Bayesian Neural Networks for Out of Distribution Detection

by

John Mitros

This dissertation is submitted to University College Dublin in fulfilment of the requirements for the degree of Doctor of Philosophy

School of Computer Science

Submission: December, 2021
Defence: March, 2022
Corrections: May, 2022

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Empirical studies have demonstrated that point estimate deep neural networks despite being expressive estimators capturing rich interactions between covariates, nevertheless, exhibit high sensitivity in their predictions leading to overconfident misclassifications due to changes in the underlying representation of data distributions. This implication lead us to study the problem of out-of-distribution detection in identifying and characterising out-of-distribution inputs. This phenomenon has real world implications especially in high-stake applications where it is undesirable and often prohibitive for an estimator to produce overconfident misclassified estimates.

Alternatively, Bayesian models present a principled way of quantifying uncertainty over predictions represented in the estimator's parameters but at the same time they pose challenges when applied to large high dimensional datasets due to computational constraints requiring estimating high dimensional integrals over a large parameter space. Moreover, Bayesian models among others present properties leading to simple and intuitive formulation and interpretation of the underlying estimator.

Therefore, we propose to exploit this synergy between Bayesian inference and deep neural networks for out-of-distribution detection. This synergy leads to Bayesian neural networks exhibiting the following benefits (i) providing efficient and flexible neural network architectures applicable to large high dimensional datasets, (ii) estimating the uncertainty over the predictions captured in the predictive posterior distribution via Bayesian inference.

We validate our findings empirically across a number of datasets and performance metrics indicating the efficacy of the underlying methods and estimators presented in regard to calibration, uncertainty estimation, out-of-distribution detection, detection of corrupted adversarial inputs and finally the effectiveness of the proposed contrastive objectives for out-of-distribution detection. We hope that the methods and results presented here reflect the importance of how brittle an estimator can be due to discrepancies between train and test distribution leading to real world implications of particular interest to reliable and secure machine learning.

The algorithmic advances and research questions presented in this dissertation extend the domains of out-of-distribution detection and robustness against ambiguous inputs, in addition to exploring auxiliary information that can be incorporated during training. The resulting estimators overall are high dimensional exhibiting efficient detection.

Statement of Authorship

I hereby certify that the submitted work is my own work, was completed while registered as a candidate for the degree stated on the Title Page, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work.

Signature _____

ACKNOWLEDGEMENTS

This dissertation can be considered as a byproduct of the interactions and support provided by many interesting individuals with whom I had the pleasure to interact with during this journey. First, I would like to thank my supervisor for his support in allowing me to pursue my own research interests and kindly forgiving my foolish mistakes throughout my PhD.

The Insight Centre for Data Analytics in University College Dublin (UCD) fosters a vibrant, diverse and energetic research environment including all the amenities for interesting and constructive ideas to flourish. I deem myself privileged for the time spent in that environment and most importantly being in the presence of such kind and intelligent colleagues leading to interesting conversations about life and research.

I enjoyed reminiscing about life decisions and the implications and hurdles of pursuing a PhD at an advanced age in the company of Francesca Madia Mele, Gevorg Poghosyan, Ellen Rushe and Weipeng Huang. Furthermore, I would like to thank Antonio Bevilacqua and Gevorg Poghosyan for their insightful feedback during my transfer assessment presentation and for taking the time to read the proposal. Moreover, I had a great time collaborating and discussing a range of topics with Arjun Pakrashi, Ellen Rushe, and Mehran Hossein Zadeh Bazargani.

Thank you, Antonio Bevilacqua, Gevorg Poghosyan, Arjun Pakrashi, Séverin Gsponer and Francesca Madia Mele for your company while drinking our sorrows away at the Clubhouse or over a great foosball game (at which I suck by the way), not to mention the numerous nice dinners which were a pleasant distraction from research when it was most needed.

Finally, I would like to thank my family for their perpetual love, encouragement, inspiration and support all these years, without them this journey would not have been possible. We appreciate all your efforts to provide us with access to education even in the face of discrimination or financial constraints.

Last but definitely not least, I would like to thank all those inconspicuous heroes striving to maintain academia a toxic free environment beyond the privileged few by providing opportunities to a wide audience of socioeconomic backgrounds.

This acknowledgement is far from being exhaustive, apologies if you are not probably mentioned. I hope one day to pay forward this kindness by avoiding the same mistakes occasionally imposed upon us leading to unnecessary discomfort and lost life years.

Acronyms

Acronym	Description
DNN	Point estimate deep neural networks
BNN	Bayesian neural networks
ID	In-distribution
OOD	Out-of-distribution
MCD, MC-Dropout	Stochastic Monte-Carlo sampling of parameters
SWAG	Stochastic weight averaging of Gaussian parameters
JEM	Joint energy based model
DPN	Dirichlet prior networks
GP	Gaussian process
GAN	Generative adversarial network

Symbols

Notation	Description
$\mathcal{X}$	Input space
$\mathcal{Y}$	Target space
S	A data sample or sequence or dataset
P	Probability distribution
$\mathcal{P}$	A collection of probability distributions
H	Entropy of estimates
I	Mutual information of estimates
KL	Kullback-Leibler divergence
$\mathcal{F}$	Hypothesis class
$\mathcal{A}$	Learning algorithm
L	Loss function or objective
$\mathcal{D}$	n-fold product space
$\mathcal{E}$	Expected error on unknown test data distribution
$\mathbb{1}$	Indicator function equals to 1 if proposition is true else 0

CHAPTER

ONE

INTRODUCTION

Deep neural networks (DNN) are flexible and powerful estimators emitting a hierarchical structure composed of computational units at multiple levels. Even though DNN models typically outperform conventional estimators, their estimates can be unreliable when presented with "ambiguous inputs" leading to concerns over security implications in high stakes applications (Alemi et al., 2018; Guo et al., 2017; Hill, 2019; Kumar et al., 2019; Marin et al., 2012).

To clarify the term "ambiguous inputs" we have devised the following example from supervised learning. Given a neural network estimator trained on a binary classification problem to discriminate between cats and dogs with high accuracy (Figure 1.1), what should be the expected response of this estimator when presented with a novel category? (e.g. an image of a bird, Figure 1.2).

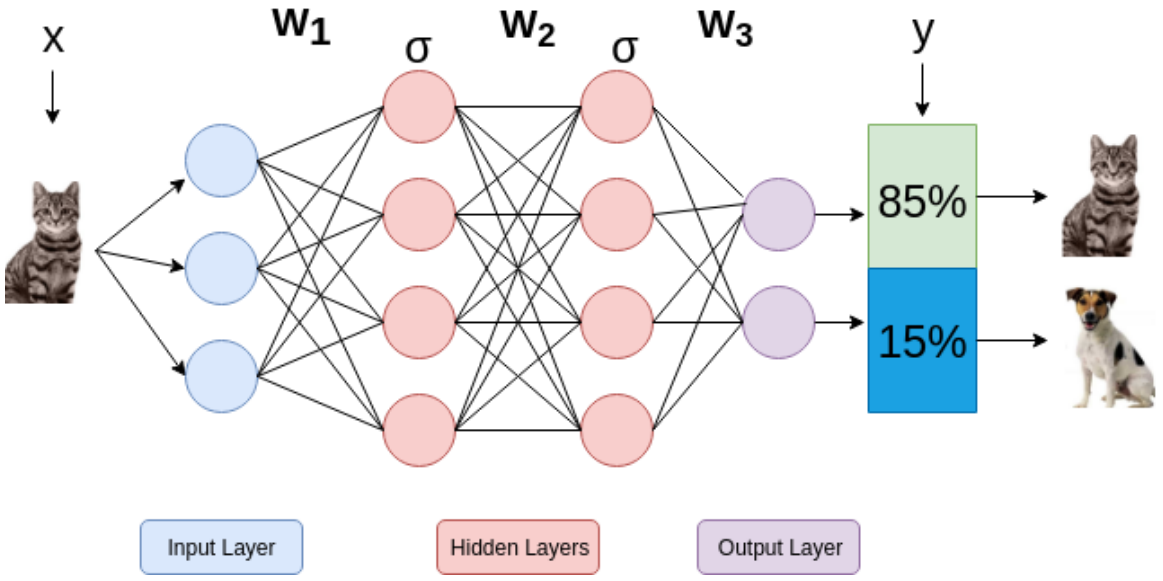


Figure 1.1: Neural network estimator trained to distinguish cats from dogs.

A valid assumption would be that the bird image would be assigned equal probability of each category. But, since the estimator was not exposed to birds before during training it implies that we do not have any control over its behaviour for such novel categories, often leading to an undefined behaviour of the estimator (Nguyen et al., 2015).

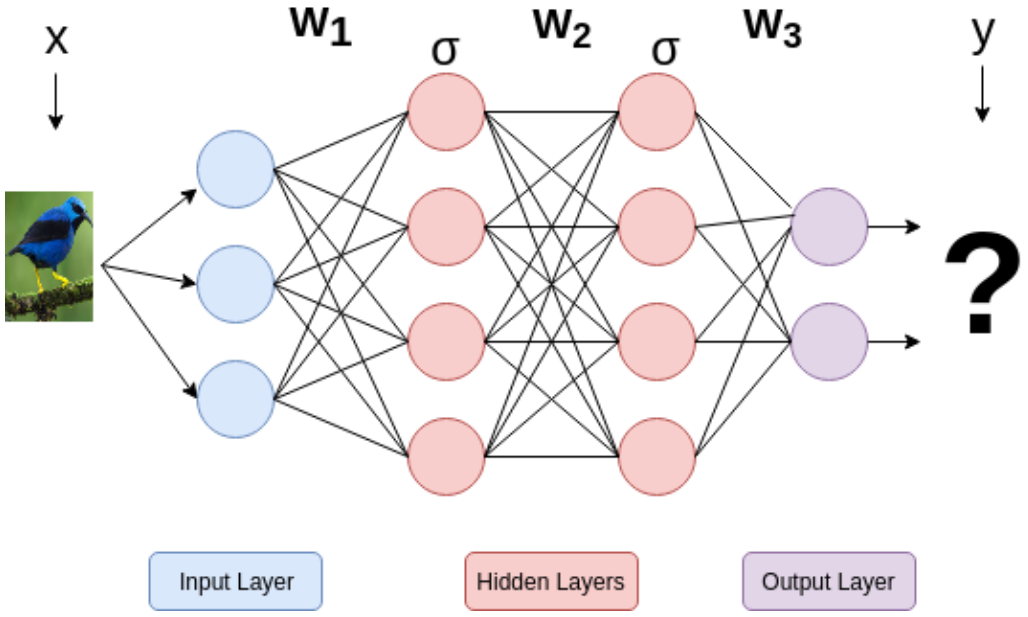


Figure 1.2: Undefined behaviour of estimator for novel inputs.

This example is based on the assumption that novel categories not seen during training must be distinctive and conceptually different from the already existing categories in the train set. In such situations the underlying assumptions of a priori knowledge regarding the unknown test distribution can be categorised into: (i) the estimator having access to a small size labelled test data; (ii) the estimator having access to a large size unlabelled test data; (iii) the estimator having access to a small size unlabelled test data; (iv) the estimator having no access or knowledge about the test data (Liu et al., 2018; Qiao et al., 2020; Sun et al., 2020).

In fact modern neural networks exhibit their lack of ability to handle novel inputs in sometimes startling ways. We perform the following experiment (similar to Gal and Ghahramani (2016)) to illustrate this. We train an estimator on the following categories *dog*, *cat*, *automobile*, *horse* and *bird*. Then, for a data point randomly sampled from the test distribution we rotate it by 10 degree increments between an angle of $0 \leq \theta \leq 180$ degrees and record the estimator's prediction after each rotation. These predictions are depicted on the y-axis in Figure 1.3, where the x-axis represents the amount by which the test image has been rotated (in 10 degree increments). Notice how the estimator is able to predict the true label (i.e. "dog") with approximately 100% confidence for an

angle $\theta \leq 30^o$ while for an angle $\theta > 30^o$ it misclassifies the test image to one of the remaining categories ("cat", "automobile", "horse", or "bird") with high confidence.

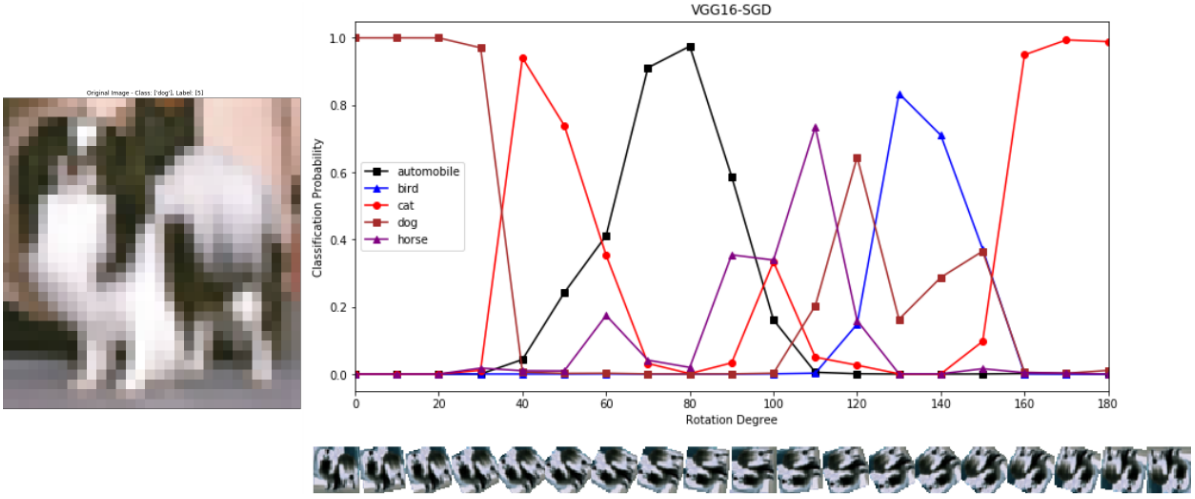


Figure 1.3: Example of in-distribution (ID) shift resulting in ambiguous inputs.

This example illustrates the fact that although the train and test data are derived from the same distribution nevertheless the relationship between predictors and targets (i.e. inputs and outputs) might have changed causing the estimator to misclassify with high confidence. As a consequence this estimator would be characterised as unreliable since it does not provide any uncertainty over its estimates (Schulam and Saria, 2019).

Even though there exist numerous ways to retrieve the uncertainty of an estimator over its estimates, for simplicity, in the remainder of this dissertation with the term uncertainty we will often be referring to the entropy of the predictive posterior distribution since it provides interesting connections among mutual information, differential entropy and relative entropy (i.e. Kullback-Leibler divergence) which can also be used to express and quantify uncertainty or the inherent randomness in the estimates of an estimator.

With slight abuse of terminology the examples above can be described as *out-of-distribution* (OOD) detection (Hendrycks and Gimpel, 2017). OOD is often used to describe phenomena such as distribution shift or subtle changes in the data distribution (e.g. train distribution $\neq$ test distribution). One of the main challenges regarding the problem of OOD detection, is that it violates the identical and independently distributed (i.i.d) assumption (Figure 1.4), thus, rendering classic learning theory inapplicable (definitions deferred in Chapter 2).

Figure 1.5 presents an illustrative example of OOD data in a multiclass classification setting. The in-distribution (ID) data are the *tori*, *squares*, and *triangles*, while the OOD data are the *crosses*. The question of interest then becomes whether it is possible to train

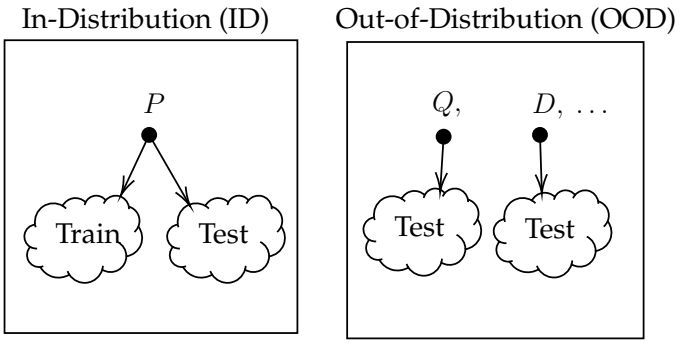


Figure 1.4: Violation of the i.i.d assumption, OOD data sampled from distributions Q, D different from P in ID data.

an estimator only on in-distribution data and at test time detect the OOD data? Notice that the difficulty in this example lies within the fact that the OOD data encompasses the ID data, therefore, any max-margin estimator trained on this dataset will evidently misclassify a portion of OOD data as in-distribution.

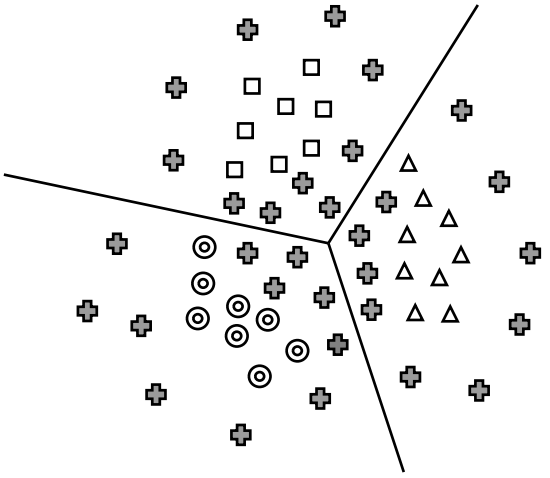


Figure 1.5: Example of OOD data (grey crosses) far from and in-between ID data.

The overall aim of this dissertation is to investigate reliable deep learning methods that are robust against ambiguous inputs. In that endeavour, we examine among other things, the role of calibration in overconfident predictions, how to quantify uncertainty for OOD data utilising Bayesian neural networks, the effect of adversarial examples on the predictions of an estimator and whether these examples can be detected as OOD, and finally the role of regularisation and auxiliary information in constructing estimators that are robust against OOD data.

Calibrating Estimators

Often modern deep learning estimators are hindered by their poor calibration. Guo et al. (2017) demonstrated that modern neural networks suffer from poor calibration influenced by factors such as their depth (# layers), width (# neurons per layer), weight decay (i.e. regularisation) and batch normalisation (i.e. standardising input data). Prior work by Niculescu-Mizil and Caruana (2005) arrived at the same conclusion for a group of classic estimators that also suffer from poor calibration. Moreover, it has been shown that calibration properties of an estimator also decay with distribution shift and choice of architecture (Minderer et al., 2021) leading to over-confident predictions.

Informally stated, a poorly calibrated estimator outputs probabilities that are either over-confident or under-confident, similarly, a calibrated estimator outputs probabilities that are neither over-confident nor under-confident. In essence, calibration states that the probabilities obtained by the model should reflect the true likelihood of the ground truth (Dawid, 1982; Murphy and Epstein, 1967). For instance, given 100 predictions each with 80% confidence then 80 of them should be classified correctly.

A poorly calibrated estimator often does not accurately reflect either the true likelihood of the ground truth predicted category or the estimated predictive uncertainty, leading to impaired OOD detection which is essential for the safe application of neural networks. Although different notions of calibration exist in the literature (Vaicenavicius et al., 2019; Widmann et al., 2019), in this dissertation we will mostly be concerned with the confidence of the top predicted category for each data point (Guo et al., 2017; Nixon et al., 2019).

Quantifying Predictive Uncertainty Using Bayesian Neural Networks

Unfortunately, deep learning models do not innately possess the ability to characterise OOD data with high uncertainty (Nalisnick et al., 2019; Schulam and Saria, 2019). By definition neural networks are discriminative estimators representing a parametric class of functions expressed as a composition of a number of nonlinear functions also known as layers

$$f_\theta(x) = W_N \, \phi_N \left(W_{N-1} \ldots \phi_2 \left(W_2 \, \phi_1 \left(W_1 \, x \right) \right) \right). \tag{1.1}$$

where $\theta = \{W_n\}_{n=1}^{N}$ defines the model parameters and ϕ denotes a nonlinear function.

Neural networks provide a deterministic output for a given input because their parameters express a point estimate for each input. We refer to this kind of neural network as a point estimate neural network and for simplicity abbreviate it as DNN. As such, it is non-trivial to properly equip neural networks with uncertainty estimation resulting in trustworthy predictions for OOD data (Ovadia et al., 2019). One approach would be to provide a probabilistic formulation of neural networks expressing a distribution over their estimates such that the spread of the distribution would characterise the certainty or lack thereof of the estimator's predictions.

Fortunately, Bayesian neural networks (BNN) (Neal, 1996; Wang and Yeung, 2016) provide such a predictive distribution which could be utilised to identify OOD data. BNNs specify a distribution over the estimator's parameters and estimating this posterior distribution permits them to capture uncertainty in their predictions.

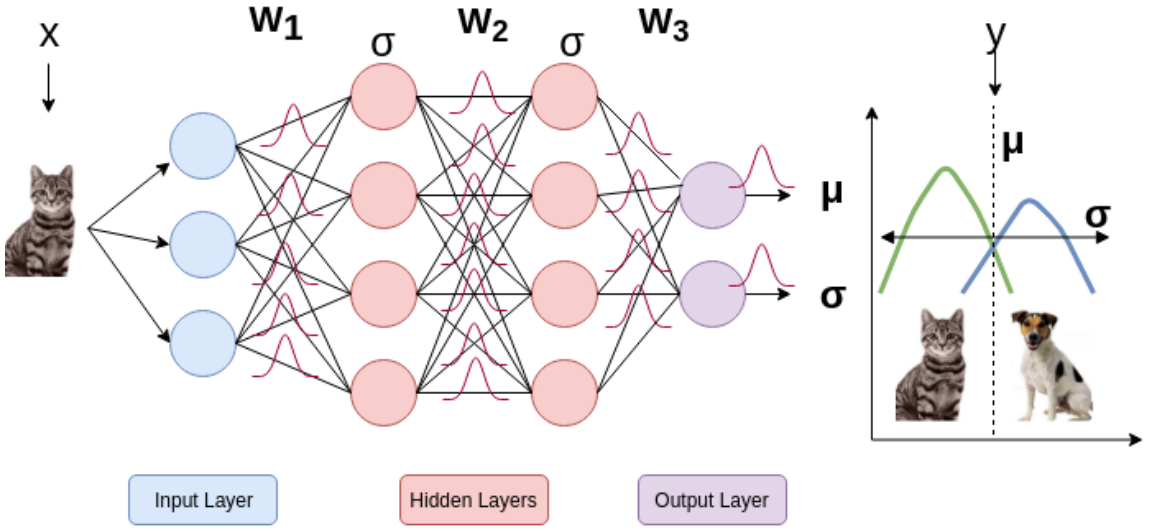


Figure 1.6: Example of Bayesian neural network.

Another difference is that in a BNN the parameters represent a source of randomness expressed as random variables, whereas, in a DNN the parameters are considered fixed with randomness emitted from the actual data. In Figure 1.6 the random variables correspond to $\theta = \{W_1, W_2, W_3\}$ and quantities μ, σ can be estimated from the input x.

Given a training set $S = \{(x_i, y_i)\}_{i=1}^{n}$ of n input and target pairs we can construct a BNN through the choice of a likelihood function $p(S|\theta)$ (e.g. neural network f_θ) and a prior distribution over the estimator's parameters $p(\theta)$. The uncertainty over the predictions can then be capture by approximating the posterior distribution $p(\theta|S)$.

$$\overbrace{p(\theta|S)}^{\text{posterior}} = \frac{\overbrace{p(S|\theta)}^{\text{likelihood}}\overbrace{p(\theta)}^{\text{prior}}}{\underbrace{p(S)}_{\text{evidence}}}. \tag{1.2}$$

$$p(\theta|x,y) = \frac{p(y,\theta|x)}{p(y|x)} = \frac{p(y|x,\theta)p(\theta)}{\int_\theta p(y,\theta|x)d\theta}.$$

A BNN can be seen as a joint distribution of parameters θ and observed outputs y given inputs x. The posterior distribution indicates how likely a particular setting of parameters are after observing the training set. The likelihood expresses how likely the data are given a particular setting of parameters. Finally, the prior encodes belief with respect to the distribution of the parameters without seeing any data.

One difficulty in obtaining the posterior distribution is the *evidence* since it involves marginalising over all parameter settings (i.e. integral in Eq. 1.2). Usually this integral is intractable that is why we often resolve in approximations of the posterior distribution. The existing methods to approximate the posterior could be categorised into *sampling* or *local* approximations. Both categories are broadly recognised as approximate inference methods (Marin et al., 2012).

Bayesian principles for neural networks provide a framework for estimating uncertainty based on information pre and post data collection. This information is captured in the posterior denoting the probability distribution over the space of plausible answers given the observed data. It also enables sequential learning and reduces overfitting via Bayesian model averaging (Hoeting et al., 1999).

In general such principles have also been employed in deep learning but due to computational restrictions scaling to large datasets has been challenging resulting in less principled approximations (i.e. MC-Dropout) whose posterior approximation does not contain mass over the whole parameter space (Kirkpatrick et al., 2017).

