	Figure C.4	Figure C.5
DE	Figure C.6	Figure C.7	Figure C.8	Figure C.9
Conflictual DE	Figure C.10	Figure C.11	Figure C.12	Figure C.13

TABLE C.1: Mapping of models and the predictions of the inputs from Figure C.1.

FIGURE C.2: Predictions by MC-Dropout for Figure 4.6a (the true label is 0).
Logits on the left and Softmax on the right.

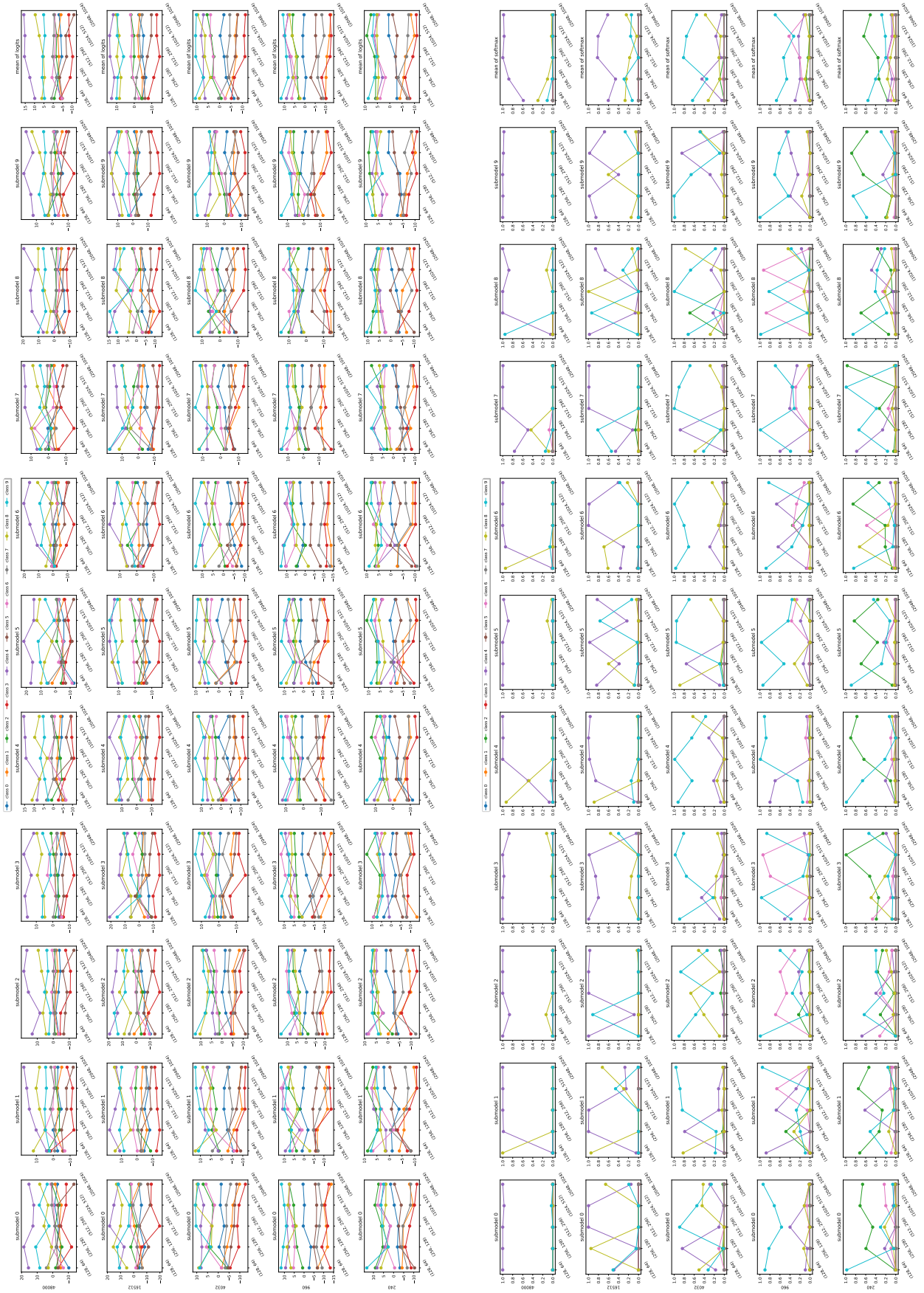


FIGURE C.3: Predictions by MC-Dropout for Figure 4.6b (the true label is 4).
Logits on the left and Softmax on the right.

FIGURE C.4: Predictions by MC-Dropout for Figure 4.6c (the true label is 9).
Logits on the left and Softmax on the right.

FIGURE C.5: Predictions by MC-Dropout for Figure 4.6d (the true label is 7).
Logits on the left and Softmax on the right.

FIGURE C.6: Predictions by DE for Figure 4.6a (the true label is 0).
Logits on the left and Softmax on the right.

FIGURE C.7: Predictions by DE for Figure 4.6b (the true label is 4).
Logits on the left and Softmax on the right.

FIGURE C.8: Predictions by DE for Figure 4.6c (the true label is 9).
Logits on the left and Softmax on the right.

FIGURE C.9: Predictions by DE for Figure 4.6d (the true label is 7).
Logits on the left and Softmax on the right.

FIGURE C.10: Predictions by Conflictual DE for Figure 4.6a (the true label is 0). Logits on the left and Softmax on the right.

FIGURE C.11: Predictions by Conflictual DE for Figure 4.6b (the true label is 4). Logits on the left and Softmax on the right.

FIGURE C.12: Predictions by Conflictual DE for Figure 4.6c (the true label is 9). Logits on the left and Softmax on the right.

FIGURE C.13: Predictions by Conflictual DE for Figure 4.6d (the true label is 7). Logits on the left and Softmax on the right.

APPENDIX D

UNINFORMATIVE PRIORS: BEYOND UNIFORMITY

In Section 5.5, we showed that, in addition to a calibrated epistemic uncertainty, Conflictual Model behaves similarly to Conflictual DE, and shows similar trends especially in the last layer. In Figure 5.13, we analyzed an MLP with two hidden layers (512 and 256), each followed by a Dropout layer ($p = 0.3$) and a ReLU activation function. This model was trained on 24000 samples from MNIST, and the last layer is indeed Conflictual Layer.