Protecting Against Adversarial Examples

Adversarial examples describe meticulously crafted noise imperceptible to the human eye causing an estimator to produce overconfident misclassified predictions. This particular type of noise is also classified as ambiguous input. Both Bayesian neural networks (BNN) and point estimate deep neural networks (DNN) exhibit susceptibility to adversarial examples (Carlini and Wagner, 2017). Therefore, in this work we extend the efficacy of Bayesian neural networks from the OOD detection setting to the adversarial

domain by evaluating a number of Bayesian neural network estimators and examining their stability against adversarial attacks. Furthermore, we present an extension to Bayesian neural networks integrating them with adversarial defence techniques leading to estimators that overall produce stable predictions on adversarial inputs. Our findings with respect to the OOD detection ability of Bayesian neural networks with and without adversarial defence techniques on both natural and adversarial inputs on real world datasets, indicate that adversarial defence techniques provide protection against adversarial attacks and occasionally improve OOD detection to the expense of degraded in-distribution accuracy.

Regularising With Auxiliary Information for OOD Detection

Since adversarial defence techniques tend to degrade the in-distribution accuracy, to ameliorate this issue we propose two novel objectives to extend and augment any estimator by integrating auxiliary information inspired by contrastive learning (Oord et al., 2018). These objectives are devised for OOD detection utilising auxiliary information similarly to outlier exposure (Hendrycks et al., 2019a), allowing domain knowledge to be incorporated into the existing training framework for OOD detection. In addition, one can interpret these objectives as explicit regularisation of the unlabelled OOD data. We empirically evaluate the proposed objectives against state-of-the-art methods and datasets and present our findings indicating that estimators trained with these objectives exhibit efficient OOD detection on ambiguous inputs.

To summarise, this dissertation addresses the following limitations of modern neural networks – lack of confidence calibration and inability to detect OOD inputs – through uncertainty estimation and calibration via Bayesian neural networks, adversarial training and novel loss functions.

1.1 Research Questions

Throughout the development of this dissertation we have been intrigued and almost obsessed in understanding why neural networks fail in the presence of ambiguous inputs. Our goal has always been to understand these behaviours and failure modes of neural networks and hopefully provide intuitive mitigations. We are primarily interested in developing and deploying techniques that do not alter the underlying problem but as a consequence provide robustness against ambiguous inputs. Below we present

the main research questions (RQs) investigated throughout the duration of this dissertation:

- **RQ1**: *Are Bayesian neural networks better calibrated than point estimate neural networks?*
- **RQ2**: *Can Bayesian neural networks quantify uncertainty of ambiguous inputs better than point estimate neural networks?*
- **RQ3**:*Are Bayesian neural networks capable of detecting OOD inputs, and how do they compare with point estimate neural networks for this task?*
- **RQ4**: *Do Bayesian neural networks exhibit any robustness against adversarial inputs by default, and, if not can we make them more robust?*
- **RQ5**: *Is it possible to devise custom objectives in order to improve OOD detection ability in neural networks utilising auxiliary information as a form of regularisation?*

1.2 Contributions

The contributions claimed in this dissertation are:

- **CB1**: *We show that Bayesian neural networks are better calibrated than their point estimate counterparts and that there seems to be no correlation between calibration and uncertainty estimation ability of the underlying estimator.*
- **CB2**: *We show that Bayesian neural networks are indeed effective at uncertainty estimation and moreover can outperform point estimate neural networks in OOD detection.*
- **CB3**: *We show that adversarial defence techniques improve detection of OOD inputs in Bayesian neural networks while also withstanding against adversarial noise.*
- **CB4**: *We propose two custom objectives that can extend any estimator to improve OOD detection while utilising auxiliary information as an additional form of regularisation. We demonstrate that estimators trained using these objectives outperform a number of current approaches while maintaining competitive performance with the rest.*

1.3 Publications

Below we present a list of publications supporting the contributions mentioned above in addition to Table 1.1 that connects each research question with its own contribution supported by its publication.

- John Mitros and Brian Mac Namee. On the Validity of Bayesian Neural Networks for Uncertainty Estimation. In *Proceedings for the 27th Irish Conference on Artificial Intelligence and Cognitive Science*, (AICS), volume 2563, 2019.
- John Mitros, Arjun Pakrashi, and Brian Mac Namee. A Comparison of Bayesian Deep Learning for Out-of-Distribution Detection and Uncertainty Estimation. In *Proceedings of the 37th International Conference of Machine Learning Workshops*, (ICML), volume 119, 2020.
- John Mitros, Arjun Pakrashi, and Brian MacNamee. Ramifications of Approximate Posterior Inference for Bayesian Deep Learning in Adversarial and Out-of-Distribution Settings. In *Proceedings of the 16th European Conference on Computer Vision Workshops*, (ECCV), volume 12535, 2020.
- John Mitros and Brian Mac Namee. On the Importance of Regularisation & Auxiliary Information in OOD detection. In *Proceedings of the 28th International Conference on Neural Information Processing*, (ICONIP), CCIS series, volume 1517, 2021.

Table 1.1: List of research questions corresponding to each contribution supported by its publication and expanded upon on the chapter indicated below.

Research Question	Contribution	Chapter	Publication
RQ1 and **RQ2**	**CB1**	Chapter 3	Mitros and Mac Namee (2019)
RQ3	**CB2**	Chapter 4	Mitros et al. (2020a)
RQ4	**CB3**	Chapter 5	Mitros et al. (2020b)
RQ5	**CB4**	Chapter 6	Mitros and Mac Namee (2021)

1.4 Dissertation Structure

The remainder of this document is structured as follows. Chapter 2 outlines necessary background knowledge, presents related work and finally provides parallel formulations of the learning problem under OOD data. Chapter 3 presents results regarding calibration and uncertainty estimation in BNN related to **RQ1** and **RQ2**. Chapter 4 addresses research question **RQ3** related to the ability of Bayesian neural networks to detect OOD inputs. Chapter 5 investigates the robustness of Bayesian neural networks against adversarial inputs and addresses **RQ4**. Chapter 6 improves upon existing methods related to **RQ5** by introducing two new objectives for classification utilising auxiliary information to robustify predictions against ambiguous inputs. Finally, Chapter 7 summarises the work presented and describes directions for future work.

CHAPTER

TWO

BACKGROUND AND RELATED WORK

In this Chapter we present the necessary background information and related work in out-of-distribution (OOD) detection outlining the basic concepts, notation and definitions for calibration, uncertainty estimation, and Bayesian neural networks. We also present two formulations for the OOD detection setting presenting alternative perspectives of the same underlying problem. The first formulation states OOD detection as a problem of identifying and learning invariant covariates in the presence of distribution shift, whereas the second formulation presents OOD detection as a binary classification problem between in-distribution (ID) and out-of-distribution (OOD) data.

2.1 Confidence Calibration

Calibration scores have always been a subject of study in the machine learning community. Kuleshov and Liang (2015) produced calibrated probabilities for structured output spaces and for inputs derived from potentially adversarial sources in an online setting. Lakshminarayanan et al. (2017) introduced ensembles of randomly initialised neural networks producing aggregated calibration scores whereas Kuleshov et al. (2018) proposed a simple procedure for calibrating regression algorithms. Finally, Widmann et al. (2019) suggested hypothesis testing under different statistics to verify calibration properties of estimators and Gupta and Ramdas (2021) provided calibration guarantees for histogram binning independent of any distribution assumptions.

Essentially, an estimator f outputs a pair of a category and an associated vector of probabilities (y, p), for a given input x. Often the probability vector p is distorted resulting in misclassified estimates with high confidence. Distortions might arise due to (i) the sum of elements in the probability vector not adding up to 1, violating Kolmogorov's second axiom of probability theory, (ii) the accuracy of the estimator not matching its

confidence. Thus, an estimator f predicting a category y with confidence p is considered calibrated if the confidence matches its accuracy for the ground truth category, across all possible confidence levels q and categories y.

$$\underbrace{\mathbb{P}(Y = y \mid p = q)}_{\text{accuracy}} = \underbrace{q}_{\text{confidence}}, \ \forall q \in [0, 1], \ \forall y \in Y. \tag{2.1}$$

Definition 1 *(Confidence Calibration): Given an input space $X \in \mathcal{X} \subseteq \mathbb{R}^d$, a target space $Y \in \mathcal{Y} \subseteq \mathbb{Z}^+$ and an interval of possible confidence values $q \in [0 - 1]$ which represents the true ratio of observations in each category $y \in Y$ over the total observations in the sample size. Then, an estimator $f \in \mathcal{F}$ with $f : \mathcal{X} \to \mathcal{Y}$ is considered multicategory calibrated if for any category $y \in Y$ the predicted probability (i.e. accuracy) matches the confidence level q (i.e. the true ratio of data belonging to category y over all remaining data in the sample).*

$$\mathbb{P}(Y = \arg\max_y f_y(x) \mid \max_y f_y(x) = q) - q. \tag{2.2}$$

Subtracting the confidence q from both sides of Eq. 2.1 leads to the *calibration error*. Estimating the expectation over the *calibration error* provides the *expected calibration error* (ECE) (Guo et al., 2017).

$$\text{ECE} = \mathop{\mathbb{E}}_{\max_y f_y(x)} \left[\mathbb{P}(Y = \arg\max_y f_y(x) \mid \max_y f_y(x) = q) - q \right]. \tag{2.3}$$

We cannot directly estimate the probability inside the expectation since it involves the unknown ground truth joint distribution, therefore, we approximate Eq. 2.3 by discretising the probability interval $[0 - 1]$ into a fixed number of bins B, and, for each bin b assign the predicted probability $\max_y f_y(x)$, if its value is enclosed in that bin, over all data points n (Vaicenavicius et al., 2019).

$$\begin{aligned} \text{ECE} &= \sum_{b=1}^{B} \frac{|b|}{n} |\operatorname{acc}(b) - \operatorname{conf}(b)|. \\ \operatorname{acc}(b) &= \frac{1}{|b|} \sum_{i \in b} \mathbb{1}\left\{ Y_i = \arg\max_y f_y(x)_i \right\}. \\ \operatorname{conf}(b) &= \frac{1}{|b|} \sum_{i \in b} \max_y f_y(x)_i. \end{aligned} \tag{2.4}$$

Could calibration alleviate overconfidence in misclassified ambiguous inputs for deep learning models? Suppose we have a binary estimator $f : \mathcal{X} \to \{0, 1\}$ which outputs the predictions in Table 2.1. Notice that the accuracy of the estimator $\mathbb{P}(Y = 1 \mid \max_y f_y(x) = q)$ is the same as its confidence $q = 0.5$ on the ground truth category 1. That means that the estimator is perfectly calibrated but completely inaccurate since its predictions are as good as a random coin toss.

Table 2.1: Example of calibrated but inaccurate estimator.

$Y =$ True Category	0	0	0	0	0	1	1	1	1	1
$f(x) =$ Estimator's Prediction	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

Thus, calibration might not always guarantee an appropriate measure of confidence.

In this context next we present some common calibration methodologies in the literature.

Plat scaling (Platt, 1999) introduces calibration as a form of parametric estimation. The idea is based on training a logistic regression estimator on the predicted estimates of a model in order to retrieve calibrated confidence scores.

Histogram binning (Zadrozny and Elkan, 2001) assigns predictions to bins on the unit interval based on minimising an objective among different bins. To retrieve calibrated confidence scores it suffices to estimate the calibration score from the amount of predictions inside each bin matching the confidence for that particular bin. This non-parametric approach to calibration can accommodate large datasets.

Isotonic regression (Zadrozny and Elkan, 2002), produces calibrated confidence scores similar to histogram binning by learning a function minimising an objective between the predictions of an estimator and the corresponding categories.

Bayesian Binning Quantiles (BBQ) (Naeini et al., 2015) refers to an extension of histogram binning producing calibrated estimates based on marginalisation over different bins. BBQ performs Bayesian model averaging on the probabilities corresponding to each binning scenario. This process permits BBQ to compute a calibrated probability output for any prediction emitted by the estimator given a specific binning scenario and the validation dataset.

2.2 Uncertainty Estimation

Measures of uncertainty convey information regarding the amount of noise in the estimator's predictions. A noisy or corrupted test datum often deviates from the support of the training distribution (ID) leading to invalid conclusions derived from uncertain estimates. For instance, sources of uncertainty could be found in (i) *noisy data*, as a consequence of measurement imprecision or noisy targets of a dataset (i.e. inherent noise), (ii) *estimator parameters*, due to noise originating from the approximate posterior and not knowing a priori the optimal parameters explaining the observed data.

Definition 2 *(Uncertainty Estimation): Given an input space $X \in \mathcal{X} \subseteq \mathbb{R}^d$ and an output space $Y \in \mathcal{Y} \subseteq \mathbb{Z}^+$ then, an estimator $f \in \mathcal{F}$ mapping inputs to outputs $f : \mathcal{X} \to \mathcal{Y}$ can be considered uncertainty aware if its density function is concentrated around the groundtruth likelihood for distinct inputs $x, x' \sim P$ drawn from the same distribution P, while at the same time its density becomes uniformly low for inputs $x_{ood} \sim Q$ drawn from a different distribution Q deviating from the groundtruth likelihood, for $\epsilon, \delta > 0$.*

$$\mathbb{P}(|f(x) - f(x')| > \epsilon) < \delta \quad \wedge \quad \mathbb{P}(|f(x) - f(x_{ood})| > \epsilon) \geq 1 - \delta. \tag{2.5}$$

There are two types of uncertainty associated with the predictions of an estimator, aleatoric and epistemic uncertainty (Der Kiureghian and Ditlevsen, 2009). Aleatoric uncertainty typically originates from the randomness in the latent variables or due to the inherent noise in the target labels, and unfortunately observing more data will not affect it (Depeweg et al., 2018; Huang et al., 2021; Liu et al., 2019). Instead, epistemic uncertainty has its primary source of randomness stemming from the estimator's parameters θ due to the approximate posterior $q(\theta)$ and contrary to aleatoric uncertainty it is reducible by observing more data (Depeweg et al., 2018; Huang et al., 2021; Liu et al., 2019).

Thus, a test datum can be identified as ambiguous (i.e. outlier, out-of-distribution, noisy etc.) relative to the in-distribution training data if its epistemic uncertainty is high, or, it can be identified as inherently ambiguous due to insufficient information with respect to latent variables if its aleatoric uncertainty is high (Depeweg et al., 2018).

Regularly in classification tasks we can decompose predictive uncertainty as the sum of aleatoric and epistemic uncertainty (Amaral et al., 2014; Depeweg et al., 2018).

$$\underbrace{H(y|x,S)}_{\text{predictive uncertainty}} = \underbrace{I(y,\theta|x,S)}_{\text{epistemic uncertainty}} + \underbrace{\mathop{\mathbb{E}}_{\theta\sim p(\theta|S)}\left[H(y|x,\theta)\right]}_{\text{aleatoric uncertainty}}. \tag{2.6}$$

According to Gal (2016) the *predictive uncertainty* can be estimated as the entropy of the softmax output from the estimator $f(x)$, and *epistemic uncertainty* is the joint mutual information on predictions and parameters (Kendall and Gal, 2017). Whereas, *aleatoric uncertainty* originating from the inputs often represents the average entropy for a particular value of parameters (Amaral et al., 2014; Depeweg et al., 2018). In essence, epistemic uncertainty captures uncertainty in the parameters of an estimator, and, aleatoric uncertainty captures uncertainty with respect to the inherent noise in the input data.

2.2.1 Aleatoric Uncertainty

In order to estimate aleatoric uncertainty we have to replace the intractable exact posterior $p(\theta|S)$ with the approximate variational density $q(\theta)$ and finally obtain m samples of parameters θ_m to be utilised in inference (Amaral et al., 2014; Depeweg et al., 2018).

$$\begin{aligned}
\mathop{\mathbb{E}}_{\theta\sim p(\theta|S)}\left[H(y|x,\theta)\right] &= \mathop{\mathbb{E}}_{\theta\sim p(\theta|S)}\left[-\sum_{k=1}^{K} p(y=k|x,\theta)\ \ln p(y=k|x,\theta)\right] \\
&= -\int p(\theta|S)\sum_{k=1}^{K} p(y=k|x,\theta)\ \ln p(y=k|x,\theta)\ d\theta \\
&= -\int q(\theta)\sum_{k=1}^{K} p(y=k|x,\theta)\ \ln p(y=k|x,\theta)\ d\theta \\
&= -\frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K} p(y=k|x,\theta_m)\ \ln p(y=k|x,\theta_m).
\end{aligned} \tag{2.7}$$

2.2.2 Predictive Uncertainty

To estimate the predictive uncertainty we follow the same steps as in the aleatoric uncertainty by expanding the pedictive poterior $p(y = k|x, S)$ as a marginalisation over parameters and replacing the intractable exact posterior $p(\theta|S)$ with the approximate variational density $q(\theta)$. Finally, obtaining m samples of parameters θ_m to be utilised in inference (Amaral et al., 2014; Depeweg et al., 2018).

$$
\begin{aligned}
H(y \mid x, S) &= -\sum_{k=1}^{K} p(y = k|x, S)\ \ln p(y = k|x, S) \\
&= -\sum_{k=1}^{K} \mathop{\mathbb{E}}_{\theta \sim p(\theta|S)} [p(y = k|x, \theta)]\ \ln \mathop{\mathbb{E}}_{\theta \sim p(\theta|S)} [p(y = k|x, \theta)] \\
&= -\sum_{k=1}^{K} \frac{1}{M} \sum_{m=1}^{M} p(y = k|x, \theta_m)\ \ln \frac{1}{M} \sum_{m=1}^{M} p(y = k|x, \theta_m).
\end{aligned} \tag{2.8}
$$

2.2.3 Epistemic Uncertainty

Epistemic uncertainty can be decomposed similarly to predictive and aleatoric uncertainty estimated from Eq. 2.7 and 2.8 (Amaral et al., 2014; Depeweg et al., 2018). Decomposing and rearranging terms we get.

$$
\begin{aligned}
I(y, \theta|x, S) &= H(y|x, S) - \mathop{\mathbb{E}}_{\theta \sim p(\theta|S)} [H(y|x, \theta)] \\
&= \frac{1}{M} \sum_{m=1}^{M} \sum_{k=1}^{K} p(y = k|x, \theta_m)\ \ln p(y = k|x, \theta_m) \\
&\quad - \sum_{k=1}^{K} \frac{1}{M} \sum_{m=1}^{M} p(y = k|x, \theta_m)\ \ln \frac{1}{M} \sum_{m=1}^{M} p(y = k|x, \theta_m).
\end{aligned} \tag{2.9}
$$

2.3 Approximate Bayesian Inference

The inference problem in Bayesian neural networks amounts to estimating the posterior distribution $p(\theta|S)$ (Eq. 1.2) capturing the estimator's uncertainty. Doing so is necessary in order to derive the *predictive* distribution $p(y|x, S)$, this is the conditional probability of y given x (Marin et al., 2012).

$$\begin{aligned} p(y|x, S) &= \int_{\theta} p(y|x, \theta)\ p(\theta|S)\ d\theta \\ &= \mathop{\mathbb{E}}_{p(\theta|S)} \left[p(y|x, \theta) \right]. \end{aligned} \tag{2.10}$$

The posterior *predictive* distribution can also be interpreted as the likelihood given parameters $p(y|x, \theta)$, where each parameter configuration is weighted by the posterior $p(\theta|S)$. The likelihood reflects the plausibility of the training data S under a particular parameter setting θ. For instance, if S contains independent data points then the likelihood is the product of conditional probabilities for each data point given parameters θ (Christian, 2014).

$$p(y|x, \theta) = \prod_{i=1}^{n} p(y_i|x_i, \theta). \tag{2.11}$$

Estimating marginal likelihoods often requires calculating high dimensional integrals resulting in analytically intractable posterior and hence predictive distributions. In such situations or when a closed form solution does not exist we resolve in approximations. In order to estimate the posterior parameter $p(\theta|S) \approx p(y|x, \theta)$ we frequently optimise the following objectives (Alquier, 2020).

$$\text{MAP} = \arg\max_{\theta} \sum_{i=1}^{n} \ln p(y_i|x_i, \theta) + \ln p(\theta). \tag{2.12}$$

$$\text{MLE} = \arg\max_{\theta} \sum_{i=1}^{n} \ln p(y_i|x_i, \theta). \tag{2.13}$$

Both objectives can be optimised by gradient based algorithms. The first objective is the maximum a posteriori (MAP) estimation while the second is the maximum likelihood estimation. Moreover, since θ is high dimensional depending on the underlying neural network architecture and depth, we resolve in approximating the predictive posterior

distribution via Monte Carlo sampling, which is a method for drawing samples from the posterior $\theta_m \sim p(\theta|S)$.

$$p(y|x, S) = \frac{1}{m} \sum_{i=1}^{m} p(y|x, \theta_i). \tag{2.14}$$

A common approach to simulate samples from the posterior is MC-Dropout (Gal and Ghahramani, 2016). MC-Dropout provides an alternative interpretation of a regularisation mechanism (i.e. Dropout (Srivastava et al., 2014)) as Monte Carlo sampling. Initially Dropout was proposed to prevent overfitting by eliminating parameters correlated against the same data covariates during training of a deep learning model. Instead, MC-Dropout utilises the same regularisation mechanism during inference (i.e. during prediction on the test set) to approximate the marginal posterior distribution by simulation of different parameter settings from a trained estimator. Given a trained neural network estimator f_θ we draw m samples from the posterior $\theta_m \sim p(\theta|S)$ such that the histogram of θ_m approximates the true posterior density and the posterior mean value. This avoids the need of any integration.

2.3.1 Dropout as Approximate Bayesian Inference

MC-Dropout (Gal and Ghahramani, 2016) is based on prior work of Damianou and Lawrence (2013) which established a connection between neural networks with dropout and Gaussian Processes (GP). It provides an interpretation of dropout during inference as approximate Bayesian inference. Given a dataset (X, Y) the posterior over the GP is formulated as

$$F|X \sim \mathcal{N}(0, \mathbf{K}(X, X')). \tag{2.15}$$

$$Y|F \sim \mathcal{N}(F, I). \tag{2.16}$$

$$y|Y \sim \text{Categorical}\left(\frac{e^{y_i}}{\sum_{j \neq i} e^{y_j}}\right). \tag{2.17}$$

where y denotes a target label, F denotes distributions on functions that the GP represents, and I denotes the identity matrix. An integral part of the GP is the choice of the covariance matrix $\mathbf{K}$ representing the similarity between two inputs as a scalar value. The connection between neural networks and Gaussian processes is established by ap-

proximating the kernel $\mathbf{K}$ in Eq. 2.15 with the choice of a neural network estimator formulated below as a Gaussian kernel.

$$\int p(\theta)\ \phi(\theta^T x)\ \phi(\theta^T x')\ d\theta. \tag{2.18}$$

Because the integral is intractable we resolve in Monte Carlo sampling to approximate it, hence, the name Monte Carlo Dropout.

$$\mathbf{K} = \frac{1}{m} \sum_{i=1}^{m} \phi(\theta_i^T x)\ \phi(\theta_i^T x'). \tag{2.19}$$

Let us now consider a one hidden layer neural network with dropout $(\theta_2^T M_2)\phi(x(\theta_1^T M_1))$ where $M_1, M_2 \sim$ Bernoulli(p_1, p_2) are samples from a Bernoulli distribution with parameters p_1 and p_2 representing dropout masks. Utilising the approximate kernel one can express the parameters $\theta_{1,2}$ as

$$\theta_{1,2} = M_{1,2}\,(\mathbf{K}_{1,2} + \sigma\,\epsilon_{1,2})(1 - M_{1,2})\,\sigma\,\epsilon_{1,2}. \tag{2.20}$$

resembling the neural network formulation with dropout. Finally, to retrieve the posterior predictive distribution we simulate Monte Carlo samples over subsets of parameters of the trained estimator at test time from the different dropout masks aggregating the predictions.

2.3.2 Variational Methods as Approximate Bayesian Inference

Variational inference (Jordan et al., 1999) is an alternative strategy to Markov chain Monte Carlo (MCMC) sampling in order to approximate the target posterior densities for Bayesian models with a family of approximate parameterised densities. The key idea behind variational inference is to utilise optimisation rather than sampling, transforming the underlying inference problem into an optimisation problem.

In variational inference the intractable posterior $p(\theta|S)$ is approximated with a simpler density $q(\theta)$ over the estimator's parameters. To find a suitable density $q(\theta)$ we minimise the Kullback-Leibler (KL) divergence over this density to the exact posterior. The retrieved density $q(\theta)$ from the optimisation process is then utilised in the estimator's predictions rather than the exact posterior (Hoffman et al., 2013).

$$
\begin{aligned}
KL(\underbrace{\overbrace{q(\theta)}^{\text{approximate}}}_{\text{posterior}} \| \underbrace{\overbrace{p(\theta|S)}^{\text{exact}}}_{\text{posterior}}) &= \mathop{\mathbb{E}}_{q(\theta)}\left[\ln \frac{q(\theta)}{p(\theta|S)}\right] \\
&= \mathop{\mathbb{E}}_{q(\theta)}\left[\ln q(\theta)\right] - \mathop{\mathbb{E}}_{q(\theta)}\left[\ln p(\theta|S)\right] \\
&= \mathop{\mathbb{E}}_{q(\theta)}\left[\ln q(\theta)\right] - \mathop{\mathbb{E}}_{q(\theta)}\left[\ln \frac{p(S|\theta)p(\theta)}{p(S)}\right] \\
&= \mathop{\mathbb{E}}_{q(\theta)}\left[\ln q(\theta)\right] - \mathop{\mathbb{E}}_{q(\theta)}\left[\ln p(S|\theta)p(\theta)\right] + \overbrace{\mathop{\mathbb{E}}_{q(\theta)}\left[\ln p(S)\right]}^{\int_\theta q(\theta)\ln p(S)d\theta = \ln p(S)\cancelto{1}{\int_\theta q(\theta)d\theta}} \\
&= \mathop{\mathbb{E}}_{q(\theta)}\left[\ln q(\theta)\right] - \mathop{\mathbb{E}}_{q(\theta)}\left[\ln p(S|\theta)p(\theta)\right] + \ln p(S) \\
\ln p(S) &= -\mathop{\mathbb{E}}_{q(\theta)}\left[\ln q(\theta)\right] + \mathop{\mathbb{E}}_{q(\theta)}\left[\ln p(S|\theta)p(\theta)\right] + \underbrace{KL(q(\theta)\|p(\theta|S)}_{\text{intractable but positive} \geq 0} \\
\ln p(S) &\geq \underbrace{-\mathop{\mathbb{E}}_{q(\theta)}\left[\ln q(\theta)\right] + \mathop{\mathbb{E}}_{q(\theta)}\left[\ln p(S|\theta)p(\theta)\right]}_{\text{Evidence Lower Bound}}.
\end{aligned}
\tag{2.21}
$$

Thus, minimising the KL divergence amounts to maximising the evidence lower bound (ELBO) objective (Blei et al., 2017).

$$
\begin{aligned}
ELBO(q) &= -\mathop{\mathbb{E}}_{q(\theta)}\left[\ln q(\theta)\right] + \mathop{\mathbb{E}}_{q(\theta)}\left[\ln p(S|\theta)\right] + \mathop{\mathbb{E}}_{q(\theta)}\left[\ln p(\theta)\right] \\
&= \mathop{\mathbb{E}}_{q(\theta)}\left[\ln p(S|\theta)\right] - \mathop{\mathbb{E}}_{q(\theta)}\left[\ln q(\theta)\right] + \mathop{\mathbb{E}}_{q(\theta)}\left[\ln p(\theta)\right] \\
&= \underbrace{\mathop{\mathbb{E}}_{q(\theta)}\left[\ln p(S|\theta)\right]}_{\text{expected reconstruction error}} - \underbrace{\mathop{\mathbb{E}}_{q(\theta)}\left[\ln \frac{q(\theta)}{p(\theta)}\right]}_{KL(\underbrace{q(\theta)}_{\text{approx. posterior}} \| \underbrace{p(\theta)}_{\text{prior}})}.
\end{aligned}
\tag{2.22}
$$

To obtain the optimal parameters we can resolve in stochastic optimisation methods (i.e. SGD) to maximise ELBO. The optimal parameters provide access to a suitable candidate density $q(\theta) \approx p(\theta|S)$ close to the exact posterior from which we can sample during inference to obtain the predictive posterior distribution (Paisley et al., 2012).

A common assumption regarding the retrieved density $q(\theta)$ after optimising KL is that it factorises into a product of distributions treating the estimator's parameters as independent variables. Thus, if $q(\theta)$ was following a Gaussian distribution then during inference we would sample from a Normal distribution rather than the exact posterior (Matthew and David, 2015).

Both sampling methods (based on Monte Carlo simulations similar to MC-Dropout) and variational inference are different approaches to solving the same problem, although, variational inference tends to be faster and easier to scale to large data compared to sampling. That is the reason it is often employed in Bayesian neural networks (e.g. SWAG (Maddox et al., 2019) and DPN (Malinin and Gales, 2018)).

2.4 Bayesian Neural Networks

Approximate Bayesian inference facilitates the synergy between optimisation and inference mechanisms introducing the capability to train large neural network estimators through gradient based algorithms and obtain a predictive posterior distribution during inference. A culmination of this combination leads to Bayesian neural networks (Wang and Yeung, 2016). The remainder of this section describes leading Bayesian network approaches.

2.4.1 Stochastic Weight Averaging of Gaussian Samples

Stochastic weight averaging of Gaussian samples (SWAG) by Maddox et al. (2019) describes an extension of prior work on stochastic weight averaging (SWA) (Izmailov et al., 2018). The weights of a neural network are averaged during different SGD iterations, which in essence can be interpreted as approximate Bayesian inference (Blei et al., 2017), tracing back to works of (Polyak and Juditsky, 1992; Ruppert, 1988). First we will explain SWA which can also be interpreted as averaged SGD (Polyak and Juditsky, 1992; Ruppert, 1988). The main difference between SWA and averaged SGD is that SWA utilises a simple moving average instead of an exponential one, in conjunction with a high constant learning rate, instead of a decaying one. In essence, in SWA one maintains a running average $\bar{\theta}_{\text{SWA}} = \frac{1}{n}\sum_{i=1}^{n}\theta_i$ over the weights of a NN during the last 25% of the training process utilised in updating the first and second moments of batch-normalisation. This leads to better generalisation since the SGD projections are smoothed out during the average process leading to wider optima in the optimisation landscape of the neural network. SWAG is an approximate Bayesian inference technique for estimating the covariance from the parameters of a neural network. SWAG maintains a running average of the second uncentered moment over the estimator parameters $\overline{\theta^2} = \frac{1}{n}\sum_{i=1}^{n}\theta_i^2$ in order to compute the covariance $\mathbf{\Sigma} = \text{diag}(\overline{\theta^2} - \bar{\theta}_{\text{SWA}})$ to derive the approximate Gaussian posterior $\mathcal{N}(\bar{\theta}_{\text{SWA}}, \mathbf{\Sigma})$. At test time the parameters of the neural network are drawn from this posterior $\theta_m \sim \mathcal{N}(\bar{\theta}_{\text{SWA}}, \mathbf{\Sigma})$ in order to per-

form Bayesian model averaging to retrieve the predictive posterior density along with the uncertainty estimates from the first and second moments.

2.4.2 Joint Energy Model

The *Joint Energy Model* (JEM) (Grathwohl et al., 2020) provides the ability to reinterpret a discriminative estimator $p(y|x)$ as an energy based model (EBM) of the joint distribution $p(y, x)$. An EBM (LeCun et al., 2006) parameterises a probability density via an unnormalised log-density function. An interesting aspect of this is that any probability density $p(x)$ can be expressed as an energy function.

$$p(x) = \frac{e^{-\overbrace{E_\theta(x)}^{\text{energy}}}}{\underbrace{p(\theta)}_{\text{evidence}}}. \tag{2.23}$$

where the energy function maps each data point $x \in \mathbb{R}^d$ to a scalar value and the evidence $p(\theta) = \int_x e^{-E_\theta(x)} dx$ acts as a normalising constant with respect to particular data x. Notice that as a statistical quantity the energy function is interpreted as a cross entropy estimator.

The key observation is to define the energy function of the joint distribution on data x and categories y via the logits of the discriminative estimator, that is $E_\theta(x) = -f_\theta(x)[y]$.

$$p(x, y) = \frac{e^{-f_\theta(x)[y]}}{p(\theta)}. \tag{2.24}$$

Marginalising out y on the numerator of Eq. 2.24 yields $p(x)$ and obtaining the predictive density $p(y|x)$ via $p(x, y)/p(x)$ amounts to parameterising a categorical distribution on logits via the softmax function, hence, indicating that there exist a generative estimator hidden in every standard discriminative estimator.

JEM can be trained on unlabelled data improving calibration, robustness, and out-of-distribution detection while also enabling to generate samples.

2.4.3 Dirichlet Prior Networks

Dirichlet prior networks (Malinin and Gales, 2018) (DPN) are an alternative approach to estimating the uncertainty of predictions. A subtle difference between this approach and those presented thus far is that DPN explicitly formulates a parameterisation of

a distribution using a neural network estimator while previous approaches aimed at constructing an implicit conditional distribution. An additional detail is that previous approaches only model the mismatch between train and test data through the parameters of the estimator by imposing a prior distribution on them, instead DPN considers it as a separate source of uncertainty and it formulates it explicitly as a distribution over predictive categorical densities $p(c|x, \theta)$ with point estimate categorical distribution c. This is illustrated in Eq. 2.25 where the middle term, $p(c|x, \theta)$ related to distribution mismatch between train and test data, is omitted in previous Bayesian neural networks.

$$p(y|x, S) = \iint \underbrace{p(y|c)}_{\text{data uncertainty}} \underbrace{p(c|x, \theta)}_{\text{distribution uncertainty}} \underbrace{p(\theta|S)}_{\text{model uncertainty}} \, dc \, d\theta. \tag{2.25}$$

Training the estimator involves optimising the neural network parameters while minimising the KL divergence between in-distribution and out-of-distribution data. The in-distribution data has been modelled with a prior sharp Dirichlet distribution while the out-of-distribution data has been modelled with a prior flat Dirichlet distribution. Once the estimator is trained we can obtain uncertainty measures for each prediction.

2.5 Out-of-Distribution Detection

The problem of out-of-distribution detection amounts to identifying a robust and stable estimator from a hypothesis class whose performance does not deteriorate with inputs from the unknown test distribution, implying that the underlying chosen estimator can recognise such inputs. An important aspect of this formulation is that the unknown test distribution is different from the train distribution upon which the estimator was previously exposed.

Because this violates a central assumption in supervised learning, in that regard, we present three formulations of the supervised OOD detection problem highlighting the essential difference among (i) an independent and identically distributed (i.i.d) setting where there is no distribution shift or ambiguous inputs, (ii) an out-of-distribution setting where there exists a distribution shift between train and test data, and (iii) an adversarial setting where there exists ambiguous inputs deliberately crafted from an adversary.

Let us begin by briefly recalling the definition of a data sample which will be prevalent in the following definitions of the different supervised OOD detection problems.

Definition 3 *(Data Sample): Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the input space and $\mathcal{Y} \subseteq \mathbb{R}$ the target space. Let $\mathcal{D}$ denote the n-fold product space $(\mathcal{X} \times \mathcal{Y})^n$, then a sample is a sequence $S = \{(x_i, y_i)\}_{i=1}^n \in \mathcal{D}$ of n independent and identically distributed (i.i.d.) observations according to an unknown joint distribution $P \in \mathcal{P}$, where $\mathcal{P}$ denotes the collection of all probability distributions on the space $\mathcal{X} \times \mathcal{Y}$.*

2.5.1 Learning Without Out-of-Distribution Data

Every learning problem often entails a random sample S, a pair of an input space and a target space $(\mathcal{X}, \mathcal{Y})$, and a quantity of interest $\mathbb{P}(Y|X)$. Every pair of values (x, y) in S denotes an i.i.d realisation of the random variables (X, Y) with X drawn from the input space $\mathcal{X}$ and Y from the target space $\mathcal{Y}$.

The main quantity of interest is the conditional distribution $\mathbb{P}(Y|X)$ relating covariates to targets, often expressed as a joint density over inputs and targets $\mathbb{P}(X, Y) = \mathbb{P}(Y|X)\mathbb{P}(X)$ (Shalev-Shwartz and Ben-David, 2014).

Denoting a random variable as a sum of its expected value and variance permits expressing the realisation of any pair (x, y) of X and Y as

$$y = \underbrace{\mathbb{E}[Y|X = x]}_{\text{mean}} + \underbrace{\delta}_{\text{variance or noise term}} . \tag{2.26}$$

and letting the estimator f approximate the mean value $\mathbb{E}[Y|X = x] = f(x)$ in Eq. 2.26, permits to describe the relation between inputs and targets as $y = f(x) + \delta$ without explicitly referring to the conditional distribution $\mathbb{P}(Y|X)$ (Shalev-Shwartz and Ben-David, 2014).

Thus, the main object of study is how well does f estimate the true mean value. In supervised learning this is defined by minimising the expected test error $\mathcal{E}$ of an estimator f from a hypothesis class $\mathcal{F}$ on a random test sample S derived from an underlying joint distribution P (Shalev-Shwartz and Ben-David, 2014). This is formally depicted in Definition 4.

Definition 4 *(Expected Error): Given an estimator $f \in \mathcal{F}$, an objective function L, and a distribution $P \in \mathcal{P}$, the expected test error $\mathcal{E}$ is defined as:*

$$\begin{aligned} \mathcal{E}_P(f) &= \mathop{\mathbb{E}}_{(x,y)\sim P}[L(f(x), y)] \\ &= \int_{\mathcal{X}\times\mathcal{Y}} L(f(x), y)\, dP(x, y). \end{aligned} \tag{2.27}$$

A related problem in estimating the expected test error $\mathcal{E}$ with respect to the joint distribution P is that we cannot directly compute it since we do not have access to this distribution. Instead, we often resolve in approximating the test error via the empirical train error $\widehat{\mathcal{E}}_S$ on a sample S of fixed length n.

Definition 5 *(Empirical Error): Given an estimator $f \in \mathcal{F}$, an objective function L, and a training set $S \sim P^n$, the empirical error is defined as:*

$$\widehat{\mathcal{E}}_S(f) = \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i). \tag{2.28}$$

There are two important questions which we need to answer. First, how can we obtain an estimator f that produces small train error? Second, how can we guarantee that the same estimator will also produce small test error? Answering the first question is the task of the learning algorithm which is responsible for identifying an f with small train error. Let $\mathcal{A}$ be a learning algorithm $\mathcal{A} : \mathcal{D} \rightarrow \mathcal{F}$ retrieving a suitable f based on a sample $S \in \mathcal{D}$, according to a predefined criterion $L : \mathcal{F} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ assigning an error for each inaccurate estimation of f.

To answer the second question we need to define a measure that can quantify how good of a proxy of the expected test error the empirical train error is. This is usually defined as the probability of the worst case deviation among the two error terms not exceeding a small scalar value ϵ (Shalev-Shwartz and Ben-David, 2014).

$$\mathbb{P}\left[\sup_{f \in \mathcal{F}} |\mathcal{E}_P(f) - \widehat{\mathcal{E}}_S(f)| > \epsilon\right]. \tag{2.29}$$

Thus, the goal of supervised learning is to asymptotically minimise the probability of the event in Eq. 2.29 in the limit of obtaining more data samples. Notice that this formulation does not introduce any statements in regards to distribution shift.

2.5.2 Learning With Out-of-Distribution Data

An important assumption in supervised learning is that both train and test data are derived from the same underlying joint distribution $P(X, Y)$ in the i.i.d setting. Suppose instead that the test data are derived from a different joint distribution $Q(X, Y)$ than the train data. How can supervised learning be formulated in this setting of distribution shift?

We present two key detection frameworks for the out-of-distribution setting. The first framework suggests learning robust estimators based on invariant associations between covariates rather than spurious correlations (Arjovsky et al., 2019). An important assumption of this framework is having access to data sampled from different distributions which all seem to share an invariant dependence of the target variable of interest Y on some of the covariates X among all distributions.

The common empirical error minimisation approach in supervised learning cannot distinguish between invariant associations and spurious correlations due to the restrictive setting of only observing i.i.d data from a generative process. The main idea is based on identifying a representation of covariates such that the final estimator would be optimal across all distributions.

Definition 6 (Arjovsky et al., 2019; Wald, 1945) is based on the assumption that the unknown test distribution P_{test} could belong to an infinite collection of distributions $\mathcal{P} = \{P_{test}\}_{n=1}^{\infty}$ interpreted as minimising the worst error over all test distributions $P_{test} \in \mathcal{P}$. Instead, the best one can hope for is to approximate $\mathcal{P}$ utilising a finite set of training distributions $\{P_{train}\}_{n=1}^{k} \subseteq \mathcal{P}$. This is the underlying assumption relating the train distributions to the unknown OOD test distributions through the existence of invariant covariates among them in order to minimise the expected invariant error.

Definition 6 *(Out-of-Distribution Detection as Invariant Error Minimisation): Given a collection of distributions $\mathcal{P}$, a hypothesis space $\mathcal{F}$ over randomly drawn unspecified data samples $S \in \mathcal{D}$ according to some unspecified joint distribution P_{test}, then the expected OOD test error for an estimator $f \in \mathcal{F}$ is defined as:*

$$\begin{aligned} \mathcal{E}_{OOD}(f) &= \sup_{P_{test} \in \mathcal{P}} \mathcal{E}_{P_{test}}(f) \\ &= \sup_{P_{test} \in \mathcal{P}} \mathop{\mathbb{E}}_{(x,y) \sim P_{test}} \left[L(f(x), y) \right]. \end{aligned} \tag{2.30}$$

where $\mathcal{E}_{P_{test}}(f)$ is the expected loss governed by distribution P_{test}.

The second framework formulates learning in the presence of distribution shift as an out-of-distribution detection problem in a binary classification setting, minimising misclassification between train P_X and test Q_X marginal distributions (see Definition 7) (Ben-David et al., 2006). Often we cannot minimise the expected out-of-distribution test error $\mathcal{E}_{OOD}$ directly due to a lack of prior knowledge on Q_X. On such occasions one assumes access to auxiliary information in the form of unlabelled data sampled from a distribution U_X different from P_X and Q_X.

Definition 7 *(Out-of-Distribution Detection as Binary Classification): Given input space $\mathcal{X} \subseteq \mathbb{R}^d$ and target space $\mathcal{Y} \subseteq \{0, 1\}$. Let Y denote a uniform draw from $\{0, 1\}$. Let P_X and Q_X be distinct marginal distributions on $\mathcal{X}$ for in- and out-distribution respectively. Let P be a joint distribution on $\mathcal{X} \times \mathcal{Y}$. At test time, sample a new instance from the joint distribution P with conditional distributions $\mathbb{P}(X|Y = 1) = P_X$, and, $\mathbb{P}(X|Y = 0) = Q_X$. Then given an estimator $f \in \mathcal{F}$ the expected OOD test error is:*

$$\begin{aligned} \mathcal{E}_{OOD}(f) &= \inf_{f \in \mathcal{F}} \mathcal{E}_P(f) \\ &= \inf_{f \in \mathcal{F}} \mathop{\mathbb{E}}_{(x,y)\sim P} [L(f(x), y)] + \underbrace{d(P_X, U_X)}_{\textit{metric function}}. \end{aligned} \tag{2.31}$$

Notice the difference between Definition 6 and Definition 7. In Definition 6 it is assumed that the estimator f has access to a diverse group of labelled in-distribution data during training in order to identify potential invariant covariates in these groups leading to robust estimators at test time detecting OOD inputs. Instead, in Definition 7 it is assumed that the estimator f might have access to unlabelled data during training different from the train data but not as diverse as those in Definition 6, assisting in identifying and detecting between in- and out-of-distribution inputs.

2.5.3 Learning With Adversaries

Often the data presented to an optimal estimator during inference can be maliciously corrupted leading to overconfident erroneous predictions. Such corruptions can occur either naturally due to measurement imperfection or due to an adversary maliciously seeking to exploit the estimator's errors. This is also a common example of distribution shift often investigated in an out-of-distribution detection setting.

In such occasions the solution usually resolves in a robust optimisation formulation also known as adversarial learning. The key idea is to identify an estimator that minimises the worst case empirical error on a finite set of plausible malicious perturbations.

Definition 8 (Cullina et al., 2018) states that it is possible to find a perturbation δ such that when added to the original input x then it will result in a misclassification while maintaining the visual semantic content of an image intact. Notice that the inner maximisation can be estimated by finding the worst case adversarial perturbation utilising either the fast gradient method (FGM) (Goodfellow et al., 2015) or the projected gradient method (PGM) (Kurakin et al., 2017).

Definition 8 *(Out-of-Distribution Detection as Adversarial Learning): Given an estimator $f \in \mathcal{F}$, a distribution $P \in \mathcal{P}$ and an input $x \in \mathcal{X} \subseteq \mathbb{R}^d$, there exists a ball $\mathcal{B}_\epsilon(x) = \{\delta : \|x - \delta\|_p \leq \epsilon\}$ of radius ϵ on a manifold $\mathcal{M}$ and a perturbation $\delta \in \Delta$, such that $\Delta = \{\delta : \delta \in \mathcal{B}_e(x) \cap [-1, 1]^n\}$ denotes the feasible region of admissible perturbations and $f_\theta(x + \delta) \neq f_\theta(x)$ while $\|x - (x + \delta)\|_p \leq \epsilon$ for all $\epsilon > 0$, where $\mathcal{B}_\epsilon(x)$ describes an ℓ_p–ball for ϵ small centred at x. Then, the expected OOD test error for an estimator f is:*

$$\begin{aligned} \mathcal{E}_{OOD}(f) &= \inf_{f \in \mathcal{F}} \sup_{\delta \in \Delta} \mathcal{E}_p(f) \\ &= \min_{f \in \mathcal{F}} \mathop{\mathbb{E}}_{(x,y) \sim P} \left[\underbrace{\max_{\delta \in \Delta} L(f(x + \delta), y)}_{\textit{adv. attack: } \{FGM, PGM\}} \right]. \end{aligned} \tag{2.32}$$

The idea is based on iterative gradient ascent to generate the adversarial samples and then project them back onto the feasible perturbation set Δ. The update rule for PGM is given in Eq. 2.33 where $\Gamma(\cdot)$ denotes the projection of inputs onto the acceptable range of pixel values Δ.

$$\begin{aligned} \underbrace{x_{t+1}}_{\text{adv. example}} &= \Gamma_\Delta(x_t + \eta \operatorname{sign}(\nabla L(f(x_t + \delta), y))) \\ \Gamma_\Delta(z) &= \epsilon \frac{z}{\max\{\epsilon, \|z\|_p\}} \end{aligned} \tag{2.33}$$

The concept is illustrated in Figure 2.1. Consider two categories, tori and crosses, resembling data points on a manifold $\mathcal{M}$ induced by a DNN. If we draw a ball $\mathcal{B}$ of radius ϵ centred at the cross data point closest to the decision boundary, then, there exists a perturbation δ and optimising in its direction would result into a misclassification by the DNN perceiving the target cross as a torus category.

Even though both Definition 6 and Definition 8 are expressed as minimising the worst case error over OOD data they should not be conflated. The main distinction between them is based on the assumption they impose on how the OOD data was generated. In Definition 6 this assumption was stated as a set of naturally occurring distribution shifts among different groups of data, whereas, in Definition 8 the set of distribution shifts is restricted to a finite number of worst case perturbations deliberately devised by an adversary on purpose to maximise the error on OOD data. Understanding better the set of worst case perturbations allows us to impose prior domain knowledge expressing which perturbations might occur in the future equipping an estimator f with the ability to recognise and detect OOD inputs.

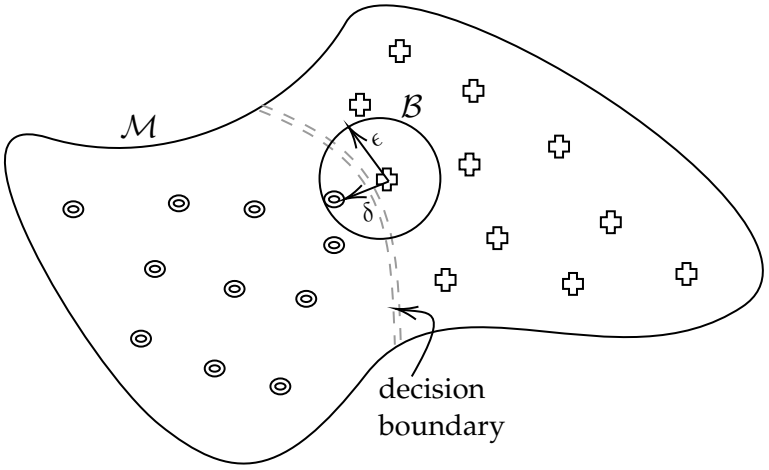


Figure 2.1: Adversarial example generation on data manifold $\mathcal{M}$.

2.5.4 Out of Distribution Detection Methods

Early methods for OOD detection relied on scores derived from multi-class estimators. Hendrycks and Gimpel (2017) proposed to utilise the confidence score (often referred to as maximum softmax probability (MSP)) to identify OOD inputs. Their approach was later refined by Liang et al. (2018) proposing ODIN which introduced rescaling the MSP scores via temperature scaling in addition to input perturbations. Lee et al. (2018b) demonstrated that one could obtain standard Gaussian density estimates from the estimator's intermediate activations conditioned on the target variable of interest to identify OOD inputs. Further improvement by Lee et al. (2018a) led to the Mahalanobis detector fitting a Gaussian distribution to the activation of the last layer of the estimator and performing OOD by measuring the Mahalanobis distance from the outputs to the in-distribution data.