To show the generalization of the results, we replaced the MLP with a convolutional neural network which consists of two convolutional blocks with 32 and 64 filters respectively, each followed by dropout ($p = 0.5$), max pooling, and ReLU activation to reduce overfitting and introduce non-linearity. The extracted features are flattened and passed through a fully connected layer with 256 units, followed by a ReLU and a Dropout layer ($p = 0.2$) then a final Conflictual layer for classification. The results are shown in Figure D.1, and we can clearly see the specialization in Conflictual Layer in this convolutional neural network.

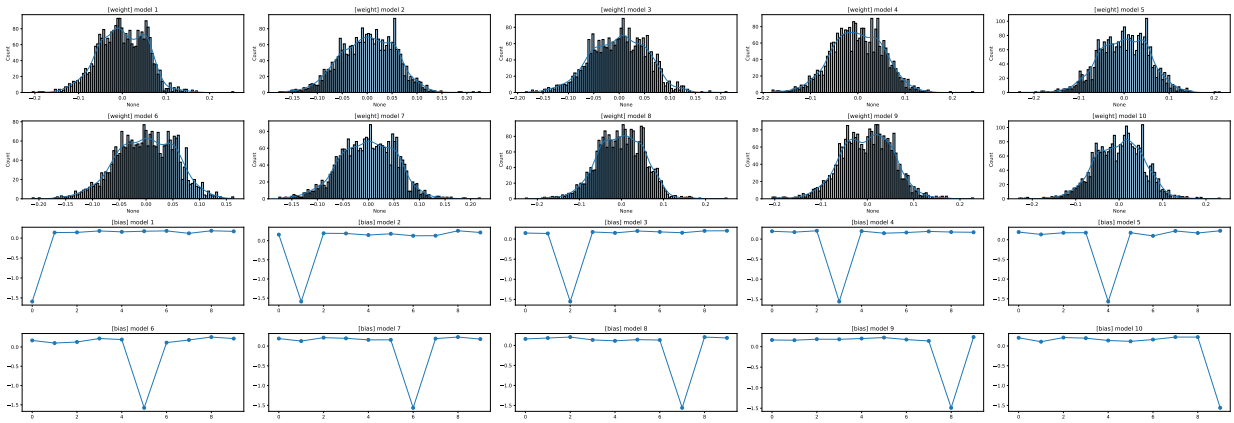


FIGURE D.1: Histograms of the weights and the biases of a Conflictual convolutional neural network with the last layer being a Conflictual Layer (for $\lambda = 0.05$).

APPENDIX E

TWO MOONS DATASET

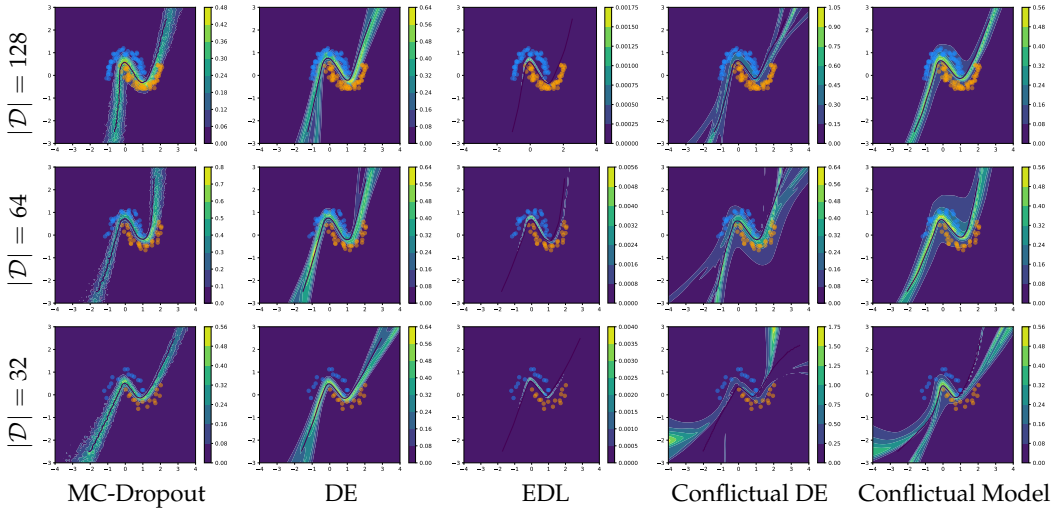


FIGURE E.1: Energy, after applying an exponential transformation. Each subplot uses a different colormap scale.

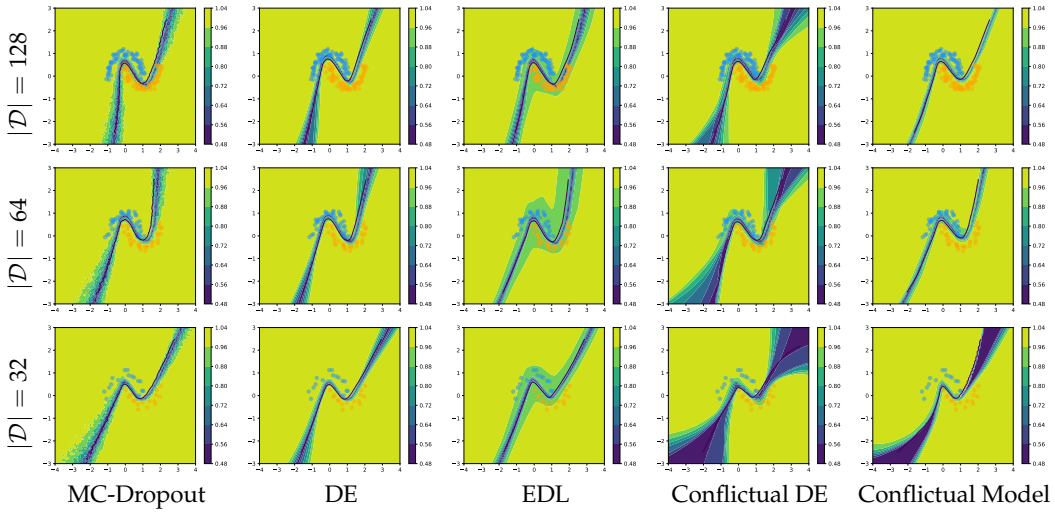


FIGURE E.2: MSP. Each subplot uses a different colormap scale.