Furthermore, a number of methods rely on auxiliary objectives based on the assumption of access to additional available data. For instance, MSRep (Shalev et al., 2018) introduced the idea of combining multiple semantic representations of several words obtained from different corpora or architectures as target labels to improve OOD detection, whereas Hendrycks et al. (2019a) in outlier exposure (OE) assumed access to additional unlabelled data to improve OOD detection on a joint objective. In addition, Hendrycks et al. (2019b) used self-supervised learning with auxiliary augmentations to retrieve robust estimators against OOD inputs. In the same spirit Mohseni et al. (2020) proposed self-supervised training coupled with a rejection mechanism for OOD inputs during inference.

Another popular approach to improve robustness and uncertainty estimation against OOD data relies on ensemble learning, aggregating estimates from either similar or different architectures of independently trained discriminative or generative estimators (Choi and Jang, 2018; Lakshminarayanan et al., 2017; Vyas et al., 2018). Unfortunately ensembling multiple estimators can be computationally intensive which led to development of methods striving to improve quality of latent variables in hopes of obtaining more informative representations to distinguish OOD inputs from in-distribution data.

Sastry and Oore (2020) demonstrated that the obtained Gram matrices from multiple latent variables can be utilised to improve OOD detection while Masana et al. (2018) proposed a multi-head architecture to amplify the quality of latent variables for OOD detection.

Ren et al. (2019) discovered that the likelihood score of deep generative estimators for OOD detection is highly affected by the background statistics of the input data, therefore, they proposed a likelihood ratio method which effectively corrects for these confounding background statistics and enhances the in-distribution specific covariates for OOD detection. Morningstar et al. (2021) introduced DoSE to ameliorate the issue often found in generative estimators assigning higher likelihood to OOD than in-distribution data. To identify OOD samples they constructed an estimator on several summary statistics of the in-distribution data, and then during inference, they mark the data points that have low support under the observed densities of the measurements as out-of-distribution.

Another interesting approach to OOD detection is contrastive learning. In that regard Tack et al. (2020) proposed contrasting shifted instances (CSI), a training scheme that in addition to contrasting a given sample with other instances it also contrasts the sample with distributionally shifted augmentations of itself. In addition, to detect OOD inputs they proposed a new detection score specific to their training scheme. Winkens et al. (2020) also proposed a contrastive learning approach and a new detection score which quantifies the difficulty of the OOD detection by capturing the similarity of inlier and outlier datasets. Their scoring mechanism is based on the probability with which an estimator confuses outliers with inliers. Instead, Liu et al. (2020) proposed an energy score for OOD detection (EnergyOOD) demonstrating its efficacy over traditional softmax scores while being less susceptible to the overconfidence issue.

Finally, Lee et al. (2020) suggested a meta-learning estimator that adaptively balances the effect of meta-learning and task-specific learning within each task exhibiting robustness against OOD inputs. By learning the different balancing variables between tasks they can decide to obtain a solution by relying on either the meta-knowledge

or the task-specific learning. Their objective is formulated into a Bayesian inference framework and solved using variational inference.

2.6 Evaluation of Out-of-Distribution Detection

Evaluating OOD detection algorithms is not a trivial task, as such, we present a general experiment methodology outlining the datasets, performance metrics, methods and evaluation process followed throughout this dissertation {unless specifically stated otherwise}.

2.6.1 Datasets

The datasets used in this dissertation can be categorised into the following three categories: synthetic datasets, real datasets and real corrupted datasets. The synthetic datasets are often constructed for the purpose of evaluating a hypothesis and obtaining insights regarding the underlying question under investigation. Instead, the real and real corrupted datasets refer to well-established image classification datasets in the machine learning literature to empirically validate and confirm a promising hypothesis already evaluated on synthetic datasets.

Synthetic Datasets

The synthetic dataset consists of samples drawn from Gaussian distributions with different mean and standard deviation parameters assembled into unique clusters resembling categories of a multi-class classification setting. This dataset is presented in Chapter 6 consisting of 50,000 train samples and 10,000 test samples comprised of three distinct categories for training and four categories for testing in order to simulate OOD inputs for detection.

Real Datasets

The real datasets used in this dissertation mostly comprised of colour images representing distinct categories. For instance, *CIFAR-10* consists of ten categories depicting animals and vehicles (including planes), while, *CIFAR-100* consists of one hundred categories depicting people, animals, vehicles, plants, and foods. *SVHN* is a result of the Google street view project representing colour images of house numbers of ten

categories, one for each digit. *FashionMNIST* represents greyscale images of fashion accessories and clothing, from ten categories. Finally, *LSUN* consists of of ten categories of scenes depicting different places and twenty categories of objects including people, animals, plants, household items, and vehicles (including planes). Further details regarding the number of instances and categories of each dataset are described below.

- The *CIFAR-10* (Krizhevsky, 2009) dataset consists of 60,000 colour images of dimensionality 32x32 with 10 categories. Each category contains 6,000 images. In total there are 50,000 train images and 10,000 test images.
- The *CIFAR-100* (Krizhevsky, 2009) dataset consists of 60,000 colour images of dimensionality 32x32 with 100 categories. Each category contains 600 images. In total there are 50,000 train images and 10,000 test images.
- The *SVHN* (Netzer et al., 2011) dataset consists of 99,289 colour images of dimensionality 32x32 representing digits of house numbers. There exist 10 categories one for each digit. In total there are 73,257 train images, and 26,032 test images.
- The *FahsionMNIST* (Xiao et al., 2017) dataset consists of 70,000 greyscale images of dimensionality 28x28 with 10 categories. Each category contains 6,000 images. In total there are 60,000 train images and 10,000 test images.
- The *LSUN* (Yu et al., 2015) classification dataset consists of colour images of dimensionality 512x512 with 10 categories. Each category contains numerous images ranging from approximately 120,000 to 3,000,000. The validation data includes 300 images, and the test data has 1000 images per category.

Real Corrupted Datasets

Three datasets that add corruptions to the images in the CIFAR-10 and CIFAR0-100 datasets are also used.

The *CIFAR-10-C* dataset consists of 10,000 colour test images of dimensionality 32x32 with 10 categories corrupted with 15 common corruptions (see Figure 2.2) at 5 different severity levels, with 1 being the lowest and 5 the highest. The total number of corrupted test images is 15 x 5 x 10,000 = 750,000.

The *CIFAR-100-C* dataset is identical to *CIFAR-10-C* also corrupted in a similar manner except that it has 100 categories instead of 10.

Finally, the *CIFAR-10-Adv.* dataset consists of 10,000 colour test images of dimensionality 32x32 with 10 categories corrupted with adversarial noise via the projected gradi-

ent descent mechanism of generating adversarial samples. This dataset is presented in Chapter 5 whereas *CIFAR-10-C* and *CIFAR-100-C* are presented in Chapter 6.

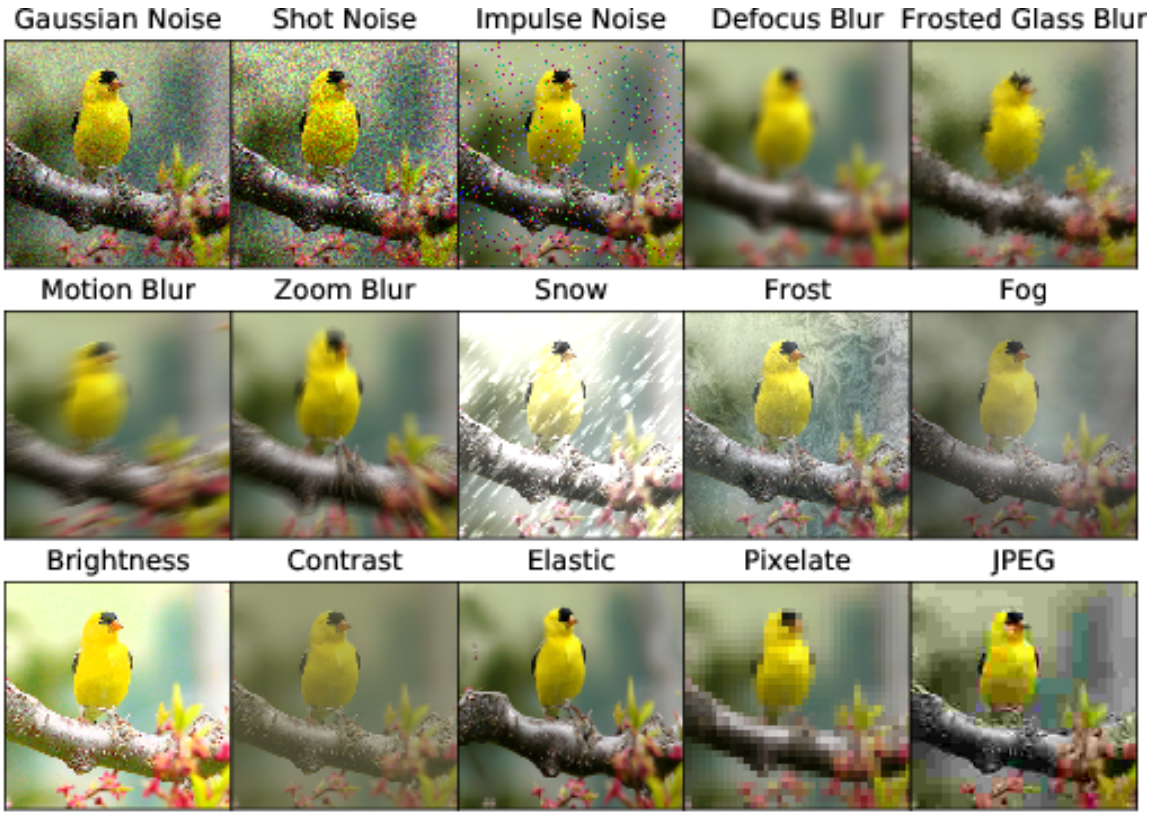


Figure 2.2: Common corruptions for *CIFAR-10-C* and *CIFAR-100-C* (reproduced from Hendrycks and Dietterich (2019)).

2.6.2 Performance Metrics

Existing methods for OOD detection often convert their outputs into a score indicating the likelihood that an input instance is out-of-distribution. Usually common metrics utilised to calculate these scores include *confidence*, *entropy*, *mutual information*, and *differential entropy*. These metrics can also be applied to the outputs of a deep neural network to perform OOD detection.

In-Distribution Classification Metrics

To evaluate the efficacy of the estimator f on the in-distribution multi-class classification task we utilised the accuracy on the predictions along with its confidence, often referred to as maximum softmax probability (MSP) or normalised exponential function.

$$\text{acc}(f) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{f(x_i) = y_i\}. \tag{2.34}$$

$$\text{conf}(f) = \frac{1}{n}\sum_{i=1}^{n} \max_{y} f(x_i)[y]. \tag{2.35}$$

Calibration Metrics

To validate whether an estimator is calibrated we utilised the expected calibration error (ECE) metric defined in Eq. 2.36, on the validation set in combination with reliability diagrams (example shown in Figure 3.3).

$$\text{ECE} = \sum_{b=1}^{B} \frac{|b|}{n} |\,\text{acc}(b) - \text{conf}(b)|. \tag{2.36}$$

Uncertainty Metrics

To capture the uncertainty of an estimator on its outputs we utilised entropy over its predictions defined in Eq. 2.8 in combination with the symmetric KL divergence along with Figure 3.4.

$$KL(P\|Q) + KL(Q\|P) = \int p(x) \ln \frac{p(x)}{q(x)} dx + \int q(x) \ln \frac{q(x)}{p(x)} dx. \tag{2.37}$$

Typically the continuous analogue of the discrete relative entropy is referred to as differential entropy.

$$KL_D(P\|Q) = -\int p(x) \ln \frac{p(x)}{q(x)} dx. \tag{2.38}$$

In addition, mutual information can also be employed to capture uncertainty describing the relative entropy between the joint distribution and the product of its marginals.

$$\begin{aligned} I(X;Y) &= KL(P_{(X,Y)}\|P_X \times P_Y) \\ &= \iint p(x,y) \ln \frac{p(x,y)}{p(x)q(x)} dxdy. \end{aligned} \tag{2.39}$$

Out-of-Distribution Detection Metrics

A common metric utilised in out-of-distribution detection to identify whether inputs are derived from the in-distribution data rather than the out-of-distribution data is the area under the curve of the receiver operating characteristic function.

$$\text{AUC-ROC}(f) = \frac{\sum\limits_{x_{id} \in S_{ID}} \sum\limits_{x_{ood} \in S_{OOD}} \mathbb{1}\left[f(x_{id}) < f(x_{ood})\right]}{|S_{ID}| \cdot |S_{OOD}|}. \tag{2.40}$$

The indicator function returns 1 if the inequality is true otherwise 0. The in-distribution data is denoted by S_{ID} while the out-of-distribution data with S_{OOD}. Of course, one could also use confidence, entropy, mutual information or even differential entropy to construct AUC-ROC scores for ood detection.

Robustness Against Corruptions Metrics

In order to measure the underlying robustness of an estimator against common corruption we utilise the mean corruption error (mCE) similar to (Hendrycks and Dietterich, 2019) except for we do not normalise the obtained scores with the errors of a reference estimator like for instance AlexNet.

$$\text{mCE}(f) = \frac{1}{C} \sum_{c=1}^{C} \sum_{v=1}^{V} f(x_c^v). \tag{2.41}$$

The notation x_c^v indicates an input corrupted with corruption c at severity level v.

2.6.3 Models and Methods

To assess the performance of a proposed method it is important to carefully select a set of estimator architectures for comparison and evaluation. Among established point estimate deep neural network architectures one might find VGG16 (Simonyan and Zisserman, 2015), PreResNet164 (He et al., 2016), and WideResnet28x10 (Zagoruyko and Komodakis, 2016). To maintain a fair evaluation across point estimate deep neural networks and Bayesian neural networks (see Chapter 2.4) we utilised the same underlying architectures.

For instance, in Chapter 3 the estimator architectures for both point estimate deep neural networks and Bayesian neural networks are VGG16, PreResNet164 and WideResNet28x10. Instead, in Chapters 4, 5, and 6 the underlying estimator architecture is WideResNet28x10.

2.6.4 Experiment Design

For every experiment performed on real datasets there is a corresponding in-distribution (ID) and out-of-disribution (OOD) dataset. The in-distribution dataset is predominantly used for the multi-target classification setting and is split into train, validation and test sets. The validation set is used to tune hyperparameters of the estimators whereas the test set is only exposed during inference. In most occasions during the training phase of the underlying estimator we do not assume access to additional labelled or unlabelled data and whenever that is not the case we state it explicitly (e.g. Chapter 4, Chapter 6). Once the training process is over we obtain the estimator with the best accuracy on the validation set ready to be further evaluated during inference on the final test set.

2.6.5 Summary

The research programme presented in this dissertation aims at better understanding Bayesian neural networks in terms of calibration, uncertainty estimation, out-of-distribution detection and adversarial robustness. We demonstrate that Bayesian neural networks by default are better calibrated than point estimate deep neural networks. Moreover, we show that Bayesian neural networks perform better at uncertainty estimation and out-of-distribution detection than point estimate deep neural networks. In addition, we show that by default neither Bayesian neural networks nor point estimate deep neural networks can withstand against adversarial inputs causing the underlying estimators to confidently misclassify them, and additional mitigation techniques such as adversarial defences might be necessary to amplify robustness and out-of-distribution detection. Finally, we present two objectives improving out-of-distribution detection utilising auxiliary information from unlabelled data sources assisting the underlying estimators in appropriately decorrelating between in-distribution and out-of-distribution covariates.

CHAPTER

THREE

UNCERTAINTY ESTIMATION IN BAYESIAN NEURAL NETWORKS

This chapter evaluates the ability of Bayesian neural networks in terms of calibration and uncertainty estimation in an effort to answer research questions **RQ1** and **RQ2**.

3.1 Introduction

Deep neural networks (DNN) have been adopted at an increasing pace by industry as predictive estimators, but their sensitivity and lack of robustness on ambiguous inputs poses real challenges for critical domain applications. As such, the problem of quantifying uncertainty regarding the estimator's predictions in the presence of ambiguous inputs has attracted less attention and techniques formulated on the basis of a Bayesian framework such as Bayesian neural networks (BNN) which could potentially capture data uncertainty are no exception.

We propose to close this gap by evaluating the validity of BNN in terms of uncertainty estimation over their predictions against point estimate deep neural networks (DNN). The key idea is based on leveraging Bayesian inference in combination with neural networks in order to quantify the uncertainty over predictions along with approximate measures such as the posterior predictive uncertainty.

The **key questions** investigated in this chapter are presented below which stem directly from research questions **RQ1** and **RQ2** introduced in Section 1.1 of Chapter 1:

1. Are Bayesian neural networks better calibrated than point estimate neural networks?

2. Are Bayesian neural networks better at uncertainty estimation in prediction tasks than point estimate neural networks?

3. Does better calibration in Bayesian neural networks imply overall better uncertainty estimation?

We address these questions using four datasets and comparing three DNN against six BNN on three different metrics. Additionally, we investigate any potential correlation between calibration and accuracy, as well as, calibration and uncertainty estimation between in and out of distribution inputs. First, we rank all estimators based on their accuracy on the test set of every dataset. Next, we turn to calibration error as a means of identifying potentially miscalibrated estimators. Finally, we examine the ability of all estimators to identify and express uncertainty over ambiguous inputs between in-distribution and out-of-distribution data.

The **key contributions** of this chapter are listed below, and they contribute towards **CB1** and to a certain extent **CB2** introduced in Section 1.2 of Chapter 1:

- An empirical validation showing that Bayesian neural networks are better calibrated than point estimate neural networks.
- An empirical study demonstrating the validity of uncertainty estimation in Bayesian deep learning methods.
- A study examining the correlation between calibration and uncertainty estimation in Bayesian neural networks.

3.2 Experiment Design

This study utilises four well-established datasets in the machine learning literature: *CIFAR-10*, *SVHN*, *FashionMNIST* and *CIFAR-100* (Section 2.6.1). The empirical evaluation presented consists of the following three components: (i) estimators, (ii) confidence calibration and (iii) uncertainty estimation. Three point estimate deep neural networks were used: VGG16, PreResNet164 and WideResNet28x10 (Section 2.6.3). Two Bayesian neural networks were used: MC-Dropout (Section 2.3.1) and SWAG (Section 2.4.1), utilising the same underlying architecture similar to the point estimate deep neural networks (i.e. VGG16, PreResNet164, WideResNet28x10).

To evaluate the generalisation ability of the estimators on a hold-out test set we utilised accuracy, Eq. 2.34. Each estimator was trained on half of the randomly chosen

categories from the four datasets with the remaining half categories withheld for the purpose of evaluating their ability to associate out-of-distribution instances with high uncertainty, given that the withheld categories were only introduced at inference. The duration of training was 300 epochs with the final estimator selected being the best performing on the validation set.

The selected optimiser during the experiments was stochastic gradient descent (SGD) (Robbins and Monro, 1951) with initial learning rate set to $\eta = 0.05$, with momentum set to $m = 0.9$, and weight decay γ in the range $[3e^{-4}, 5e^{-4}]$. Additionally, data augmentation utilising random horizontal flip, random cropping and random pixel distortion were applied on the training set.

To measure the calibration of an underlying estimator we utilised the expected calibration error (ECE), Eq. 2.4, in combination with the equivalent reliability plots. This metric captures the disagreement between the estimator's predictions and the true empirical proportion of instances for each category often expressed as a weighted average between accuracy and confidence. In order to validate that the results are not a product of random chance and that the estimators are indeed calibrated provided that their expected calibration errors are adjacent, we utilised bootstrap hypothesis testing for calibration on the test set of each dataset under the null hypothesis that the estimator's outputs are calibrated (Vaicenavicius et al., 2019). The underlying approach to hypothesis testing and its algorithmic procedure are described below.

Given an estimator f we analyse if its estimates $f(x)$ on a test set S are calibrated according to an arbitrary statistic $h(f(S))$, such as for instance, ECE, logloss, Brier score, etc. Since we are mostly concerned with expected calibration error then the statistic h refers to the ECE score. The overall procedure can be described in the following steps:

1. Set the null hypothesis of the estimator to be calibrated $H_0 : h(f(S), y_b) = h(f(S), y)$.
   Set the alternative hypothesis to be not calibrated $H_1 : h(f(S), y_b) > h(f(S), y)$.

2. Generate B bootstrapped category sets y_b for $b \in \{1, \ldots, B\}$, such that $y_{b,i}$ is sampled from $f(x_i)$.

3. Calculate the ECE score $h(f(S), y_b)$ for each bootstrapped category set y_b evaluated on the actual test set S.

4. Reject the null hypothesis if the probability of the ECE score $\mathbb{P}(h(f(S), y_b)) > h(f(S), y))$ on the bootstrapped category set y_b exceeds that of the true category set y.

Algorithm 1 Bootstrap Hypothesis Test for Calibration

procedure BOOTSTRAPTEST(f, h, S, y)
  for trial $\leftarrow 1$ to $10,000$ **do**
    $b \leftarrow \{1, \ldots, |S|\}$ ▷ resample with replacement from the set of indices
    $y_b \leftarrow \arg\max_y f_y(S)[b]$ ▷ estimate labels for random sample on test set
    $s \leftarrow h(f(S), y_b)$ ▷ compute ECE for the estimated set of labels
    $p \leftarrow p + \sum_{i=1}^{|S|} \mathbb{1}\{s_i > h(f(S), y)\}$ ▷ estimate probability of an extreme ECE
  end for
  if $(p/10,000) < 0.05$ **then**
    return Reject H_0 ▷ Reject null hypothesis if probability is small
  end if
end procedure

To characterise out-of-distribution inputs with high uncertainty we estimated the predictive uncertainty, Eq. 2.8, over the estimator's predictions. Recall that every dataset is split randomly into two parts. One of the parts entailing half of the categories representing the in-distribution data on which the estimator was trained, with the other part denoting as out-of-distribution instances the remaining categories which were utilised only during inference.

Finally, to conveniently compare and summarise the performance of the different estimator's regarding their ability to characterise and identify out-of-distribution inputs with high uncertainty the scalar value of the symmetric KL divergence, Eq. 2.37, between the predictive uncertainty, Eq. 2.8, over the predictions of the in-distribution data and the predictive uncertainty over the predictions of the out-of-distribution data, was selected as a sensible candidate of a summary statistic to interpret Figures 3.5, 3.6 and 3.7. The symmetric KL divergence permits to evaluate how similar are two distributions with larger values denoting a noticeable distinction between them. In essence, large values of the symmetric KL divergence denote the ability of the underlying estimator to capture and associate out-of-distribution samples with high uncertainty.

3.3 Results

In this section we describe the results from the experiments along with the findings arising in regard to the questions outlined in Section 3.1. The accuracy of all the estimators across datasets is depicted in Table 3.1 and equivalently in Figure 3.1 where it is shown that Bayesian neural networks overall exhibit higher accuracy. For instance, on *Fashion MNIST* and *SVHN*, WideResNet28x10-SWAG has the highest accuracy while on *CIFAR-10*, WideResNet28x10-MC Dropout represents the highest accuracy.

Table 3.1: Accuracy of estimators on datasets *CIFAR-10/100*, *SVHN* & *FashionMNIST*.

Estimators	CIFAR-10	SVHN	FashionMNIST	CIFAR-100
VGG16	94.40	97.10	95.76	74.92
VGG16-MC Dropout	93.26	96.87	95.60	72.58
VGG16-SWAG	93.80	96.83	96.30	74.44
PreResNet164	93.56	97.90	96.86	80.62
PreResNet164-MC Dropout	94.68	97.73	97.12	80.58
PreResNet164-SWAG	93.14	97.69	**97.18**	82.24
WideResNet28x10	94.04	97.44	97.16	81.44
WideResNet28x10-MC Dropout	**95.54**	97.63	97.08	81.98
WideResNet28x10-SWAG	95.12	**97.95**	**97.18**	**83.40**

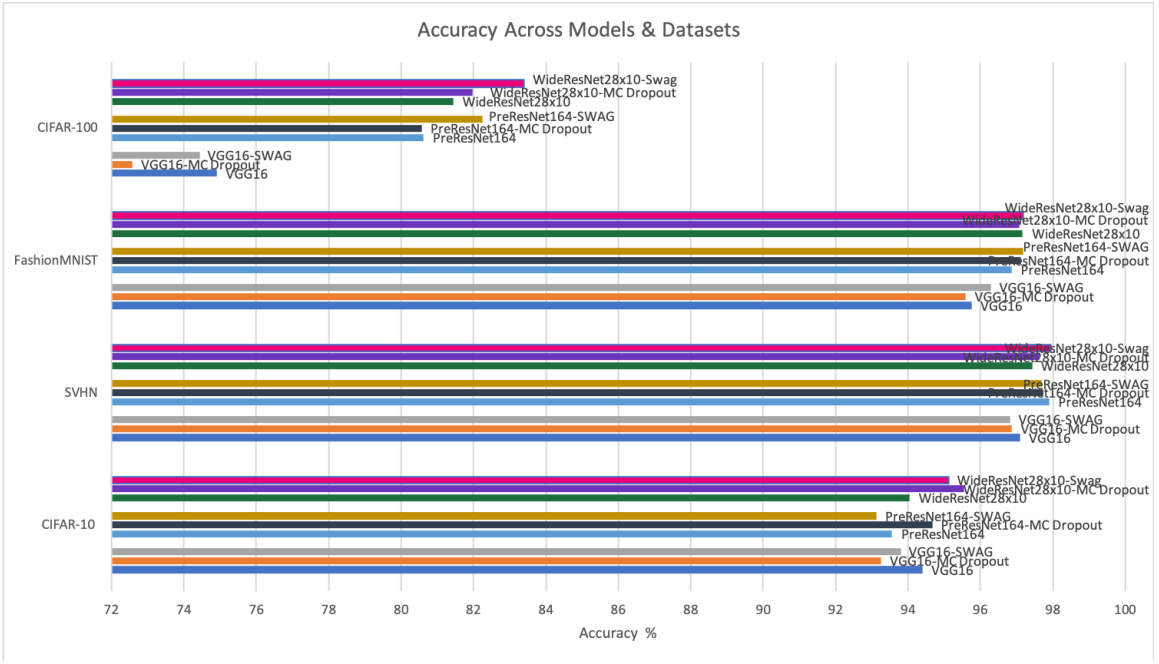


Figure 3.1: Accuracy across estimators and datasets.

In order to answer the first question of whether Bayesian neural networks exhibit better calibration compared to point estimate deep neural networks we draw the attention of the reader to Figures 3.2, and 3.3 in conjunction with Table 3.2. The expected calibration errors (ECE) in Table 3.2 measuring the degree of model miscalibration seem to be in accordance with the results from Guo et al. (2017) demonstrating that modern neural networks are indeed poorly calibrated. In addition, the reliability plots presented in Figure 3.3 indicate a perfectly calibrated estimator by the diagonal line. Anything below the diagonal represents an over-confident estimator, while anything above the diagonal represents an under-confident estimator. Observe in Figures 3.2 and 3.3 that the

majority of the estimators are to an extent miscalibrated. Some Bayesian approaches, however—in particular estimators based on MC-Dropout and SWAG—are better calibrated than their point estimate deep neural networks counterparts.

Table 3.2: Expected calibration errors (ECE) in % for *CIFAR-10*, *SVHN*, *FashionMNIST*, and *CIFAR-100*. Values $< 1\%$ indicate calibrated estimators (Guo et al., 2017).

Estimators	CIFAR-10	SVHN	FashionMNIST	CIFAR-100
VGG16	6.77	3.08	3.29	16.18
VGG16-MC Dropout	4.23	1.55	1.55	11.81
VGG16-SWAG	4.99	2.05	1.73	7.53
PreResNet164	3.10	2.38	2.16	9.67
PreResNet164-MC Dropout	3.38	1.61	2.01	9.97
PreResNet164-SWAG	4.93	**0.90**	1.54	1.70
WideResNet28x10	2.00	2.10	2.12	5.11
WideResNet28x10-MC Dropout	2.55	1.47	2.08	3.96
WideResNet28x10-SWAG	**0.98**	**0.82**	1.15	**0.96**

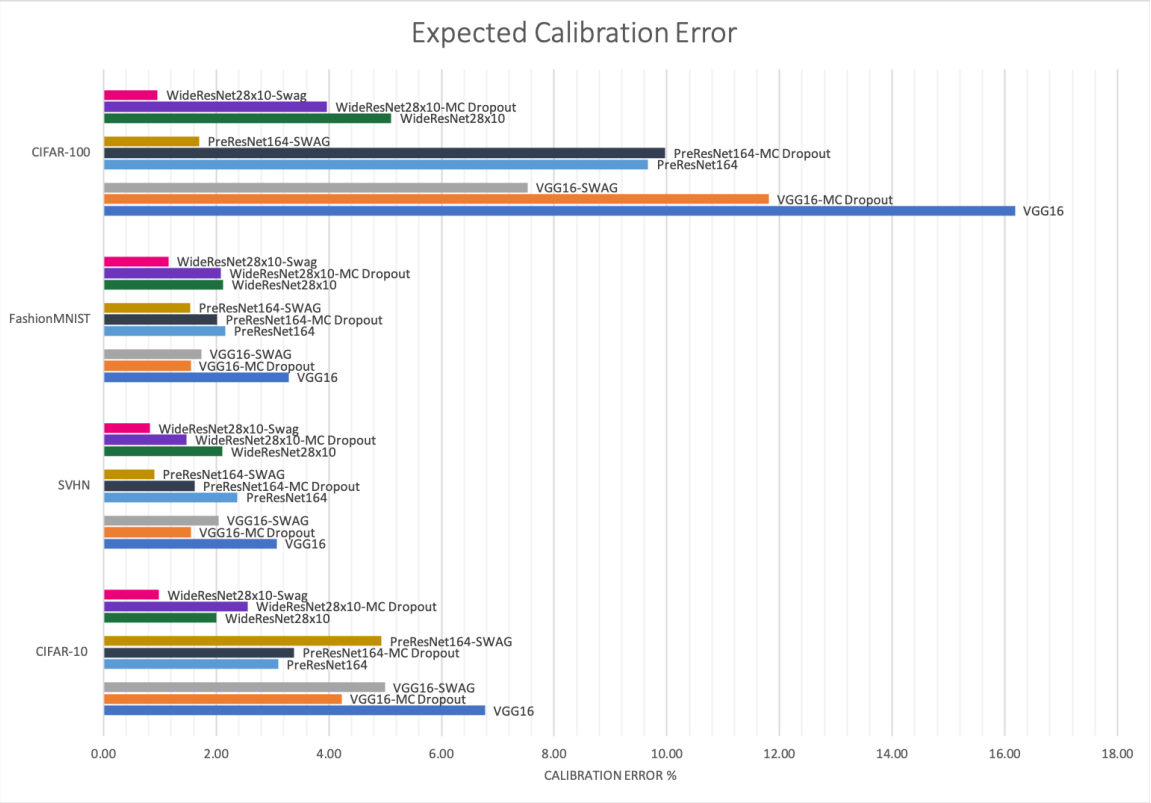


Figure 3.2: Expected calibration error across datasets and estimators.

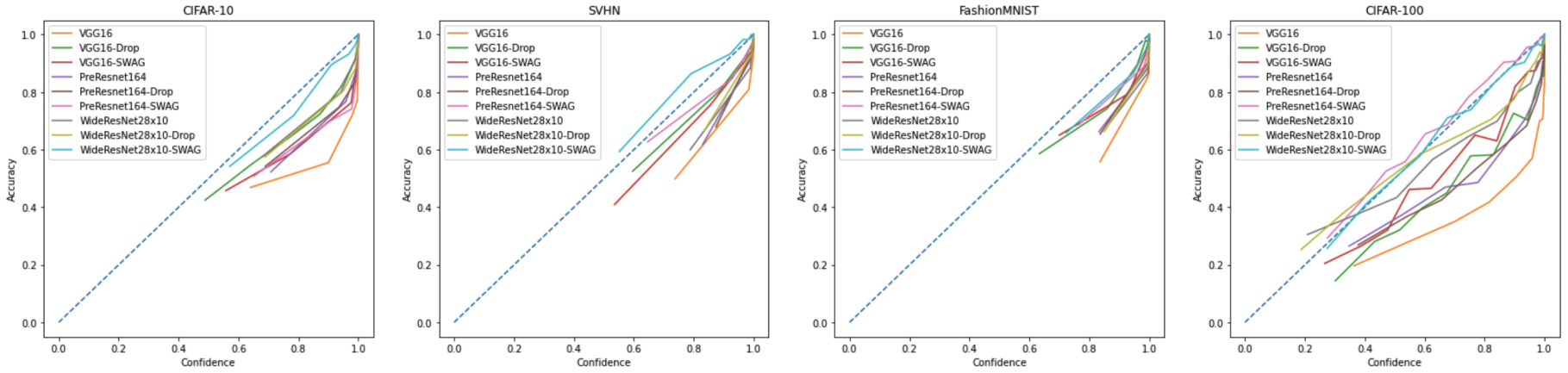


Figure 3.3: Reliability plots across all estimators on *CIFAR-10*, *SVHN*, *FashionMNIST* and *CIFAR-100* datasets.

To further evaluate that the ECE scores are not solely a product of randomness in the test set we perform a bootstrap hypothesis test in order to indicate which estimators are truly calibrated and which are not. In Table 3.3 we see that two BNNs, VGG16-MC Dropout and WideResNet28x10-SWAG indicate to be calibrated across all datasets.

Table 3.3: Bootstrap hypothesis testing of esitmators across each dataset, representing p-values. Bold values indicate miscalibrated estimators.

Estimators	CIFAR-10	SVHN	FashionMNIST	CIFAR-100
VGG16	**0.001**	**0.001**	**0.001**	**0.001**
VGG16-MC Dropout	0.987	0.624	0.986	0.581
VGG16-SWAG	**0.001**	**0.001**	0.16	0.99
PreResNet164	**0.001**	**0.001**	**0.001**	**0.001**
PreResNet164-MC Dropout	**0.001**	**0.001**	**0.001**	**0.001**
PreResNet164-SWAG	**0.001**	0.746	**0.001**	0.99
WideResNet28x10	0.086	**0.001**	**0.001**	0.99
WideResNet28x10-MC Dropout	**0.001**	**0.001**	**0.001**	0.99
WideResNet28x10-SWAG	0.99	0.98	0.765	0.99

Next, to answer the second question with regard to whether Bayesian neural networks are better at uncertainty estimation than point estimate deep neural networks we draw the attention of the reader to Table 3.4 and equivalently Figures 3.4, 3.5, 3.6, 3.7 and 3.8. The information presented in Table 3.4 provides a summary of the uncertainty estimation over OOD inputs from Figures 3.5, 3.6, 3.7 and 3.8, by measuring the symmetric KL divergence, on the predictive uncertainty between in- and -out-of-distribution instances.

Interpreting Table 3.4 we can see that Bayesian neural networks like MC Dropout and SWAG overall perform better at uncertainty estimation on three of the four datasets compared to point estimate deep neural networks such as PreResNet164 which performs better only on one dataset with MC Dropout second best. The same trend is evident also in Figures 3.5, 3.6, 3.7 and 3.8, utilising predictive uncertainty, Eq. 2.8, as a measure of expressing uncertainty over OOD instances, such that an estimator properly equipped with uncertainty estimation over its predictions should assign low entropy on in-distribution data and high entropy for out-of-distribution data. Notice that on *FahsionMNIST* in Figure 3.6, *SVHN* in Figure 3.7 and *CIFAR-100* in Figure 3.8 again two Bayesian neural networks are performing best, that is MC Dropout and SWAG, while on *CIFAR-10* in Figure 3.5 the best performing model is a point estimate deep neural network, that is PreResNet164.

Table 3.4: Symmetric KL divergence between in- and out-of-distribution splits of *CIFAR-10* (5 + 5), *SVHN* (5 + 5), *FahsionMNIST* (5 + 5) and *CIFAR-100* (50 + 50). Larger values indicate the ability of an estimator to identify OOD instances with high uncertainty.

Estimators	CIFAR-10	SVHN	FashionMNIST	CIFAR-100
VGG16	2.9527	5.6382	1.5199	0.9352
VGG16-MC Dropout	3.8494	**6.2732**	1.4053	1.4609
VGG16-SWAG	2.3750	5.0582	**2.0606**	1.6580
PreResNet164	**4.2443**	3.3753	1.6280	1.3280
PreResNet164-MC Dropout	2.8793	2.7053	1.1958	1.1980
PreResNet164-SWAG	1.8101	3.3446	1.4967	**1.7798**
WideResNet28x10	2.1810	3.0513	1.1995	1.4813
WideResNet28x10-MC Dropout	2.9291	2.9955	1.2085	1.5285
WideResNet28x10-SWAG	2.7802	3.6469	1.6674	1.4530

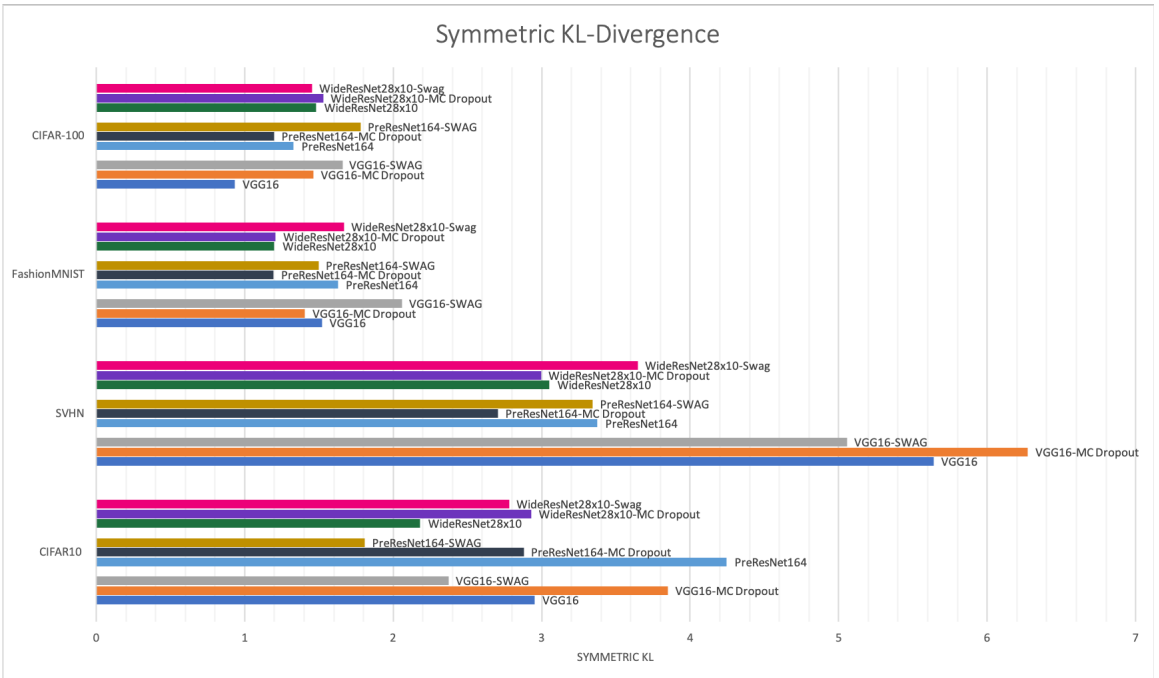


Figure 3.4: Symmetric KL diverg. on predictive uncertainty for in- & OOD predictions.

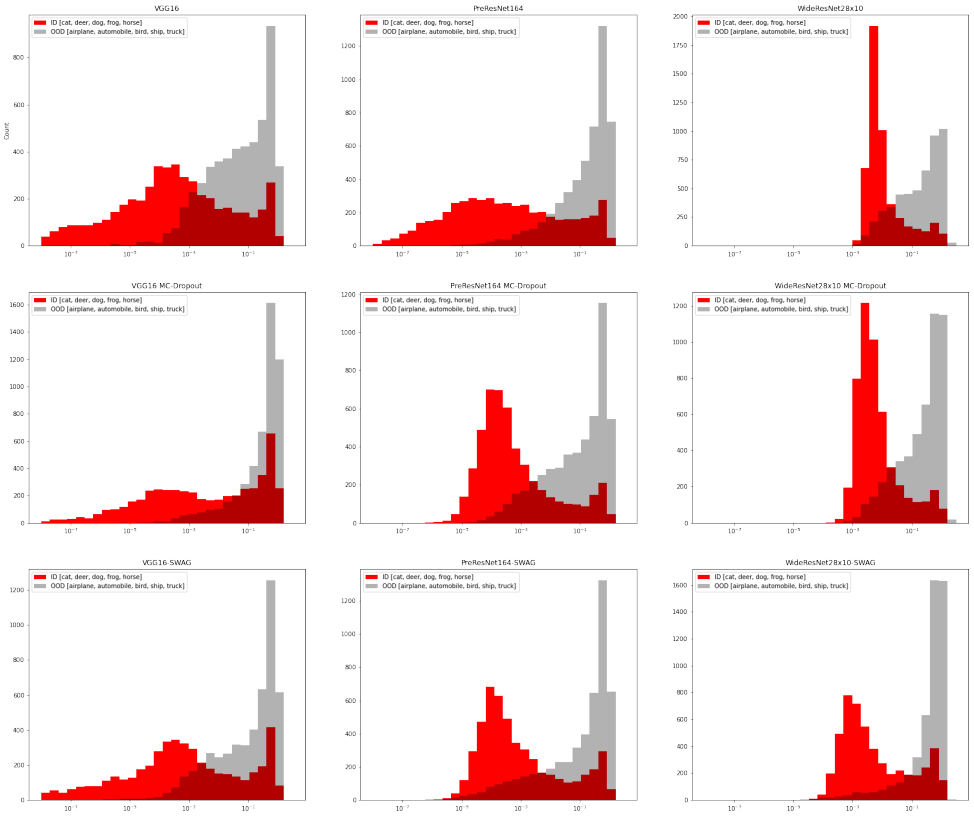


Figure 3.5: Out of sample distributional entropy plots for all estimators on CIFAR-10 (5 + 5) categories. The x-axis denotes entropy in logarithmic scale.

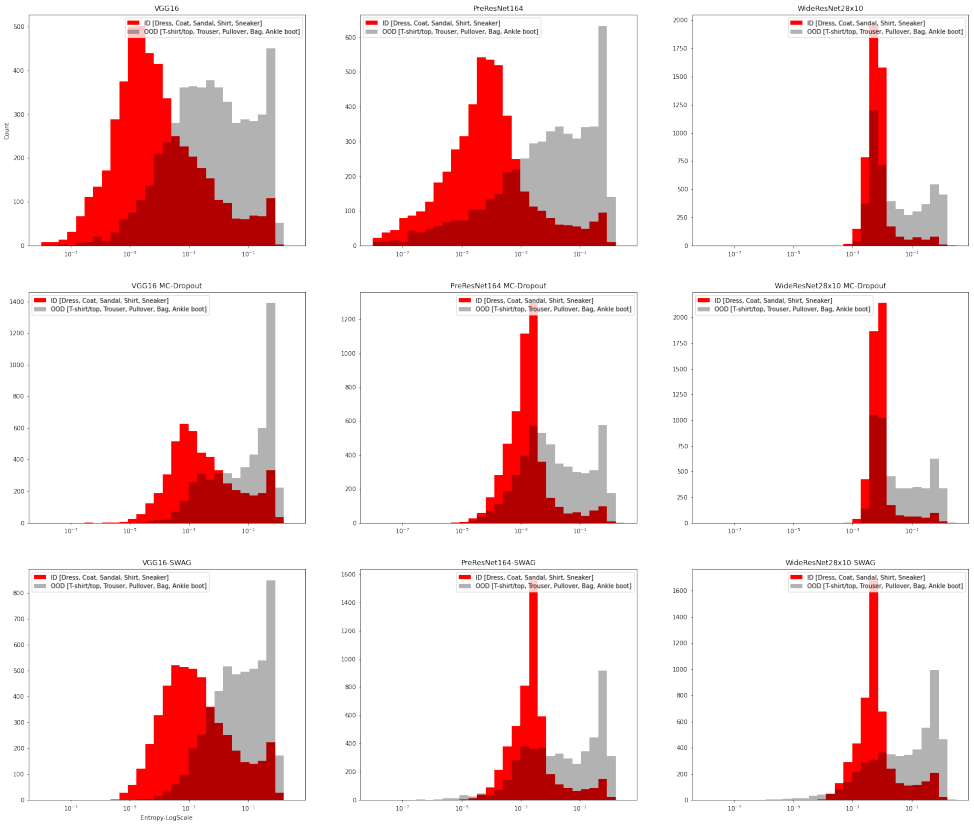


Figure 3.6: Out of sample distributional entropy plots for all estimators on FashionMNIST (5 + 5) categories.

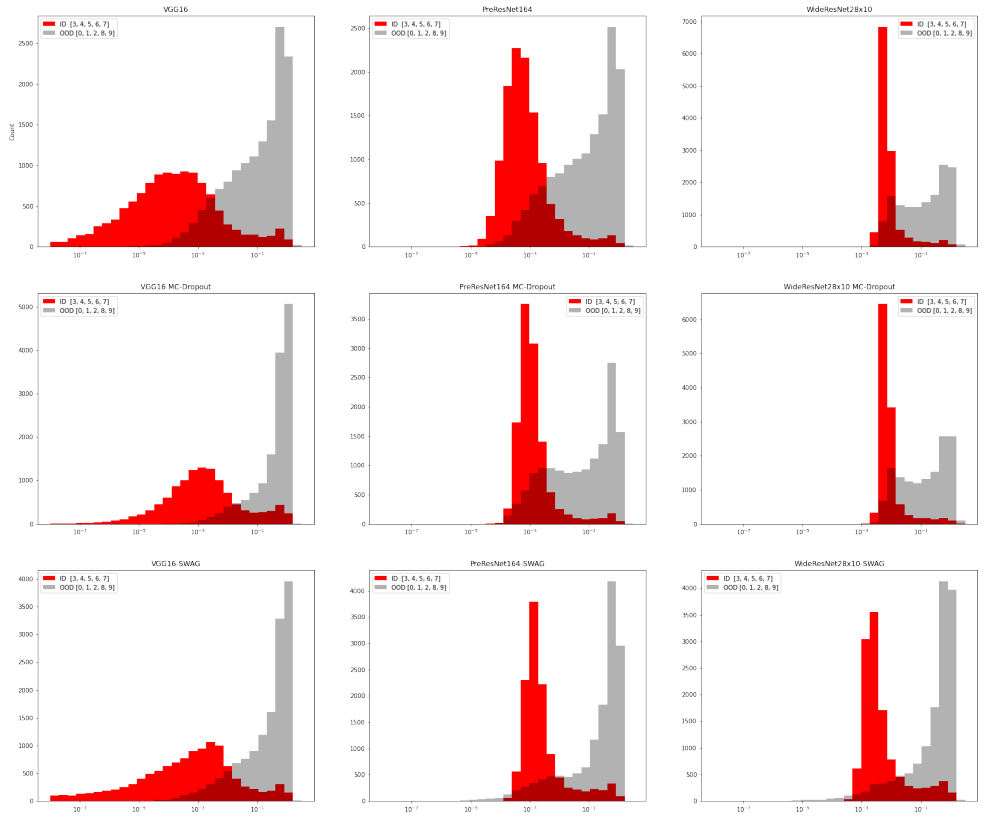


Figure 3.7: Out of sample distributional entropy plots for all estimators on SVHN (5 + 5) categories.

Finally, to answer the last question of whether better calibration in Bayesian neural networks would also imply better uncertainty estimation we examined, first, the existence of any correlation between calibration and accuracy. This would help us identify if better calibration also implies better overall accuracy, and, second, the existence of any correlation between calibration and uncertainty estimation. We present the findings in the scatter plots of Figure 3.9 examining the correlation between calibration and accuracy, and Figure 3.10 demonstrating any potential correlation between calibration and uncertainty estimation.