APPENDIX F
OVERVIEW OF SOME MODEL FAMILIES

Nowadays, machine learning is widely used in different applications with a raising attention in additional tasks. Namely, computer vision is one domain that benefited from the advance of machine learning and more precisely of deep learning. In image classification for example, the model will take the raw images as input and predict the most likely class. Although the complexity of this task, models are becoming more and more accurate.

Arguably, one of the most used family of models is deterministic deep learning models. This can be attributed to extensive research in three core fields: algorithms, data and hardware. The algorithm's aspect encompasses both the extensive research in model's architecture and the advances in the software. Undeniably, the increasing availability of the data allows the training of the model to yield high performance. In addition, data is considered a good regularizer as it contributes largely to reducing overfitting and increasing the accuracy of the trained model. Finally, advances in hardware had a direct positive influence on the field of deep learning by enabling more efficient and powerful computation. From faster computations thanks to GPUs and TPUs, to efficient and widely available storage, it is possible to train large models on distributed systems and on substantial clusters.

F.1 AlexNet

Certainly, AlexNet (Krizhevsky et al., 2012) is considered a significant milestone in the deep learning community combining advances in the three aforementioned fields. The performance of AlexNet on ImageNet (Deng et al., 2009) put the spotlight on deep learning models and advocated for the use and success of such models: the more data is available and the deeper the architecture, the better the accuracy of the model. AlexNet had a nearly 10% advantage on the ImageNet challenge once the results were published. This was also possible thanks to spreading the model across two GPUs through a parallelization of the model. It is worth mentioning that Krizhevsky et al. (2012) were not the first to introduce an implementation of convolutional layers in GPUs. Some previous work includes (Chellapilla et al., 2006; Ciresan et al., 2011). In addition, the convolutional model used in Krizhevsky et al. (2012) was inspired by the work of (Ciresan et al., 2011) with a few changes, notably the use of the activation function *ReLU* instead of *tanh* resulting in a faster training.

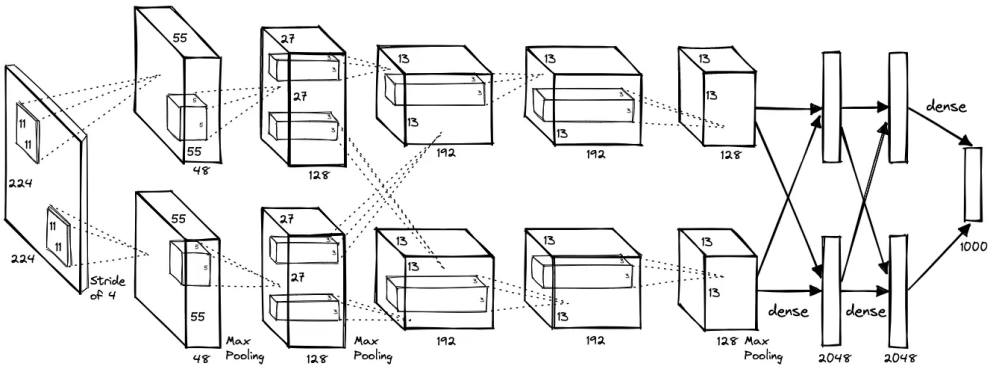


FIGURE F.1: AlexNet architecture.
Credits to https://www.pinecone.io/learn/series/image-search/imagenet

F.2 VGG

The importance of the depth of the model was highlighted in Krizhevsky et al. (2012) in regard to performance as it helps to extract abstract features. This aspect was further investigated in Simonyan and Zisserman (2015). The authors tackled this dimension by using convolutional layers with small receptive fields of 3×3. This choice is particularly interesting for a multitude of reasons. First and almost, increasing the depth of the model by using convolution layers with filters of 3×3 has a small effect on the parameters count. Secondly, this filter size is the smallest allowing to capture the direction' notions: left/right, center, and up/down. Finally, convolutional layers with larger receptive fields could be obtained by stacking multiple 3×3 convolution filters with the additional advantages of injecting more non-linearity functions and decreasing the number of parameters.

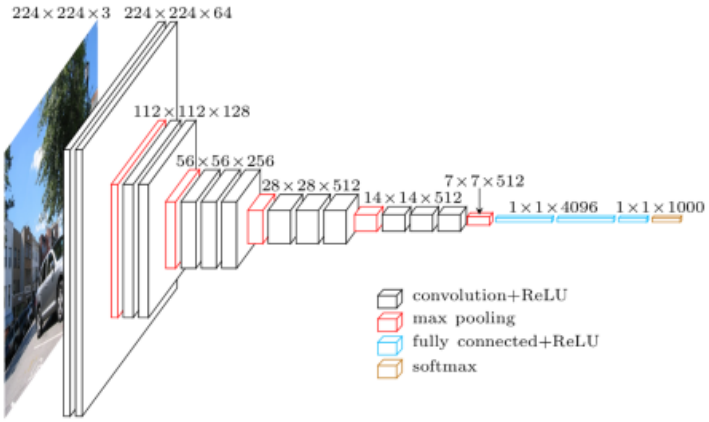


FIGURE F.2: VGG16 architecture.
Credits to: https://www.cs.toronto.edu/~frossard/post/vgg16

F.3 ResNet

On the same idea of the aforementioned architectures, the depth of the model motivated the work of (He et al., 2015). Although increasing the depth of the model should result in better performance, the authors highlighted several problems.

One of these issues is the vanishing/exploding gradients, which can be solved by a proper initialization of the model (Glorot and Bengio, 2010). A more pronounced issue is the *degradation*

problem which occurs when the depth is increased beyond a certain point leading to an increase in the training error. The latter issue was resolved by introducing deep residual learning framework (Figure F.3) which consists of having skip connections. In the worst case scenario, the main idea of residual learning is that stacking an additional layer should not result in the degradation problem and this additional layer should learn at least an identity mapping. The authors hypothesize that it is much easier to force the weights of the weights layers to zero, resulting in an identity mapping for the residual block, rather than learning an identity mapping in the absence of the identity connection.