In Figure 3.9 one should observe a slight negative trend between calibration and accuracy indicating that the more accurate an estimator is the better calibrated it is overall. This observation can be somewhat misleading. An estimator can be quite accurate but miscalibrated, for instance this is the case of PreResNet164-SWAG on *FashionMNIST* dataset which exhibits the same accuracy as WideResNet28x10-SWAG on *Fash-*

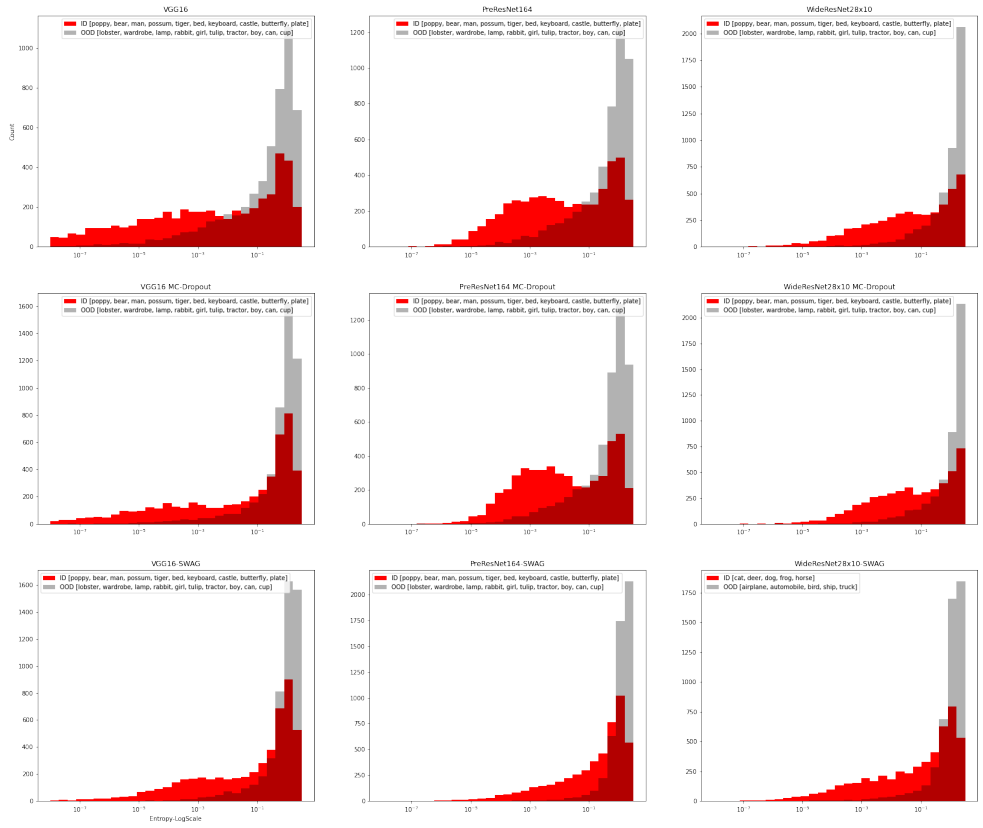


Figure 3.8: Out of sample distributional entropy plots for all estimators on CIFAR-100 (50 + 50) categories.

ionMNIST dataset but in comparison with it can be considered as miscalibrated, or equivalently an estimator can be very well calibrated but inaccurate, this is the case of WideResNet28x10-SWAG on *CIFAR-10* which compared to WideResNet28x10-SWAG on *CIFAR-100* exhibits almost the same degree of calibration but cannot be considered as accurate (see Table 3.3). Overall, it seems that any potential underlying correlation between calibration and accuracy of an estimator might be misleading. In contrast, in the scatter plot of Figure 3.10 there seems to be an absence of a general strong trend since the majority of the data points resemble almost a random pattern. Observe for instance that VGG16 on *SVHN* is the second best at expressing uncertainty over OOD inputs and still is miscalibrated when compared to WideResNet28x10-SWAG on *SVHN* dataset, indicating that an estimator can be well-equipped with uncertainty estimation but poorly calibrated. Similarly, an estimator can be calibrated, for instance this is the

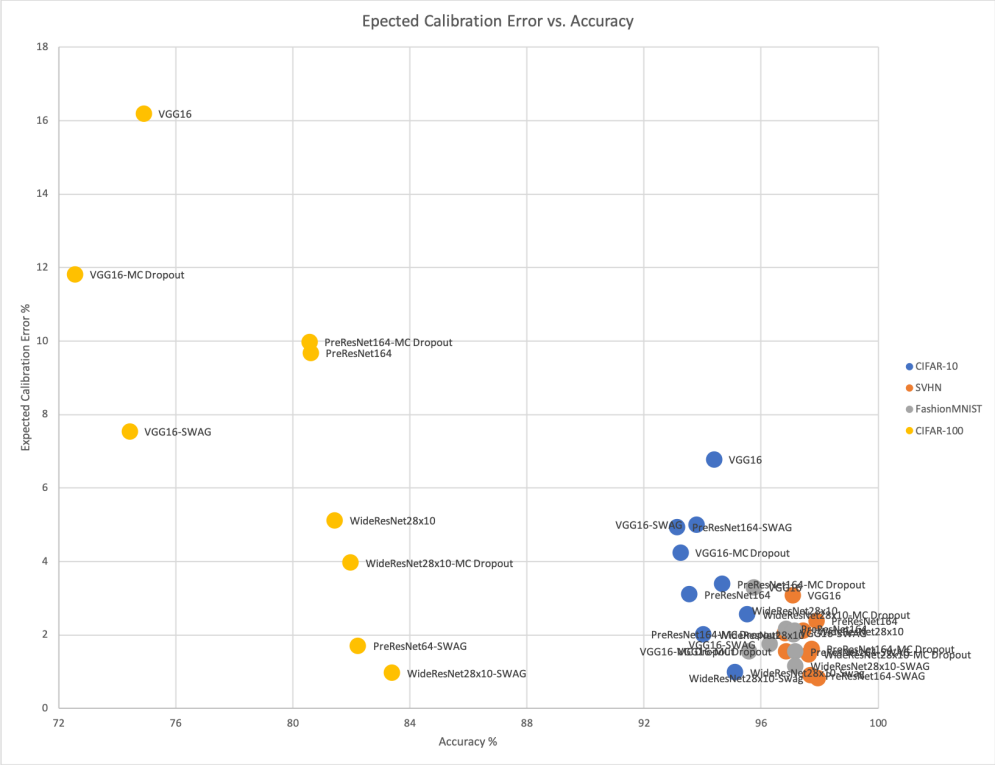


Figure 3.9: Scatter plot of calibration vs. accuracy across estimators and datasets.

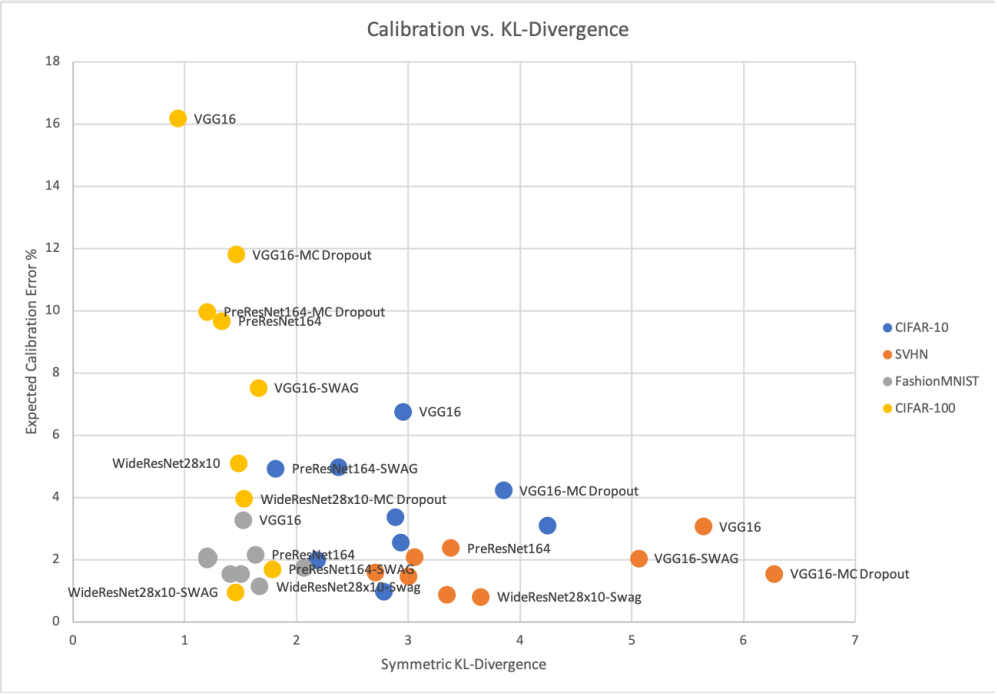


Figure 3.10: Scatter plot of calibration vs. symmetric KL across estimator and datasets.

case of WideResNet28x10-SWAG on *SVHN* indicating the best overall calibration, but poor at representing uncertainty estimation over OOD inputs.

In order to formally validate any trend or lack thereof in Figure 3.10, we conducted a Pearson correlation coefficient between expected calibration error and symmetric KL-divergence followed by a two tailed paired t-test hypothesis to indicate any statistical significance between the means of the expected calibration error μ_0 and symmetric KL-divergence μ_1 scores. The null hypothesis is set to be $H_0 : \mu_0 = \mu_1$ with the alternative hypothesis being $H_1 : \mu_0 \neq \mu_1$. The t-statistic is estimated based on the following formula $t = \frac{\rho\sqrt{n-1}}{\sqrt{1-\rho^2}}$ whereas the degrees of freedom are estimated according to $df = n - 1$.

The results in Table 3.5 indicate a negligible negative correlation between expected calibration error and symmetric KL-divergence on *CIFAR-10* and *FashionMNIST* potentially indicating any absence of influence between calibration and uncertainty estimation. Instead, we observe a stronger negative correlation on *CIFAR-100* indicating that an improvement in uncertainty estimation could result in better calibration by reducing its error. In contrast, on *SVHN* one observes a mild positive correlation indicating that an improvement in uncertainty estimation could result in an overall degradation regarding the calibration error of an estimator. This conclusion is also supported by the paired t-test hypothesis where the p-values < 0.05 on *SVHN* and *CIFAR-100* indicate the existence of statistically significant difference in the sample means between expected calibration error and symmetric KL-divergence scores.

Table 3.5: Pearson correlation coefficient and t-test hypothesis test for significant difference between expected calibration error and symmetric KL-Divergence.

	CIFAR-10	SVHN	FashionMNIST	CIFAR-100
ρ-Pearson	-0.0600	0.3566	-0.1962	-0.7639
n-samples	9	9	9	9
t-statistic	-0.1590	1.0097	-0.5295	-3.1315
df-degrees-of-freedom	8	8	8	8
p-value	0.2730	0.0006	0.0812	0.0084

Combining the results from Table 3.4 along with Figures 3.4, 3.5, 3.6, 3.7 and 3.8 seems to suggest that Bayesian neural networks have an advantage compared to point estimate deep neural networks at the task of uncertainty estimation on OOD instances. Although, the result is not undisputed and a larger study would be more appropriate, given that there were occasions where the point estimate deep neural networks would obtain better uncertainty estimation (i.e. higher divergence scores) than the Bayesian neural networks, though overall the results are indicative in the direction of Bayesian neural networks having a benefit for uncertainty estimation.

3.4 Conclusion

In conclusion, we have shown that point estimate deep neural networks indeed suffer from poor calibration and inability to identify out-of-distribution instances with high uncertainty. Bayesian deep neural networks provide a principled and viable alternative that allows the estimators to be informed about the uncertainty in their parameters and at the same time exhibits a lower degree of sensitivity against OOD inputs compared to their point estimate DNN. Even though Bayesian neural networks outperform point estimate deep neural networks in the task of uncertainty estimation over OOD inputs on the majority of datasets, we would argue that a larger study would be beneficial in evidently retrieving more conclusive results determining whether this capability of Bayesian neural networks to estimate uncertainty can also be transferred efficiently in the domain of out-of-distribution detection where the interest lies within detecting such ambiguous inputs and not simply expressing an uncertainty over them. This suggests that there is a promising research direction for improving the performance of Bayesian deep neural networks in detecting out-of-distribution instances.

CHAPTER

FOUR

OUT-OF-DISTRIBUTION DETECTION IN BAYESIAN NEURAL NETWORKS

This chapter evaluates the ability of Bayesian neural networks to detect out-of-distribution instances in an effort to answer research question **RQ3**.

4.1 Introduction

In this work we extend the benefits of uncertainty estimation in Bayesian neural networks (BNN) to the domain of out-of-distribution (OOD) detection. Two main challenges are of concern when considering the ability of an estimator to detect OOD inputs. First, the overall incapability of common estimators including modern neural networks to detect OOD samples previously not exposed to them. Second, that misclassified predictions which unintentionally are assigned high confidence. These challenges potentially can lead to problems related with trustworthiness, transparency and privacy of the underlying estimates (Schulam and Saria, 2019).

The majority of the proposed solutions have their foundations in different fields including differential privacy, information theory, robust high dimensional statistics, robust control theory and robust optimisation. Recently attention has been concentrated to methods that provide a principled approach to quantifying uncertainty through Bayesian neural networks (Gal and Ghahramani, 2016; Grathwohl et al., 2020; Maddox et al., 2019; Malinin and Gales, 2018). In spite of their sophistication, questions regarding the effectiveness of Bayesian neural networks in OOD detection still remain unanswered.

In this work, we propose to close this gap, by evaluating the efficacy of recent Bayesian neural networks in detecting OOD inputs without explicitly being trained on such inputs, in a benchmark study involving four datasets *CIFAR-10*, *SVHN*, *FashionMNIST*,

and *CIFAR-100* (Section 2.6.1) and five estimators, DNN, DPN (see Section 2.4.3), MC-Dropout (see Section 2.3.1), SWAG (see Section 2.4.1), and JEM (see Section 2.4.2), with a DNN acting as the baseline model. The key idea is to identify whether Bayesian neural networks can detect OOD inputs better than point estimate DNN without explicitly being trained on them. This formulation of OOD detection stems from Definition 7 in Chapter 2.

The **key questions** investigated in this work are presented below stemming from research question **RQ3** introduced in Section 1.1 of Chapter 1:

1. Are Bayesian neural networks effective at detecting out-of-distribution samples without previous exposure to them?
2. Is this ability of the estimators to detect OOD inputs correlated with their accuracy?

The **key contributions** of this work are the following, and they contribute towards **CB2** introduced in Section 1.2 of Chapter 1:

- A benchmark study demonstrating that Bayesian deep learning methods do indeed outperform point estimate neural networks in OOD detection.
- An empirical validation showing that from the considered approaches in this work, *Stochastic Weight Averaging of Gaussian Samples* (SWAG) (Maddox et al., 2019) is the most effective (achieving the best balance between performance in the OOD detection task and the original image classification task).
- We demonstrate that OOD detection is harder when there exists even minimal overlap between in-distribution and out-of-distribution examples and outlier exposure (Hendrycks et al., 2019a) might not be beneficial in such scenarios.

4.2 Experiment Design

We chose arbitrarily a dataset A to train the estimators. After obtaining the best estimator from the training phase on dataset A, to evaluate its performance on the OOD detection task we perform inference twice, once on the test set of dataset A and a second time on the test sets of each of the remaining datasets $B, C, D, \ldots$, different from A. Finally, to establish whether an estimator f overall is superior to an estimator g in terms of OOD detection we record and analyse the appropriate performance metrics presented in Section 2.6.2 in order to construct an informed decision.

Particularly, we compare the performance of four state-of-art likelihood methods based on Bayesian neural networks on the task of OOD detection—*DPN*, *MC-Dropout*, *SWAG*, and *JEM*—with the performance of a standard point estimate deep neural network (DNN) using a number of well-known image classification datasets—*CIFAR-10*, *CIFAR-100*, *SVHN*, and *FashionMNIST*. We utilise a 28 layers wide and 10 layers deep WideResNet28x10 (Zagoruyko and Komodakis, 2016) as the DNN estimator, and this deep neural network architecture also serves as the base architecture employed by the remaining estimators. When available, pre-trained estimators were employed (i.e. JEM (Grathwohl et al., 2020)), while for the remaining estimators (i.e. DNN, DPN, MC-Dropout, SWAG) we trained each for 300 epochs using a validation set for hyperparameter tuning and rolling back to the best estimator to avoid overfitting.

The selected optimiser during the experiments was stochastic gradient descent (SGD) (Robbins and Monro, 1951) with initial learning rate set to $\eta = 0.05$, with momentum set to $m = 0.9$, and weight decay [1] γ in the range $[3e^{-4}, 5e^{-4}]$. Additionally, data augmentation utilising random rotation with an angle $\theta \leq 25^o$ degrees, random horizontal flip, random cropping and random pixel distortion were applied on the training set.

To measure the performance of the different estimators we utilised the following metrics: accuracy, Eq. 2.34, and AUC-ROC scores, Eq. 2.40, based on *entropy*, Eq. 2.8, *mutual information*, Eq. 2.39 and *differential entropy*, Eq. 2.38. Accuracy provides an indication of the estimator's classification performance on the in-distribution (ID) inputs while AUC-ROC scores are predominantly utilised in evaluating the estimator's performance in detecting OOD inputs. First, we evaluate each estimator's accuracy on the classification problem for which it was trained using the relevant test dataset. Next, to evaluate the performance of the estimator in OOD detection we measure its ability to distinguish between in-distribution (ID) examples from the relevant test set and OOD examples from the test set from one of the other remaining datasets. For instance, when *SVHN* is used as the in-distribution training set, *CIFAR-10*, *LSUN*, and *CIFAR-100* are used as the out-of-distribution test sets. This means that for each training set we have three different evaluations of OOD detection effectiveness. One of the available out-of-distribution datasets is selected for use at training time for DPN which requires this extra data. Only the training portion of this dataset is used for this purpose, while the test portion is used for evaluation.

As already noted the predictions provided by the estimators are converted into OOD scores using the following metrics:

[1] Weight decay is a regularisation method, often "justifiably" conflated with ℓ_2-norm regularisation on the model parameters θ since the overall difference between the two methods is subtle, but in the case of adaptive optimisers this can be misleading (Loshchilov and Hutter, 2019)

- *Predictive uncertainty* (Shannon, 1948), Eq. 2.8, computing information entropy over the estimated predictions.
- *Mutual information* (Shannon and Weaver, 1949), measuring the amount of information obtained about a random variable X by observing some other random variable Y.
- *Differential entropy* (Lazo and Rathie, 1978), Eq. 2.38 also known as the continuous entropy used to measure distributional uncertainty between in-distribution and out-of-distribution data.

To avoid having to set detection thresholds we use these metrics to obtain scores for in-distribution (ID) and out-of-distribution (OOD) instances of the equivalent test sets in order to generate an AUC-ROC curve for the two categories, ID and OOD. We do this individually for each estimator and metric to generate the equivalent AUC-ROC scores.

4.3 Results

Table 4.1 and Figure 4.1 presents the overall performance of each estimator for the in-distribution image classification task, evaluated using classification accuracy. The results indicate that, for most cases, Bayesian methods perform comparatively with regard to the DNN baseline, except for DPN exhibiting lower accuracy on all datasets, SWAG exhibiting lower accuracy only on *FashionMNIST* and JEM exhibiting lower accuracy on *CIFAR-10*, *SVHN* and *FashionMNIST*.

In Tables 4.2, 4.3 and 4.4, and accordingly, Figures 4.2, 4.3, 4.4, and 4.5 we present the results of the OOD detection experiments. The scores represent AUC-ROC scores calculated using *predictive uncertainty* (i.e. *entropy*), *mutual information*, and *differential entropy* on the estimator's predictions. The values inside parenthesis indicate the percentage improvement with respect to a DNN which is treated as a baseline. The up arrow $\uparrow$ indicates an improvement and the down arrow $\downarrow$ indicates a reduction in performance. The last row of the table indicates the average percentage improvement across the dataset combinations for each approach with respect to a DNN. Finally, Table 4.5 summarises the results from Tables 4.1, 4.2, 4.3 and 4.4, representing the average performance increase based on AUC-ROC scores utilising each OOD scoring metric.

We can deduce from Tables 4.2, 4.3, and 4.4 that Bayesian methods— DPN, MC-Dropout and SWAG—almost always improve OOD detection performance over the DNN baseline. Interestingly, in our experiment JEM consistently performed poorly

Table 4.1: Accuracy of estimators for the in-distribution dataset classification task.

Estimators	CIFAR-10	SVHN	FashionMNIST	CIFAR100
DNN	95.06	96.67	95.27	77.44
DPN	88.10	90.10	93.20	79.34
MC-Dropout	96.22	96.90	95.40	78.39
SWAG	96.53	97.06	93.80	78.61
JEM	92.83	96.13	83.21	77.86

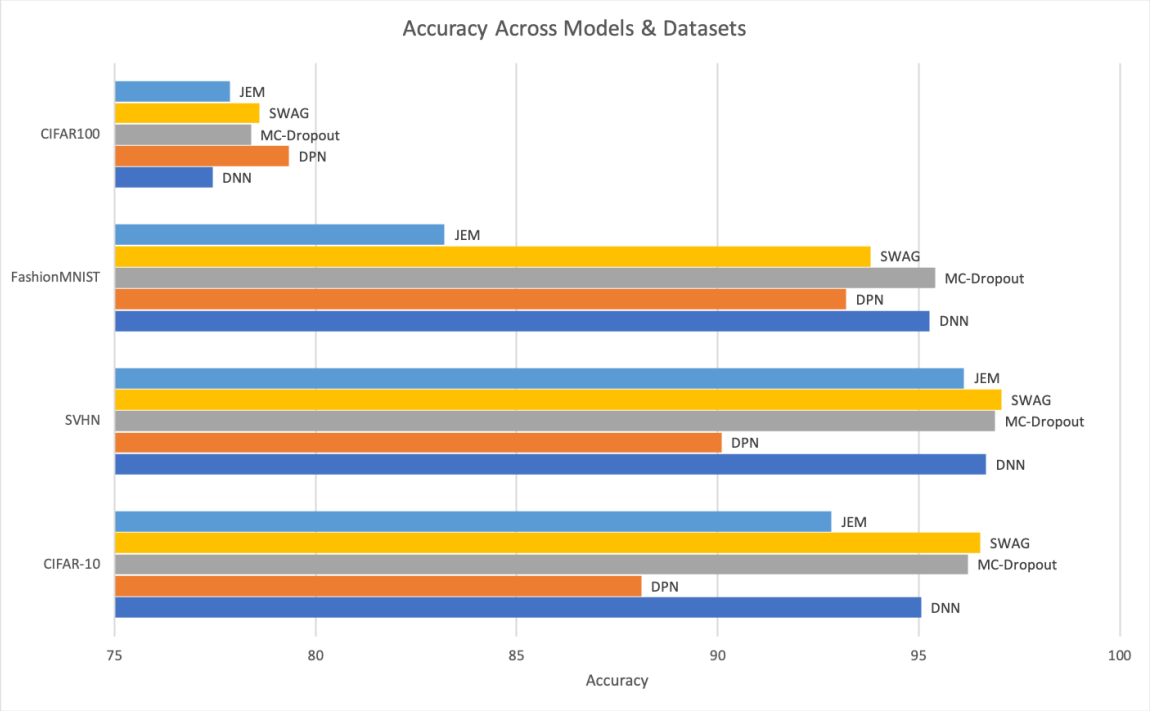


Figure 4.1: Accuracy across estimators and datasets.

with respect to the DNN. We attribute this mostly to a failure of robust selection of hyperparameter combinations. Despite our best efforts we were not able to achieve classification accuracy above 83% on the *FashionMNIST* dataset using this approach, and the OOD performance suffered from this.

However, even for the datasets in which the JEM estimator performed well for the image classification task (i.e. *SVHN* and *CIFAR-100*) its use did not lead to an increase in OOD detection. In addition, following the hyperparameter combinations in (Malinin and Gales, 2018) we were not able to successfully replicate results for DPN regarding the in-distribution case with *CIFAR-10* and accordingly the OOD case with *CIFAR-100*. In both cases the estimators diverged. Overall, we found that both JEM and DPN can be extremely sensitive to the choice of hyperparameters.

The results in Table 4.5 indicate that across the different evaluation metrics, DPN performs best for OOD detection, with SWAG following closely behind. MC-Dropout is

Table 4.2: Out-of-distribution experiment results. Scores are ***Entropy*** based AUC-ROC scores. The values in parenthesis are % improvement of the corresponding estimator w.r.t. DNN, taken as a baseline. An ↑ indicates improvement and ↓ degradation. The asterisks (*) indicate the out-distribution datasets used to train DPN.

Data		(baseline)	Entropy AUC-ROC score (% gain w.r.t. baseline)			
In-distribution	OoD	DNN	DPN	MC-Dropout	SWAG	JEM
CIFAR-10	CIFAR-100*	86.27	85.60 (↓0.78%)	89.92 (↑4.23%)	91.89 (↑6.51%)	87.35 (↑1.25%)
	SVHN	89.72	98.90 (↑10.23%)	96.25 (↑7.28%)	98.62 (↑9.92%)	89.22 (↓0.56%)
	LSUN	88.83	83.30 (↓6.23%)	92.04 (↑3.61%)	95.12 (↑7.08%)	89.84 (↑1.14%)
SVHN	CIFAR-100	93.19	99.10 (↑6.34%)	94.33 (↑1.22%)	95.97 (↑2.98%)	92.34 (↓0.91%)
	CIFAR-10*	94.58	99.60 (↑5.31%)	94.97 (↑0.41%)	96.03 (↑1.53%)	92.85 (↓1.83%)
	LSUN	92.97	99.70 (↑7.24%)	93.31 (↑0.37%)	95.71 (↑2.95%)	91.82 (↓1.24%)
FashionMNIST	CIFAR-100	91.20	99.50 (↑9.10%)	93.75 (↑2.80%)	96.19 (↑5.47%)	62.79 (↓31.15%)
	CIFAR-10*	94.59	99.60 (↑5.30%)	96.06 (↑1.55%)	94.28 (↓0.33%)	64.76 (↓31.54%)
	LSUN	93.34	99.80 (↑6.92%)	97.40 (↑4.35%)	99.05 (↑6.12%)	65.38 (↓29.96%)
CIFAR-100	CIFAR-10	78.25	85.15 (↑8.82%)	80.70 (↑3.13%)	84.92 (↑8.52%)	77.64 (↓0.78%)
	SVHN*	81.52	92.64 (↑13.64%)	85.59 (↑4.99%)	94.16 (↑15.51%)	81.22 (↓0.37%)
	LSUN	77.22	86.38 (↑11.86%)	76.58 (↓0.83%)	87.22 (↑12.95%)	77.54 (↑0.41%)
Avg % improvement			(↑6.48%)	(↑2.76%)	(↑6.60%)	(↓7.96%)

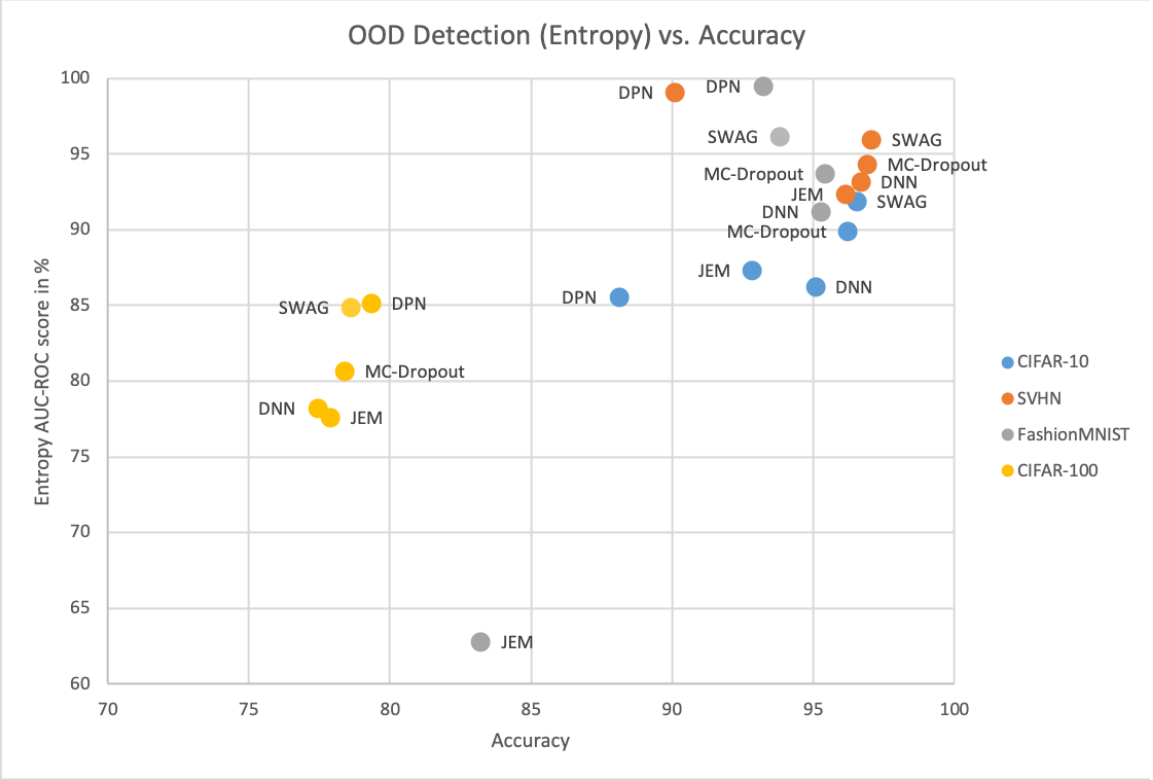


Figure 4.2: Scatter plot of OOD detection vs. accuracy based on entropy AUC-ROC scores.

third, but comparatively worse than the other two. All three Bayesian methods on average increase OOD detection performance with respect to the DNN baseline, regardless of the metric utilised to calculate OOD scores. The choice of a Bayesian neural

Table 4.3: Out-of-distribution experiment results. AUC-ROC scores are based on ***Mutual Information***. The values in parenthesis are % improvement of the corresponding estimator w.r.t. DNN, taken as a baseline. An ↑ indicates improvement and ↓ degradation. The asterisks (*) indicate the out-distribution datasets used to train DPN.

Data		(baseline)	Mutual Information AUC-ROC score (% gain w.r.t. baseline)			
In-distribution	OoD	DNN	DPN	MC-Dropout	SWAG	JEM
CIFAR-10	CIFAR-100*	86.37	83.89 (↓2.87%)	90.04 (↑4.25%)	92.06 (↑6.59%)	87.56 (↑1.38%)
	SVHN	89.82	94.22 (↑4.90%)	96.47 (↑7.40%)	98.79 (↑9.99%)	89.36 (↓0.51%)
	LSUN	88.97	80.94 (↓9.03%)	92.20 (↑3.63%)	95.36 (↑7.18%)	90.15 (↑1.33%)
SVHN	CIFAR-100	93.24	99.23 (↑6.42%)	94.42 (↑1.27%)	94.92 (↑1.80%)	92.38 (↓0.92%)
	CIFAR-10*	94.64	99.79 (↑5.44%)	95.07 (↑0.45%)	96.00 (↑1.44%)	92.91 (↓1.83%)
	LSUN	93.02	99.94 (↑7.44%)	93.40 (↑0.41%)	95.64 (↑2.82%)	91.85 (↓1.26%)
FashionMNIST	CIFAR-100	91.37	99.43 (↑8.82%)	94.00 (↑2.88%)	95.61 (↑4.64%)	63.11 (↓30.93%)
	CIFAR-10*	94.78	99.67 (↑5.16%)	96.28 (↑1.58%)	93.73 (↓1.11%)	64.98 (↓31.44%)
	LSUN	93.53	99.90 (↑6.81%)	97.62 (↑4.37%)	99.28 (↑6.15%)	65.05 (↓30.45%)
CIFAR-100	CIFAR-10	78.88	85.64 (↑8.57%)	80.76 (↑2.38%)	85.02 (↑7.78%)	75.45 (↓4.35%)
	SVHN*	81.94	92.55 (↑12.95%)	86.05 (↑5.02%)	94.37 (↑15.17%)	80.32 (↓1.98%)
	LSUN	77.18	97.36 (↑26.15%)	76.54 (↓0.83%)	87.43 (↑13.28%)	75.72 (↓1.89%)
Avg % improvement			(↑6.73%)	(↑2.73%)	(↑6.31%)	(↓8.57%)

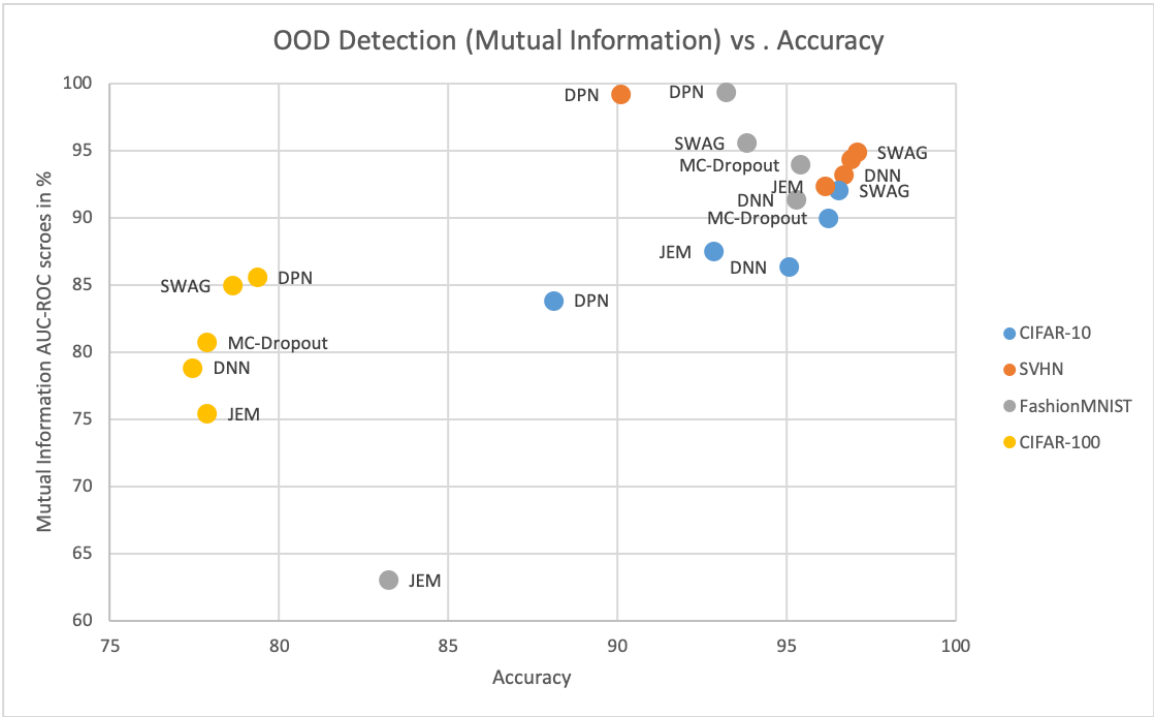


Figure 4.3: Scatter plot of OOD detection vs. accuracy based on mutual information AUC-ROC scores.

network like DPN in combination with differential entropy is quite effective. For the remaining estimators the best performance is achieved based on predictive uncertainty (i.e. entropy) as an OOD scoring metric. This indicates that overall, Bayesian methods

Table 4.4: Out-of-distribution experiment results. AUC-ROC scores are based on ***Differential Entropy***. The values in parenthesis are % improvement of the corresponding estimator w.r.t. DNN, taken as a baseline. An ↑ indicates improvement and ↓ degradation. The asterisks (*) indicate the out-distribution datasets used to train DPN.

Data		(baseline)	Differential Entropy AUC-ROC score (% gain w.r.t. baseline)			
In-distribution	OoD	DNN	DPN	MC-Dropout	SWAG	JEM
CIFAR-10	CIFAR-100*	80.67	85.28 (↑5.71%)	89.77 (↑11.28%)	88.66 (↑9.90%)	85.18 (↑5.59%)
	SVHN	82.71	93.93 (↑13.57%)	96.70 (↑16.91%)	98.09 (↑18.60%)	84.98 (↑2.74%)
	LSUN	85.09	82.49 (↓3.06%)	92.46 (↑8.66%)	93.12 (↑9.44%)	90.52 (↑6.38%)
SVHN	CIFAR-100	92.89	99.26 (↑6.86%)	93.87 (↑1.06%)	90.43 (↓2.65%)	82.90 (↓10.75%)
	CIFAR-10*	94.44	99.80 (↑5.68%)	94.55 (↑0.12%)	92.52 (↓2.03%)	82.98 (↓12.13%)
	LSUN	92.66	99.93 (↑7.85%)	92.84 (↑0.19%)	90.38 (↓2.46%)	79.97 (↓13.70%)
FashionMNIST	CIFAR-100	91.93	99.52 (↑8.26%)	95.30 (↑3.67%)	97.48 (↑6.04%)	62.24 (↓32.30%)
	CIFAR-10*	95.61	99.73 (↑4.31%)	97.33 (↑1.80%)	94.40 (↓1.27%)	64.77 (↓32.26%)
	LSUN	94.34	99.88 (↑5.87%)	98.43 (↑4.34%)	99.71 (↑5.69%)	64.94 (↓31.16%)
CIFAR-100	CIFAR-10	77.73	84.71 (↑8.98%)	79.74 (↑2.59%)	81.54 (↑4.90%)	73.89 (↓4.94%)
	SVHN*	83.54	91.89 (↑10.00%)	86.87 (↑3.99%)	94.20 (↑12.76%)	79.57 (↓4.75%)
	LSUN	74.97	96.82 (↑29.15%)	73.22 (↓2.33%)	79.92 (↑6.60%)	74.04 (↓1.24%)
Avg % improvement			(↑8.60%)	(↑4.36%)	(↑5.46%)	(↓10.71%)

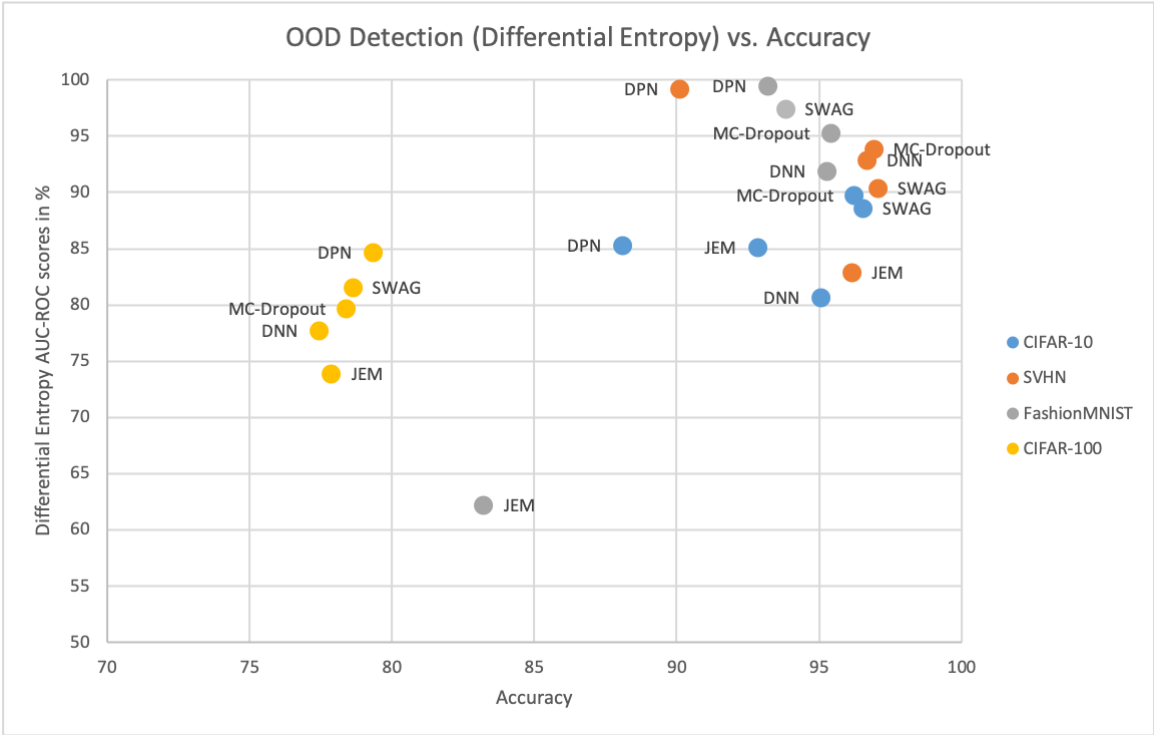


Figure 4.4: Scatter plot of OOD detection vs. accuracy based on differential entropy AUC-ROC scores.

improve the performance on OOD detection, with DPN and SWAG achieving similar performance (although JEM does not achieve improvements over the DNN baseline).

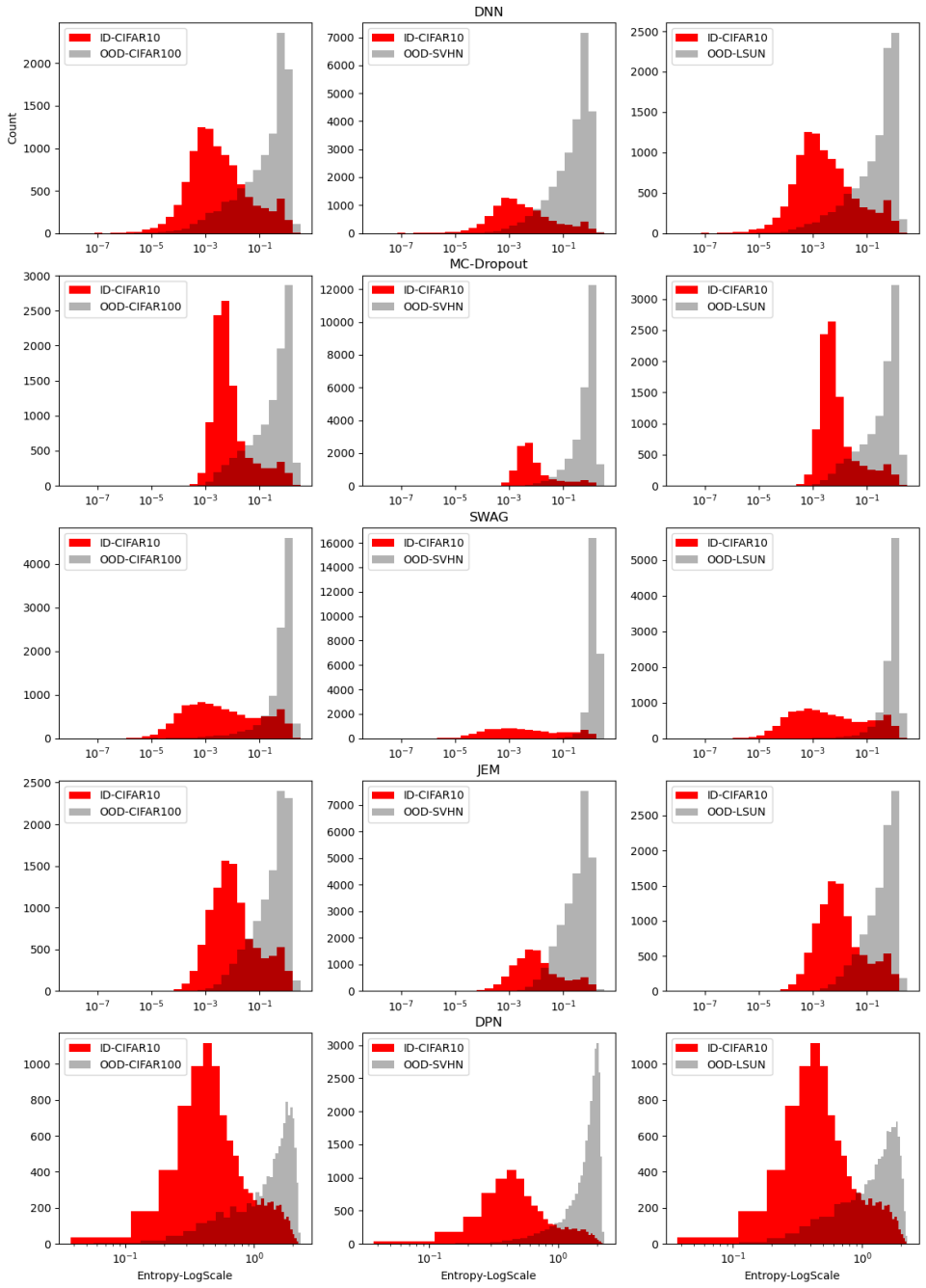


Figure 4.5: Histograms for in and out-of-distribution predictions across estimators and datasets. The x-axis represents predictive uncertainty (i.e. entropy) in log scale.

Additionally, one could resort to heuristic approaches such as for instance devising thresholds on Figure 4.5 for separating ID from OOD samples. There exist multiple ways to achieve this but an intuitive starting point would have been to take into consideration the in-distribution score of the estimator and based on that to appropriately select a threshold representing the 95% percentile. In order to avoid such heuristics we rely on AUC-ROC scores which could operate directly on outputs similar to those depicted in Figure 4.5.

Moreover, to further clarify the impact of the alternative metrics on the final results across the different estimators we direct the reader's attention to Table 4.5 representing their rankings from best (i.e. DPN) to worst (i.e. JEM) in terms of their OOD effectiveness, in addition to establishing any fluctuations caused by these metrics. We observe that for JEM the different metrics do not affect the final results since they are all ranked equally. Instead, for MC-Dropout mutual information performs best by improving overall OOD effectiveness followed by the remaining metrics providing equal contributions. Instead, for SWAG entropy and maximum probability provide the best results followed by differential entropy and mutual information. Last but not least, for DPN it seems that differential entropy and mutual information are the most effective metrics followed by entropy and maximum probability with equal contributions.

Table 4.5: % performance increase w.r.t. DNN for all the evaluation scores, relative ranks are shown in parenthesis. The last row shows average ranks.

	DPN	MC-Dropout	SWAG	JEM
Max prob.	5.895% (2)	2.618% (3)	6.067% (1)	-8.350% (4)
Mutual Info	7.262% (1)	6.243% (2)	5.337% (3)	-8.034% (4)
Entropy	6.480% (2)	6.480% (3)	6.601% (1)	-7.960% (4)
Diff. Entropy	9.232% (1)	4.386% (3)	5.495% (2)	-10.217% (4)
Avg. rank	(1.50)	(2.75)	(1.75)	(4.00)

However, it is worth noting from Table 4.1 that the DPN estimator does not perform well on the base image classification tasks compared to the remaining estimators. Therefore, we conclude that of the estimators compared in this experiment SWAG is the most effective as it achieves strong OOD detection performance, without compromising basic in-distribution classification performance. It also does not require the use of an auxiliary OOD datset during training. Although if the final objective is achieving the best possible OOD detection performance then DPN should be considered.

Finally, in order to provide a holistic overview regarding the performance of point estimate neural networks against equivalent Bayesian neural networks we employ results directly from Liu et al. (2020) depicting state-of-the-art point estimate neural networks for OOD detection. The results of this comparison are represented in Table 4.6

where it is observable the existence of variability in terms of OOD effectiveness among the different OOD datasets. The best performing estimators for each OOD dataset are shown in bold face where it is observable that a Bayesian estimator (ContRank+MC Dropout) attains the highest OOD score on *CIFAR-100* and *SVHN* whereas Mahalanobis and OE attain the highest OOD score on the *LSUN* dataset. Provided the existence of variability regarding the results which to an extent could impair conclusive statements we hypothesise that for future work a larger study in this endeavour might be beneficial.

Table 4.6: Comparison of point estimates vs. Bayesian OOD detection methods based on published results in the literature directly corresponding to our trained estimators (Liu et al., 2020).

Data		Entropy AUC-ROC scores											
		Point Estimates						Bayesian					
S_{ID}	S_{OOD}	DNN	Mahalanobis	ODIN	MSRep	OE	EnergyOOD	DPN	MCD	SWAG	JEM	ContReg+MCD	ContRank+MCD
CIFAR-10	CIFAR-100*	86.27	93.90	85.59	91.23	93.30	92.60	85.60	89.92	91.89	87.35	93.62	**96.60**
	SVHN	89.72	97.62	91.96	**99.48**	98.40	90.96	98.90	96.25	98.62	89.22	98.36	96.96
	LSUN	88.83	96.30	90.35	96.05	**97.60**	94.24	83.30	92.04	95.12	89.84	93.62	95.61
CIFAR-100	CIFAR-10	78.25	81.34	74.54	81.49	75.70	76.61	**85.15**	80.70	84.92	77.64	68.25	69.47
	SVHN*	81.52	86.01	67.26	77.54	86.66	73.99	92.64	85.59	94.16	81.22	98.98	**99.79**
	LSUN	77.22	**93.90**	78.94	79.05	79.71	79.23	86.38	76.58	87.22	77.54	64.21	61.32

4.4 Conclusion

In this work we ask if Bayesian methods improve OOD detection over DNNs for image classification tasks? We investigated four recent methods and compared them with a baseline DNN on four different datasets. The underlying model architecture was the same for each estimator (i.e. WideResNet28x10) in order to retrieve unbiased results. All methods were evaluated in OOD detection using AUC-ROC on different metrics—*entropy*, *mutual information*, and *differential entropy*. Our findings show that the Bayesian methods do indeed improve performance at the OOD detection task over the DNN baseline. DPN coupled with differential entropy for OOD scoring achieved the best OOD detection performance, but in-distribution classification accuracy was degraded. SWAG achieved a particularly good trade-off balance between in-distribution classification accuracy and OOD detection performance. We acknowledge the fact that the auxiliary information presented to DPN might be considered as an advantage compared to other existing methods but since it was part of the overall methodology originally introduced in the DPN work and provided that it was already deployed in equivalent comparisons (Malinin and Gales, 2018) we considered that it would have been best to not alter its original methodology. However, preliminary results suggest, that there is an opportunity to further advance OOD detection performance.

CHAPTER

FIVE

ADVERSARIAL ROBUSTNESS IN BAYESIAN NEURAL NETWORKS

This chapter evaluates the robustness ability of Bayesian neural networks to withstand against adversarial inputs in an effort to answer research question **RQ4**.

5.1 Introduction

In addition to the OOD detection problem described in Chapter 4, there is another equally important issue impairing deep neural networks performance, that is adversarial attacks (Akhtar and Mian, 2018; Ozdag, 2018). For a trained deep neural network estimator, adversarial noise which is imperceptible to the human visual sensory system (i.e. similar to optical illusions) can be constructed and incorporated into any data type (e.g. image, audio, text) causing the underlying estimator to classify it with high confidence as a separate category from its actual true category. Since many deep learning based image recognition systems are being deployed in real world applications, such adversarial attacks can have serious societal consequences. Therefore, developing strategies to mitigate against them is a significant open issue.

In this work we describe an evaluation of the effectiveness of different Bayesian neural networks against adversarial attacks utilising a selection of image classification datasets. In this experiment we are interested in identifying whether Bayesian neural networks can withstand untargeted adversarial attacks given that they have demonstrated their efficacy in OOD detection. This will allow us to better understand the implications in critical domain applications since it implies that the predictions from state-of-the-art deep learning models cannot be trusted (Alemi et al., 2018; Hill, 2019; Kumar et al., 2019; Marin et al., 2012). Finally, we investigate the potential benefit from

utilising adversarial defence mechanisms for both tasks of detecting OOD data, and, withstanding against adversarial attacks on the in-distribution test data.

For that purpose we designed a new experiment including the usual DNN acting as a control baseline, two Bayesian estimators MC-Dropout, and SWAG, and three adversarial defence techniques Top-k, Randomised Smoothing, and MMLDA trained on the in-distribution dataset *CIFAR-10* and evaluated on the following OOD datasets *CIFAR-10-Adv.*, *CIFAR-100*, *SVHN*, and *LSUN* before and after introducing each adversarial defence technique during the training process of each estimator. This formulation of adversarial learning for OOD detection stems from Definition 8 in Chapter 2.