It is worth mentioning that, in the case when the residual block changes the dimension of the inputs x, the skip connection consists of a projection of the inputs into a space with the desired dimensions.

Increasing the depth is of the model is now possible without the degradation problem thanks to the residual blocks. Thus, a ResNet model is the collection of multiple residual blocks followed by a classification blocks (in the case of a classification setup).

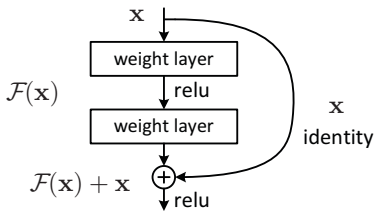


FIGURE F.3: Residual block (He et al., 2015)

At the time of the publication, an ensemble of six ResNet models with different depths had the highest accuracy on the challenge *ILSVRC* (ImageNet Large Scale Visual Recognition Challenge) 2015 (Russakovsky et al., 2015).

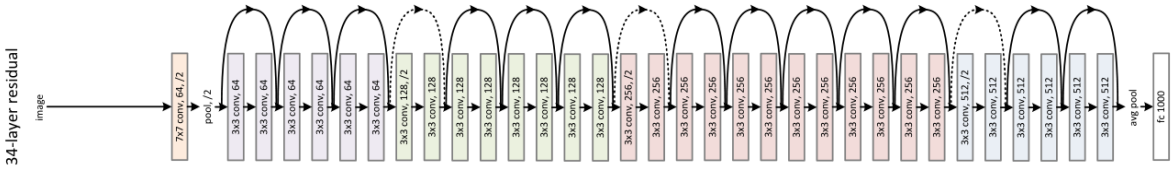


FIGURE F.4: ResNet34 architecture (He et al., 2015)

As shown by Yang and Schoenholz (2017), for large width and depth of ResNet models, gradient exploding is more noticeable and the outputs of the model become inexpressive. In Hayou et al. (2021), the authors focused on the infinite depth limit for ResNet models and argue that the output of the weight layers ($\mathcal{F}(x)$) should be scaled by a certain $\lambda_{l,L}$ with l being the corresponding index of the block, and L is the total number of residual blocks. A *Stable ResNet* is therefor a ResNet model for which the series: $u_L = \sum_{l=1}^{L} \lambda_{l,L}^2$ converges. Since $\lambda_{l,L} = 1$ for ResNet models, the series u_L diverges leading gradient exploding in the limit of an infinite depth.

Although a large set of possibilities exist for the choice of $\lambda_{l,L}$, the authors discuss mainly two: *uniform scaling* $\lambda_{l,L} = 1/\sqrt{L}$ and *decreasing scaling* $\lambda_{l,L}^{-1} = l^2 \times \log(l+1)$. It was shown in the paper that in both the low data regime and high data regime, a large depth (1000) results in a degradation in the performance for the (plain) ResNet model whereas stable ResNet models do not suffer from these phenomena (refer to Table 2 of Hayou et al. (2021)).

F.4 EfficientNet

In the ResNet paper (He et al., 2015), the importance of the depth was highlighted and the introduction of the skip connection allowed to increase the depth without affecting the convergence and performance of the model. Tan and Le (2020) further investigate the scaling of deep learning

models. While this process is not well understood, they propose *compound scaling* (Figure F.5). They show that scaling the model solely based on the depth of the model is suboptimal and advocate that it should be multidimensional: depth, width and image resolution.

On one hand, the depth contributes to capturing richer and complex features. On the other hand, increasing the width makes the training faster and allows extracting fine-grained features. Finally, changing the image resolution has an effect on the fine-grained patterns. Increasing a single aspect will lead to a saturation of performance. For instance, ResNet101 and ResNet1000 have similar accuracies even though the latter is much deeper than the former.

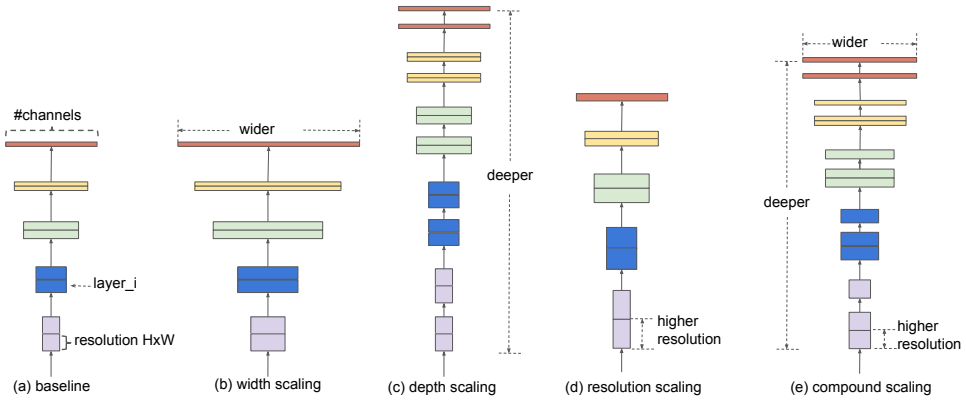


FIGURE F.5: Illustration of compound scaling (Tan and Le, 2020)

With compound scaling, the increase in the three dimension is done in a controlled manner, to solve the following optimization problem (Equation (3) from (Tan and Le, 2020)):

$$\begin{aligned} \text{depth: } d &= \alpha^{\phi} \\ \text{width: } w &= \beta^{\phi} \\ \text{resolution: } r &= \gamma^{\phi} \\ \text{such that: } \alpha.\beta^2.\gamma^2 &\approx 2 \\ \alpha \geq 1,\ \beta &\geq 1,\ \gamma \geq 1 \end{aligned}$$

d, w and r control the scaling of the depth, width and images resolution respectively of the baseline model. Intuitively, ϕ is related to the available resources making the scaling dependent on this forth dimension. Grid search was used to find the ideal values for (α, β, γ), and several models scaled from the baseline model were trained and show impressive results with fewer parameters and reduced FLOPS (floating point operations per second). Compound scaling was applied to existing models, such as ResNet50, and resulted in increasing performance.

BIBLIOGRAPHY

Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., and Woerner, S. (2021). The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409.

Adlam, B., Snoek, J., and Smith, S. L. (2020). Cold Posteriors and Aleatoric Uncertainty. *arXiv preprint arXiv:2008.00029*.

Aitchison, L. (2021). A STATISTICAL THEORY OF COLD POSTERIORS IN DEEP NEURAL NETWORKS. *arXiv preprint arXiv:2008.05912*.

Amersfoort, J. V., Smith, L., Teh, Y. W., and Gal, Y. (2020). Uncertainty Estimation Using a Single Deep Deterministic Neural Network. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9690–9700. PMLR.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete Problems in AI Safety.

Arthur, D. and Vassilvitskii, S. (2006). K-means++: The Advantages of Careful Seeding. http://ilpubs.stanford.edu:8090/778/?ref=https://githubhelp.com.

Ash, J. T., Zhang, C., Krishnamurthy, A., Langford, J., and Agarwal, A. (2020). Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds.

Atighehchian, P., Branchaud-Charron, F., and Lacoste, A. (2020). Bayesian active learning for production, a systematic study and a reusable library.

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–416.

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

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Bengs, V., Hüllermeier, E., and Waegeman, W. (2022). Pitfalls of Epistemic Uncertainty Quantification through Loss Minimisation.

Bengs, V., Hüllermeier, E., and Waegeman, W. (2023). On Second-Order Scoring Rules for Epistemic Uncertainty Quantification. In *Proceedings of the 40th International Conference on Machine Learning*, pages 2078–2091. PMLR.

Berezniuk, O., Figalli, A., Ghigliazza, R., and Musaelian, K. (2020). A scale-dependent notion of effective dimension.

Berger, J. (2006). The case for objective Bayesian analysis. *Bayesian Analysis*, 1(3).

Berger, J., Bernardo, J., and Mendoza, M. (1988). On priors that maximize expected information.

Berger, J. O., Bernardo, J. M., and Sun, D. (2009). The formal definition of reference priors. *The Annals of Statistics*, 37(2).

Berger, J. O., Bernardo, J. M., and Sun, D. (2015). Overall Objective Priors.

Bernardo, J. M. (1979). Reference Posterior Distributions for Bayesian Inference. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 41(2):113–128.

Bernardo, J. M. (2005). Reference Analysis. In *Handbook of Statistics*, volume 25, pages 17–90. Elsevier.

Blalock, D., Ortiz, J. J. G., Frankle, J., and Guttag, J. (2020). What is the State of Neural Network Pruning?

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight Uncertainty in Neural Networks.

Brier, G. W. (1950). VERIFICATION OF FORECASTS EXPRESSED IN TERMS OF PROBABILITY. *Monthly weather review*, 78:1–3.

Brodley, C. E. and Friedl, M. A. (1996). Identifying and Eliminating Mislabeled Training Instances. In *AAAI*, Portland, Oregon.

Bronevich, A. and Klir, G. J. (2008). Axioms for uncertainty measures on belief functions and credal sets. In *NAFIPS 2008 - 2008 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 1–6.

Chellapilla, K., Puri, S., and Simard, P. (2006). High Performance Convolutional Neural Networks for Document Processing. In *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft.

Cho, Y. and Saul, L. (2009). Kernel Methods for Deep Learning. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.

Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. (2011). Flexible, High Performance Convolutional Neural Networks for Image Classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1237. Citeseer.

Clarke, B. S. and Barron, A. R. (1994). Jeffreys' prior is asymptotically least favorable under entropy risk. *Journal of Statistical Planning and Inference*, 41(1):37–60.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314.

D'Angelo, F. and Henning, C. (2022). On out-of-distribution detection with Bayesian neural networks.

Dawid, A. P. (1982). The Well-Calibrated Bayesian. *Journal of the American Statistical Association*, 77(379):605–610.

Dawid, A. P., Stone, M., and Zidek, J. V. (1973). Marginalization Paradoxes in Bayesian and Structural Inference. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 35(2):189–213.

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., and Hennig, P. (2022). Laplace Redux – Effortless Bayesian Deep Learning.

de Jong, I. P., Sburlea, A. I., and Valdenegro-Toro, M. (2024). How disentangled are your classification uncertainties?

de Laplace, P. S. (1820). *Théorie Analytique Des Probabilités*, volume 7. Courcier.

de Mathelin, A., Deheeger, F., Mougeot, M., and Vayatis, N. (2023). Deep Anti-Regularized Ensembles provide reliable out-of-distribution uncertainty quantification.

de Mathelin, A., Deheeger, F., Mougeot, M., and Vayatis, N. (2025). Deep Out-of-Distribution Uncertainty Quantification via Weight Entropy Maximization. *Journal of Machine Learning Research*, 26(4):1–68.

DeGroot, M. H. and Fienberg, S. E. (1983). The Comparison and Evaluation of Forecasters. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 32(1):12–22.

Deng, D., Chen, G., Yu, Y., Liu, F., and Heng, P.-A. (2023). Uncertainty Estimation by Fisher Information-based Evidential Deep Learning.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE.

Depeweg, S., Hernández-Lobato, J. M., Doshi-Velez, F., and Udluft, S. (2018). Decomposition of Uncertainty in Bayesian Deep Learning for Efficient and Risk-sensitive Learning.

Drucker, H. and Le Cun, Y. (1992). Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 3(6):991–997.

Everett, D., Nguyen, A. T., Richards, L. E., and Raff, E. (2022). Improving Out-of-Distribution Detection via Epistemic Uncertainty Adversarial Training.

Fellaji, M. and Pennerath, F. (2023). The Epistemic Uncertainty Hole: An issue of Bayesian Neural Networks. In *Conférence Sur l'Apprentissage Automatique (Affiliated to PFIA)*, Strasbourg, France.

Fellaji, M., Pennerath, F., Conan-Guez, B., and Couceiro, M. (2024). On the Calibration of Epistemic Uncertainty: Principles, Paradoxes and Conflictual Loss. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 160–176, Vilnius, Lithuania. Springer.

Feller, W. (1966). An Introduction to Probability Theory and Its Applications.

Fortuin, V. (2021). Priors in Bayesian Deep Learning: A Review. *arXiv:2105.06868 [cs, stat]*.

Fortuin, V., Garriga-Alonso, A., Ober, S. W., Wenzel, F., Rätsch, G., Turner, R. E., van der Wilk, M., and Aitchison, L. (2022). Bayesian Neural Network Priors Revisited.

Franchi, G., Yu, X., Bursuc, A., Tena, A., Kazmierczak, R., Dubuisson, S., Aldea, E., and Filliat, D. (2022). MUAD: Multiple Uncertainties for Autonomous Driving, a benchmark for multiple uncertainty types and tasks.

Frankle, J. and Carbin, M. (2019). The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks.

Gal, Y. (2016). *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning.

Gal, Y., Islam, R., and Ghahramani, Z. (2017). Deep Bayesian Active Learning with Image Data.

Gao, M., Zhang, Z., Yu, G., Arık, S. Ö., Davis, L. S., and Pfister, T. (2020). Consistency-Based Semi-supervised Active Learning: Towards Minimizing Labeling Cost. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, volume 12355, pages 510–526. Springer International Publishing, Cham.

Gao, Y., Ramesh, R., and Chaudhari, P. (2022). Deep Reference Priors: What is the best way to pretrain a model? In *Proceedings of the 39th International Conference on Machine Learning*, pages 7036–7051. PMLR.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Gneiting, T. and Raftery, A. E. (2007). Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Gumbel, E. J. (1954). Statistical Theory of Extreme Values and Some Practical Applications. A Series of Lectures. Technical Report PB175818, National Bureau of Standards, Washington, D. C. Applied Mathematics Div.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks.

Guzmán-rivera, A., Batra, D., and Kohli, P. (2012). Multiple Choice Learning: Learning to Produce Multiple Structured Outputs. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Havasi, M., Jenatton, R., Fort, S., Liu, J. Z., Snoek, J., Lakshminarayanan, B., Dai, A. M., and Tran, D. (2021). Training independent subnetworks for robust prediction.

Hawkins, D. M. (1980). *Identification of Outliers*. Springer Netherlands, Dordrecht.

Hayou, S., Clerico, E., He, B., Deligiannidis, G., Doucet, A., and Rousseau, J. (2021). Stable ResNet.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.

Hemmer, P., Kühl, N., and Schöffer, J. (2020). DEAL: Deep Evidential Active Learning for Image Classification.

Hendrycks, D. and Gimpel, K. (2017). A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *International Conference on Learning Representations*.

Henning, C., D'Angelo, F., and Grewe, B. F. (2021). Are Bayesian neural networks intrinsically good at out-of-distribution detection?

Hodge, V. and Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, 22(2):85–126.

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

Houlsby, N., Hernandez-Lobato, J. M., and Ghahramani, Z. (2014). Cold-start Active Learning with Robust Ordinal Matrix Factorization. In *Proceedings of the 31st International Conference on Machine Learning*, pages 766–774. PMLR.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian Active Learning for Classification and Preference Learning.

Huijben, I. A. M., Kool, W., Paulus, M. B., and van Sloun, R. J. G. (2022). A Review of the Gumbel-max Trick and its Extensions for Discrete Stochasticity in Machine Learning.

Hüllermeier, E., Destercke, S., and Shaker, M. H. (2022). Quantification of Credal Uncertainty in Machine Learning: A Critical Analysis and Empirical Comparison. In *Uncertainty in Artificial Intelligence*, pages 548–557. PMLR.

Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. *Machine Learning*, 110(3):457–506.

Izmailov, P., Nicholson, P., Lotfi, S., and Wilson, A. G. (2021). Dangers of Bayesian Model Averaging under Covariate Shift.

Jeffreys, H. (1946). An Invariant Form for the Prior Probability in Estimation Problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 186(1007):453–461.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1998). An Introduction to Variational Methods for Graphical Models. In Jordan, M. I., editor, *Learning in Graphical Models*, pages 105–161. Springer Netherlands, Dordrecht.

Jøsang, A. (1997). Artificial Reasoning with Subjective Logic⋆. In *Proceedings of the Second Australian Workshop on Commonsense Reasoning*, volume 48, page 34.

Jürgens, M., Meinert, N., Bengs, V., Hüllermeier, E., and Waegeman, W. (2024). Is Epistemic Uncertainty Faithfully Represented by Evidential Deep Learning Methods? *arXiv preprint arXiv:2402.09056*.

Jürgens, M., Mortier, T., Hüllermeier, E., Bengs, V., and Waegeman, W. (2025). A calibration test for evaluating set-based epistemic uncertainty representations.

Kale, S., Sekhari, A., and Sridharan, K. (2021). SGD: The Role of Implicit Regularization, Batch-size and Multiple-epochs.

Kapoor, S., Maddox, W. J., Wilson, A. G., and Izmailov, P. (2022). On Uncertainty, Tempering, and Data Augmentation in Bayesian Classification. In *Advances in Neural Information Processing Systems*, volume 35, pages 18211–18225. Curran Associates, Inc.

Karakida, R., Akaho, S., and Amari, S.-i. (2019). Universal Statistics of Fisher Information in Deep Neural Networks: Mean Field Approach.

Kendall, A. and Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kingma, D. P. and Ba, J. (2017). Adam: A Method for Stochastic Optimization.

Kingma, D. P. and Welling, M. (2013). *Auto-Encoding Variational Bayes*. Banff, Canada.

Kirsch, A., Farquhar, S., Atighehchian, P., Jesson, A., Branchaud-Charron, F., and Gal, Y. (2023). Stochastic Batch Acquisition: A Simple Baseline for Deep Active Learning.

Kirsch, A., Mukhoti, J., van Amersfoort, J., Gal, Y., and Torr, P. (2021). On Pitfalls in OoD Detection: Predictive Entropy Considered Harmful.

Kirsch, A., van Amersfoort, J., and Gal, Y. (2019). BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning.

Knox, E. M. and Ng, R. T. (1998). Algorithms for Mining Distance-Based Outliers in Large Datasets. In *Proceedings of the 24th VLDB Conference*, New York, USA.

Konyushkova, K., Sznitman, R., and Fua, P. (2017). Learning Active Learning from Data. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Kull, M., Filho, T. S., and Flach, P. (2017). Beta calibration: A well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 623–631. PMLR.

Kull, M., Perello Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., and Flach, P. (2019). Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Lafon, M. and Thomas, A. (2024). Understanding the Double Descent Phenomenon in Deep Learning.

Lahlou, S., Jain, M., Nekoei, H., Butoi, V. I., Bertin, P., Rector-Brooks, J., Korablyov, M., and Bengio, Y. (2023). DEUP: Direct Epistemic Uncertainty Prediction. *Transactions on Machine Learning Research*.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles.

Lambert, B., Forbes, F., Tucholka, A., Doyle, S., Dehaene, H., and Dojat, M. (2022). Trustworthy clinical AI solutions: A unified review of uncertainty quantification in deep learning models for medical image analysis.

Laurent, O., Lafage, A., Tartaglione, E., Daniel, G., Martinez, J.-m., Bursuc, A., and Franchi, G. (2022). Packed Ensembles for efficient uncertainty estimation. In *The Eleventh International Conference on Learning Representations*.

Laves, M.-H., Ihler, S., Kortmann, K.-P., and Ortmaier, T. (2019). Well-calibrated Model Uncertainty with Temperature Scaling for Dropout Variational Inference. *arXiv preprint arXiv:1909.13550*, page 8.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. (2018a). Deep Neural Networks as Gaussian Processes.

Lee, K., Lee, K., Lee, H., and Shin, J. (2018b). A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., and Batra, D. (2015). Why M Heads are Better than One: Training a Diverse Ensemble of Deep Networks.

Lenc, K. and Vedaldi, A. (2015). Understanding Image Representations by Measuring Their Equivariance and Equivalence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 991–999.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. (2020a). Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness. In *Advances in Neural Information Processing Systems*, volume 33, pages 7498–7512. Curran Associates, Inc.

Liu, W., Wang, X., Owens, J. D., and Li, Y. (2020b). Energy-based Out-of-distribution Detection. In *Advances in Neural Information Processing Systems*, volume 33, pages 21464–21475. Curran Associates, Inc.

Loshchilov, I. and Hutter, F. (2019). Decoupled Weight Decay Regularization.

Lotfi, S., Izmailov, P., Benton, G., Goldblum, M., and Wilson, A. G. (2022). Bayesian Model Selection, the Marginal Likelihood, and Generalization.

Louizos, C. and Welling, M. (2017). Multiplicative Normalizing Flows for Variational Bayesian Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2218–2227. PMLR.

Lüth, C. T., Bungert, T. J., Klein, L., and Jaeger, P. F. (2023). Navigating the Pitfalls of Active Learning Evaluation: A Systematic Framework for Meaningful Performance Assessment.

MacKay, D. (1991). Bayesian Model Comparison and Backprop Nets. In *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann.

MacKay, D. J. C. (1992a). *Bayesian Methods for Adaptive Models*. PhD thesis, California Institut of Technology, Pasadena, California.

MacKay, D. J. C. (1992b). A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3):448–472.

Maddison, C. J., Tarlow, D., and Minka, T. (2015). A* Sampling.

Maddox, W. J., Benton, G., and Wilson, A. G. (2020). Rethinking Parameter Counting in Deep Models: Effective Dimensionality Revisited. *arXiv:2003.02139 [cs, stat]*.

Malinin, A. and Gales, M. (2018). Predictive Uncertainty Estimation via Prior Networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Maltz, D. and Ehrlich, K. (1995). Pointing the way: Active collaborative filtering. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 202–209, USA. ACM Press/Addison-Wesley Publishing Co.

Meister, C., Salesky, E., and Cotterell, R. (2020). Generalized Entropy Regularization or: There's Nothing Special about Label Smoothing.

Mortier, T., Bengs, V., Hüllermeier, E., Luca, S., and Waegeman, W. (2023). On the Calibration of Probabilistic Classifier Sets. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 8857–8870. PMLR.

Mucsányi, B., Kirchhof, M., and Oh, S. J. (2024). Benchmarking Uncertainty Disentanglement: Specialized Uncertainties for Specialized Tasks.

Mukhoti, J. and Kirsch, A. (2023). Deep Deterministic Uncertainty: A New Simple Baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24384–24394.

Müller, R., Kornblith, S., and Hinton, G. (2020). When Does Label Smoothing Help?

Murphy, A. H. and Winkler, R. L. (1977). Reliability of Subjective Probability Forecasts of Precipitation and Temperature. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 26(1):41–47.

Naeini, M. P., Cooper, G., and Hauskrecht, M. (2015). Obtaining Well Calibrated Probabilities Using Bayesian Binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. (2019). Deep Double Descent: Where Bigger Models and More Data Hurt.

Nakkiran, P., Venkat, P., Kakade, S., and Ma, T. (2021). Optimal Regularization Can Mitigate Double Descent.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. PhD thesis, Springer New York, New York, NY.

Neal, R. M. and Hinton, G. E. (1998). A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants. In Jordan, M. I., editor, *Learning in Graphical Models*, pages 355–368. Springer Netherlands, Dordrecht.

Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images.

Nguyen, A. T., Lu, F., Munoz, G. L., Raff, E., Nicholas, C., and Holt, J. (2022). Out of Distribution Data Detection Using Dropout Bayesian Neural Networks.

Nixon, J., Dusenberry, M., Jerfel, G., Nguyen, T., Liu, J., Zhang, L., and Tran, D. (2020). Measuring Calibration in Deep Learning.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., and Snoek, J. (2019). Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift.

Pal, N. R., Bezdek, J. C., and Hemasinha, R. (1993). Uncertainty measures for evidential reasoning II: A new measure of total uncertainty. *International Journal of Approximate Reasoning*, 8(1):1–16.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training Recurrent Neural Networks.

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., and Hinton, G. (2017). Regularizing Neural Networks by Penalizing Confident Output Distributions.

Pfau, D. (2013). A Generalized Bias-Variance Decomposition for Bregman Divergences. *Unpublished manuscript*.

Platt, J. (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Adv. Large Margin Classif.*, 10.

Quiñonero-Candela, J., editor (2009). *Dataset Shift in Machine Learning*. Neural Information Processing Series. MIT Press, Cambridge, Mass.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass.

Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., DePristo, M. A., Dillon, J. V., and Lakshminarayanan, B. (2019). Likelihood Ratios for Out-of-Distribution Detection.

Ritter, H., Botev, A., and Barber, D. (2018). A SCALABLE LAPLACE APPROXIMATION FOR NEURAL NETWORKS. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge.

Sale, Y., Bengs, V., Caprio, M., and Hüllermeier, E. (2024). Second-Order Uncertainty Quantification: A Distance-Based Approach. In *Proceedings of the 41st International Conference on Machine Learning*, pages 43060–43076. PMLR.

Sale, Y., Caprio, M., and Höllermeier, E. (2023). Is the volume of a credal set a good measure for epistemic uncertainty? In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 1795–1804. PMLR.

Saul, L. K., Jaakkola, T., and Jordan, M. I. (1996). Mean Field Theory for Sigmoid Belief Networks. *Journal of Artificial Intelligence Research*, 4:61–76.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 253–260, New York, NY, USA. Association for Computing Machinery.

Senge, R., Bösner, S., Dembczyński, K., Haasenritter, J., Hirsch, O., Donner-Banzhoff, N., and Hüllermeier, E. (2014). Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. *Information Sciences*, 255:16–29.

Sensoy, M., Kaplan, L., and Kandemir, M. (2018). Evidential Deep Learning to Quantify Classification Uncertainty.

Sethuraman, J. (1994). A Constructive Definition of Dirichlet Priors. *Statistica Sinica*, 4(2):639–650.

Settles, B. (2012). *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Springer International Publishing, Cham.

Shaker, M. H. and Hüllermeier, E. (2021). Ensemble-based Uncertainty Quantification: Bayesian versus Credal Inference.

Silva Filho, T., Song, H., Perello-Nieto, M., Santos-Rodriguez, R., Kull, M., and Flach, P. (2023). Classifier calibration: A survey on how to assess and improve predicted class probabilities. *Machine Learning*, 112(9):3211–3260.

Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Smith, L. and Gal, Y. (2018). Understanding Measures of Uncertainty for Adversarial Example Detection.

Snelson, E. and Ghahramani, Z. (2005). Sparse Gaussian Processes using Pseudo-inputs. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the Inception Architecture for Computer Vision.

Tan, M. and Le, Q. V. (2020). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.

Tange, O. (2018). *GNU Parallel 2018*. Lulu. com.

Vaicenavicius, J., Widmann, D., Andersson, C., Lindsten, F., Roll, J., and Schön, T. (2019). Evaluating model calibration in classification. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 3459–3467. PMLR.

Valdenegro-Toro, M. (2021). Exploring the Limits of Epistemic Uncertainty Quantification in Low-Shot Settings. *LatinX in AI Research Workshop @ NeurIPS 2021, Sydney, Australia*, page 12.

Valdenegro-Toro, M. and Saromo, D. (2022). A Deeper Look into Aleatoric and Epistemic Uncertainty Disentanglement.

Walley, P. (1991). *Statistical Reasoning with Imprecise Probabilities*. London ; New York : Chapman and Hall.

Wang, C. (2024). Calibration in Deep Learning: A Survey of the State-of-the-Art.

Wang, D.-B., Feng, L., and Zhang, M.-L. (2021). Rethinking Calibration of Deep Neural Networks: Do Not Be Afraid of Overconfidence. In *Advances in Neural Information Processing Systems*, volume 34, pages 11809–11820. Curran Associates, Inc.

Wang, K., Zhang, D., Li, Y., Zhang, R., and Lin, L. (2017). Cost-Effective Active Learning for Deep Image Classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600.

Wang, X. and Aitchison, L. (2021). Bayesian OOD detection with aleatoric uncertainty and outlier exposure. *arXiv preprint arXiv:2102.12959*.

Wenzel, F., Roth, K., Veeling, B., Swiatkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. (2020). How Good is the Bayes Posterior in Deep Neural Networks Really? In *Proceedings of the 37th International Conference on Machine Learning*, pages 10248–10259. PMLR.

Widmann, D., Lindsten, F., and Zachariah, D. (2019). Calibration tests in multi-class classification: A unifying framework. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Wilson, A. G. and Izmailov, P. (2020). Bayesian Deep Learning and a Probabilistic Perspective of Generalization. *arXiv:2002.08791 [cs, stat]*.

Wilson, A. G. and Izmailov, P. (2021). Deep Ensembles as Approximate Bayesian Inference. https://cims.nyu.edu/~andrewgw/.

Wilson, A. G. and Nickisch, H. (2015). Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP).

Wimmer, L., Sale, Y., Hofman, P., Bischl, B., and Hüllermeier, E. (2023). Quantifying Aleatoric and Epistemic Uncertainty in Machine Learning: Are Conditional Entropy and Mutual Information Appropriate Measures?

Wolpert, D. and Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Yang, G. and Schoenholz, S. S. (2017). Mean Field Residual Networks: On the Edge of Chaos.

Yang, J., Wang, P., Zou, D., Zhou, Z., Ding, K., Peng, W., Wang, H., Chen, G., Li, B., Sun, Y., Du, X., Zhou, K., Zhang, W., Hendrycks, D., Li, Y., and Liu, Z. (2022). OpenOOD: Benchmarking Generalized Out-of-Distribution Detection.

Yang, J., Zhou, K., Li, Y., and Liu, Z. (2024). Generalized Out-of-Distribution Detection: A Survey.

Zadrozny, B. and Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 609–616, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham. Springer International Publishing.

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. (2018a). Noisy Natural Gradient as Variational Inference. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5852–5861. PMLR.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018b). Mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. (2020). Random Erasing Data Augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):13001–13008.