The **key questions** investigated in this chapter are presented below stemming from research question **RQ4** introduced in Section 1.1 of Chapter 1:

1. Are Bayesian neural networks by default capable of detecting adversarial examples?
2. Could simple defence mechanisms *Top-k*, *Randomised Smoothing*, *MMLDA*, improve overall OOD detection in Bayesian neural networks, in addition to withstanding against adversarial attacks?

The **key contributions** of this work are the following, and they contribute towards **CB3** introduced in Section 1.2 of Chapter 1:

- We demonstrate that Bayesian neural networks are not robust to adversarial noise without prior exposure or explicit training on such noise.
- We find that the effect of different adversarial defence methods on Bayesian neural networks overall improve robustness against adversarial attacks and OOD detection on the unseen test datasets but at the same time cause the accuracy on the in-distribution test dataset to degrade.

5.2 Experiment Design

In this experiment we test the efficacy of Bayesian neural networks in detecting adversarial perturbations with and without the contribution of adversarial defence methods such as Top-k, Randomised Smoothing, and MMLDA. To achieve that we utilised a conventional DNN as our control baseline which we compared with two Bayesian neural network approaches MC-Dropout, and SWAG trained on the *CIFAR-10* dataset

and evaluated on the following metrics: accuracy, Eq. 2.34, predictive uncertainty (i.e. entropy), Eq. 2.8, and AUC-ROC scores, Eq. 2.40. We selected MC-Dropout and SWAG based on the fact that they seem to provide the best trade-off between in-distribution accuracy and OOD detection (see Chapters 3, and 4) while reducing some computational constraints during training and inference on the construction of adversarial examples.

First, we evaluated the ability of each estimator against untargeted adversarial examples generated with *Projected Gradient Descent (PGD)* (Madry et al., 2018; Ozdag, 2018) which is considered quite effective as an attack. For PGD we used the following hyperparameter values $\epsilon = 0.1$ and $\alpha = 0.01$ for ten iterations (Kurakin et al., 2017). Second, we introduced three recently proposed defence techniques against adversarial examples and evaluated their efficacy on the (i) *clean test set*, (ii) *adversarially corrupted test set*, and finally on the (iii) *OOD detection task* utilising the adversarially corrupted test set of *CIFAR-10* and the corresponding clean test set of *CIFAR-100*, *SVHN*, and *LSUN* as OOD data. The defence methods evaluated against adversarial examples are: *Randomised Smoothing* (Cohen et al., 2019), *Sparsify k-winners take all* (*Top-k*) (Xiao et al., 2020), and *Max-Mahalanobis Linear Discriminant Analysis* (MMLDA) (Pang et al., 2020).

In order to train estimators with MMLDA and RandSmooth we used the same hyperparameter settings as in the original papers that recommended them (Pang et al., 2020), (Cohen et al., 2019). For instance, for MMLDA we set the variance value equal to 10 for *CIFAR-10* in order to compute the Max-Mahalanobis centres on the features from the penultimate layer of each estimator.

5.3 Results

We first illustrate the effect of adversarial noise on the ability of different estimators to perform the underlying classification task on *CIFAR-10* in the presence of adversarial noise. Table 5.1 shows the performance of each estimator on the *CIFAR-10* classification problem on a clean dataset with no adversarial noise, while Figure 5.1 compares the defence techniques against no defence on the same clean test set. For Randomised smoothing, the values in parentheses denote the percentage of abstained predictions on the total number of instances from the test set. Randomised smoothing had a negative impact on the in-distribution performance across all estimators indicating that robustness might be at odds with accuracy (Tsipras et al., 2019). MMLDA and Top-k seem to have a positive effect on DNN and MC-Dropout but not on SWAG.

Table 5.1: Accuracy on *CIFAR-10* clean test set for each defence technique. Percentage of abstained predictions is indicated in parenthesis for the RandSmooth approach.

Estimators	No Defence	Top-k	RandSmooth	MMLDA
DNN	95.06	94.52	86.25 (13.00)	95.18
MC-Dropout	96.22	94.43	86.98 (13.39)	95.21
SWAG	96.53	91.73	79.68 (20.32)	91.30

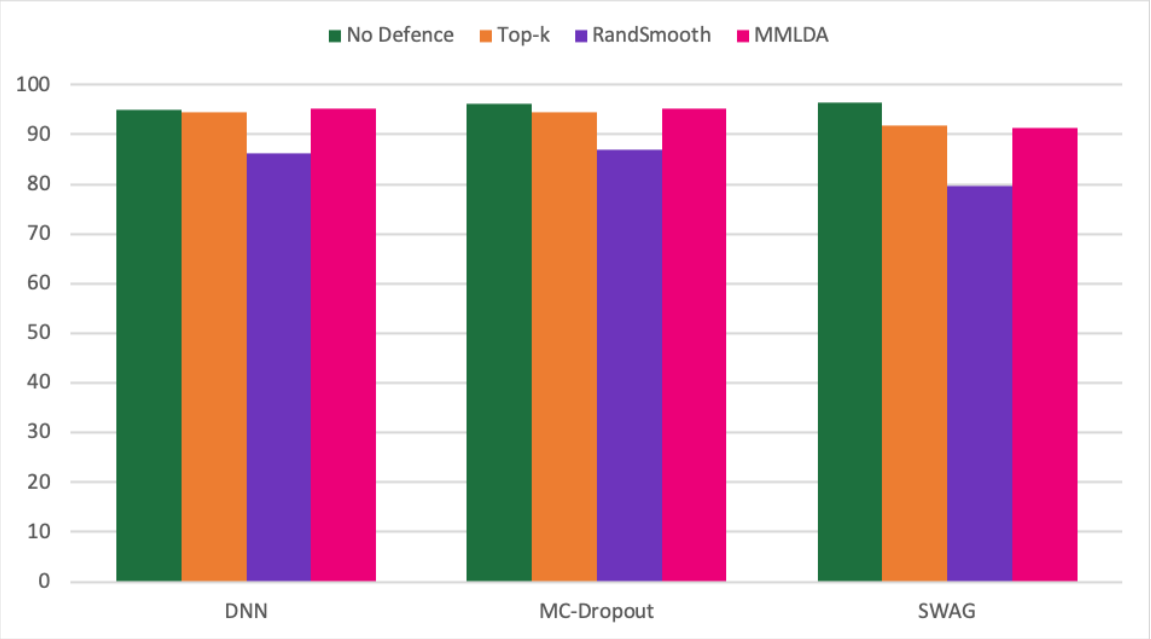


Figure 5.1: Evaluation of defence techniques vs non-defence on clean test set of *CIFAR-10*.

Table 5.2 summarises the performance of the same estimators when adversarial noise is added to the test set. The first column in Table 5.2 corresponds to the accuracy on the adversarially corrupted test set (i.e. all 10K data points in test set) for each estimator without applying any adversarial defence mechanism. Lower values indicate that the adversarial attack was successful and managed to force the estimators to misclassify instances with high confidence. The impact of adversarial noise is evident in these results. When there is no defence mechanism utilised then classification accuracy plummets to less than 2% for all estimators. The addition of defence techniques, especially MMLDA and randomised smoothing indicate improvement in performance.

Table 5.2: Accuracy on *CIFAR-10* test set corrupted with adversarial noise.

Estimators	No Defence	Top-k	RandSmooth	MMLDA
DNN	1.15	11.41	62.65 (27.75)	45.71
MC-Dropout	1.94	7.28	88.85 (17.98)	47.90
SWAG	0.55	0.79	36.80 (20.97)	47.95

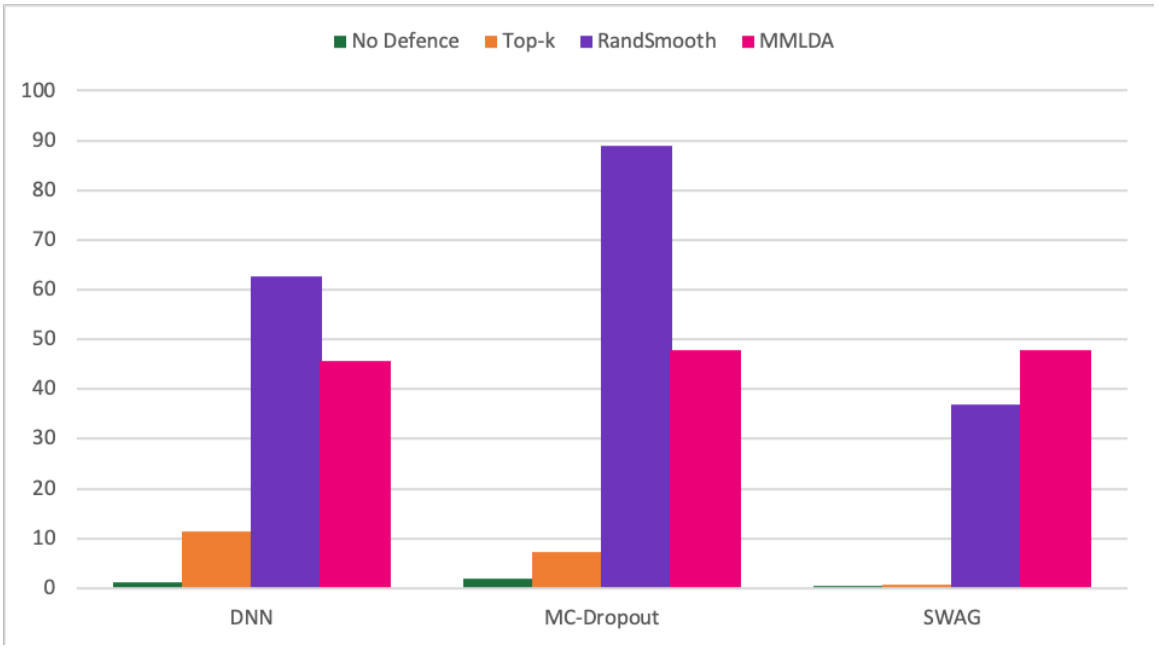


Figure 5.2: Evaluation of defence techniques vs non-defence on the adversarially corrupted test set of*CIFAR-10*.

In addition, Table 5.3, shows that the estimator predictions can be used to identify adversarial examples in the same manner that OOD examples were identified by measuring the predictive uncertainty (i.e. entropy) over the final predictions. It compares the entropy scores generated for an in-distribution test set versus those generated for adversarial examples and uses AUC-ROC scores to measure the ability of an estimator to distinguish them. First, notice that in line with the results in the previous section the Bayesian estimators perform better at this task than the baseline DNN when no defence against adversarial examples is used. SWAG, in particular, seems quite effective. When adversarial defence mechanisms are used, the performance increases dramatically. Top-k in particular seems effective in this occasion.

Table 5.3: Out-of-distribution detection results for all defence methods on clean vs adversarially corrupted *CIFAR-10*. Scores represent entropy based AUC-ROC in percentage.

Data		Entropy AUC-ROC score		
Defence	OOD	DNN	MC-Dropout	SWAG
No defence	CIFAR-10-Adv.	13.67	36.54	73.81
Top-k	CIFAR-10-Adv.	**99.83**	**99.83**	**98.90**
MMLDA	CIFAR-10-Adv.	53.15	85.65	70.38
RandSmooth	CIFAR-10-Adv.	62.68	57.06	48.29

Finally, Table 5.4 shows whether the defence techniques provide any improvements with regard to the underlying OOD detection task when the estimators are trained on

the in-distribution data *CIFAR-10* and evaluated on the OOD data *CIFAR-100*, *SVHN*, and *LSUN*. The values represent AUC-ROC scores obtained based on predictive uncertainty (i.e. entropy) of the predictions.

Notice that without introducing any adversarial defence approach on the task of OOD detection Bayesian methods like SWAG by default obtain high detection rate against OOD inputs outperforming the DNN baseline. Interestingly when introducing adversarial defence techniques we observe a variation in the OOD detection results. Take for instance Top-k and MMLDA which improve OOD detection for the DNN baseline on two datasets compared to no defence. The opposite phenomenon is observed for RandSMooth which seem to degrade overall OOD detection. In summary, Top-k and MMLDA improve overall OOD detection performance for the DNN baseline, instead, MMLDA improves OOD detection only for MC-Dropout.

Table 5.4: Out-of-distribution detection results. Scores represent entropy based AUC-ROC in percentage.

Data		Entropy AUC-ROC score		
ID {*CIFAR-10*}	OOD	DNN	MC-Dropout	SWAG
No defence	CIFAR-100	86.27	89.92	**91.89**
	SVHN	89.72	96.25	**98.62**
	LSUN	88.83	92.04	**95.12**
Top-k	CIFAR-100	**90.59**	90.45	84.72
	SVHN	91.20	92.62	**94.61**
	LSUN	**92.42**	92.39	89.81
MMLDA	CIFAR-100	**99.87**	99.24	79.78
	SVHN	99.74	**99.75**	84.56
	LSUN	**99.93**	99.67	81.72
RandSmooth	CIFAR-100	62.91	**69.33**	60.21
	SVHN	42.62	63.22	**66.48**
	LSUN	61.94	**69.10**	61.23

5.4 Conclusion

In this work we studied the effectiveness of BNN in OOD detection against adversarially corrupted datasets. We demonstrated that Bayesian neural networks by default do not possess the ability to cope with adversarial examples, despite performing better than DNN with respect to OOD detection. Although adversarial defence techniques overall degrade accuracy on the clean test set, at the same time, they robustify against adversarial examples across estimators and occasionally improve OOD detection. Randomised smoothing performed the best against adversarial inputs, followed by MMLDA. With regard to the OOD detection task it seems that Top-k improved performance for DNN while MMLDA improved performance for DNN and MC-Dropout.

CHAPTER

SIX

OBJECTIVES FOR OUT-OF-DISTRIBUTION DETECTION

This chapter introduces two custom objective functions for out-of-distribution detection utilising auxiliary information as a form of regularisation extending any common estimator in an effort to answer research question **RQ5**.

6.1 Introduction

Inspired by recent progress on the contrastive learning paradigm (Chen et al., 2020; Goyal et al., 2019; Henaff, 2020; Li and Vasconcelos, 2020; Oord et al., 2018; Poole et al., 2019) in this chapter we propose two novel objectives for out-of-distribution (OOD) detection that exploit the abundance of unlabelled data similar to (Hendrycks et al., 2019a). Unfortunately, the outlier objective function suggested in (Hendrycks et al., 2019a) is task dependent constraining the user to switch between classification and density estimation. Instead, we demonstrate empirically that our methods operate surprisingly similarly to density estimators, ideally dispersing uncertainty on out-of-distribution estimates uniformly outside the support of in-distribution data while simultaneously performing classification. Interestingly, our methods operate directly on the probability distributions of the estimates instead of the latent space which is the norm in contrastive learning, providing satisfactory decorrelation between in- and out-of-distribution samples, as well as being competitive with existing approaches. The proposed objectives presented in this chapter to an extent resemble Definition 6 in Chapter 2.

The main difference is the set of collections of unknown test distributions $\mathcal{P}$ which is approximated by P_{train} containing only one element, this is the unlabelled auxiliary OOD dataset distribution. Instead, in the actual Definition 6 the set P_{train} often entails more than one distribution. Finally, we empirically study the role of regularisation in OOD detection and robust classification.

The **key questions** investigated in this work are presented below and stem from research question **RQ5** introduced in Section 1.1 of Chapter 1:

- Can we leverage the paradigm of contrastive learning to improve OOD detection in neural networks?
- What is the role of explicit regularisation in OOD detection, does additional regularisation improve or degrade OOD detection?

The **key contributions** of this work contribute towards **CB4** introduced in Section 1.2 of Chapter 1:

- A new objective based on the cosine angle between in-distribution and out-of-distribution predictions. Similar to a contrastive loss the proposed objective enforces a separation over predictions as a form of regularisation leading to improved OOD detection.
- A new objective enforcing a margin based on the cosine angle between in- and out-of-distribution predictions, in addition to explicit regularisation imposing a uniform distribution on OOD predictions and a dirac delta on in-distribution estimates. This has the benefit of distinctly separating in-distribution from OOD estimates resulting in trained estimators with improved OOD detection.

6.2 Contrastive Objectives

The majority of objectives used in supervised machine learning (e.g. cross-entropy, mean square error, log-likelihood) have the same goal: to induce a cost in order for the underlying estimator to directly learn a target value or a set of values from a specific input. In contrast, ranking objectives strive to predict similarities (i.e. relative distances) between inputs, thus the underlying task is often identified as metric learning. The key idea is to learn a representation function mapping objects into a latent embedding space preserving similarities, such that similar objects are attracted together while dissimilar objects get repelled. The same idea is also prevalent in contrastive learning. SimCLR (Chen et al., 2020) is an illustrative example of contrastive learning. It maximises the agreement in latent representations via a contrastive objective between pairs of inputs responsible for mapping objects from the same category together in the latent space and those from different categories to different points if their similarities exceed a margin.

In the remainder of this section an introduction to contrastive and ranking objectives is presented starting with the formulation of SimCLR and then proceeding onto the formulation of ranking objectives building on the same notation of SimCLR. Let $t_i, t_j \sim \mathcal{T}$ denote distinct augmentation operations for any input x such that when evaluated on an estimator f they produce a pair of latent representations $(z_i, z_j) = f_\theta(t_{i,j}(x)), \forall x \in \mathcal{X}$. Additionally, let $\cos(u, v) = u^\top v / \|u\| \|v\|$ define a similarity score function, in this case the cosine angle between vectors u and v. Then provided a pair of latent representations (z_i, z_j) the overall SimCLR objective is formulated in Eq. 6.1 with τ representing the temperature scaling parameter.

$$L(z_i, z_j) = -\ln \frac{e^{\cos(z_i, z_j)/\tau}}{\sum_{j=1}^{2n} \mathbb{1}_{i \neq j} \left\{ e^{\cos(z_i, z_j)/\tau} \right\}}. \tag{6.1}$$

The main concept of SimCLR can also be extended to pairwise ranking objectives learning a similarity metric, for instance, an ℓ_2-norm on embedded representations (z_i, z_j). Given a set of latent representations $\{z_1, \ldots, z_n\}$ one chooses an anchor sample z, considering the remaining representations as either a positive or a negative sample z^*, based on the targets $y = \{0, 1\}$ such that $y = 1$ corresponds to a positive pair (e.g. an anchor image and an image semantically similar to the anchor) and $y = 0$ to a negative pair (e.g. an anchor image and an image semantically different from the anchor), then a **pairwise ranking objective** strives to learn representations with small distance

d between positive pairs (z, z^*) and a distance greater than a margin γ for negative pairs, (z, z^*) such that:

$$L(z, z^*, y) = \begin{cases} y\, d(z, z^*) & \text{if } y = 1 \\ (1 - y)\, \max(0, \gamma - d(z, z^*)) & \text{if } y = 0 \end{cases} \tag{6.2}$$

Similarly, in the case of **triplet ranking objective** instead of pairs of positive and negative samples as illustrated in Eq. 6.2 one directly utilises triplets (z, z^-, z^+) with an anchor z, a positive z^+, and a negative z^- sample. The goal is to learn representations with greater distance for the anchor and the negative sample $d(z, z^-)$ than the anchor and the positive sample $d(z, z^+)$. The final triplet objective is formulated as:

$$L(z, z^+, z^-) = \max(0, \gamma + d(z, z^+) - d(z, z^-)) \tag{6.3}$$

$$\begin{cases} \text{Easy triplets:} & \text{if } d(z, z^-) > d(z, z^+) + \gamma \\ \text{Semi-hard triplets:} & \text{if } d(z, z^+) < d(z, z^-) < d(z, z^+) + \gamma \\ \text{Hard triplets:} & \text{if } d(z, z^-) < d(z, z^+) \end{cases}$$

Often in triplet ranking objectives there exist three common scenarios depending on the distance between the combination of triplets in the latent embedding space. For instance, if a positive sample is closer to an anchor than a negative sample then it is known as an easy triplet, instead, if a negative sample is closer to an anchor than a positive sample then it is known as a hard triplet. Finally, if both positive and negative samples have approximately the same distance from the anchor then it is known as a semi-hard triplet

6.3 Novel Contrastive Objectives for OOD Detection

In this section we propose two new objective functions for OOD detection that are based on contrastive loss. The first is Contrastive Regularised (ContReg) objective, a new method based on the cosine angle between in-distribution (ID) and out-of-distribution (OOD) predictions utilising the abundance of unlabelled data that is typically available when building classification estimators. Let (x_{id}, y_{id}) and x_{ood} represent two data points sampled from the in- and out-of-distribution data respectively where x_{id} represents the anchor and x_{ood} represents a negative sample, and, define $p_{id} = p(y_{id}|f_\theta(x_{id}))$ to be the softmax probability for x_{id} and $p_{ood} = p(y_{ood}|f_\theta(x_{ood}))$ the softmax probability for x_{ood}. Then our objective is formulated as follows:

$$L(x_{id}, x_{ood}, y_{id}) = \underbrace{-\mathbb{E}\left[\ln\ p(y_{id}|f_\theta(x_{id}))\right]}_{\text{cross-entropy loss}} + \underbrace{\lambda \cos(p_{id}, p_{ood})}_{\text{cosine-regularisation}} \tag{6.4}$$

$$\cos(p_{id}, p_{ood}) = \frac{{p_{id}}^\top p_{ood}}{\|p_{id}\|\,\|p_{ood}\|}$$

Notice that the regularisation strength λ is a hyperparameter typically chosen using a validation set, and whenever $\lambda = -1$ then the underlying objective becomes a minimax optimisation formulation similar to those found in adversarial learning paradigms (Pang et al., 2020), with the additional benefit that is faster to train an estimator with this objective since it avoids computing gradients for worst-case perturbations on the inputs, and, at the same time it strives to disentangle the predictions based on the probabilities of the ID S_{ID} and OOD S_{OOD} data. The goal of the objective is to lower the cross-entropy error on S_{ID} while at the same time increase the cosine angle between S_{ID} and S_{OOD}. This synergy of the minimax optimisation formulation can also be found in energy-based objectives (Grathwohl et al., 2020; Liu et al., 2020) where the intention is to lower the energy for similar samples while at the same time increase the energy on dissimilar inputs. Algorithm 2 shows the implementation procedure of the Contrastive Regularised (ContReg) objective.

Algorithm 2 Contrastive Regularised Objective

procedure CONTREG(x_{id}, x_{ood}, y_{id})
- $f_\theta \leftarrow \theta$ ▷ initialise estimator
- $p_{id}, p_{ood} \leftarrow f_\theta(x_{id}), f_\theta(x_{ood})$ ▷ calc. probab. for logits $\in (S_{ID}, S_{OOD})$
- $CE \leftarrow -\mathbb{E}[\ln\ p(y_{id}|x_{id})]$ ▷ compute cross-entropy for (x_{id}, y_{id})
- $\cos \leftarrow \frac{{p_{id}}^\top p_{ood}}{\|p_{id}\|\,\|p_{ood}\|}$ ▷ compute cosine for probabilities p_{id}, p_{ood}
- $L \leftarrow CE + \lambda \cos$ ▷ compute final regularised loss
- $\theta_{t+1} = \theta_t - \eta\, \nabla_\theta L$ ▷ compute gradient w.r.t params θ and backprop errors

end procedure

Our second contribution, is Contrastive Ranking (ContRank), a new ranking objective for OOD detection that is inspired by prior work on contrastive objectives (Chen et al., 2020; Li et al., 2021; Oord et al., 2018). We propose utilising the cosine similarity as a metric learning function inside the ranking objective in addition to explicit ℓ_2 and ℓ_1 regularisation for S_{ID} and S_{OOD} respectively. This leads to the following objective function.

$$L(x_{id}, x_{ood}, y_{id}) = \underbrace{\max(0, \gamma + \cos(p_{id}, p_{ood}))}_{\text{ranking objective}} + \underbrace{\lambda_1 \left| p_{ood} - 1/k \right|}_{\ell_1\text{-regularisation on } \mathcal{S}_{OOD}} + \underbrace{\lambda_2 \left\| y_{id}\, p_{id} - \alpha \right\|}_{\ell_2\text{-regularisation on } S_{ID}} \tag{6.5}$$

$$\cos(p_{id}, p_{ood}) = \frac{{p_{id}}^\top p_{ood}}{\|p_{id}\| \, \|p_{ood}\|}$$

Notice that $k \in \mathbb{Z}$ refers to the number of categories in the ID data S_{ID}, and y_{id} represents a one-hot encoding of the targets, while $\alpha \in \mathbb{R}$ is a user defined scalar indicating the ideal desired confidence for the in-distribution predictions.

Algorithm 3 Contrastive Ranking Objective

procedure CONTRANK(x_{id}, x_{ood}, y_{id})
- $f_\theta \leftarrow \theta$ ▷ initialise estimator
- $p_{id}, p_{ood} \leftarrow f_\theta(x_{id}), f_\theta(x_{ood})$ ▷ calc. probab. for logits $\in (S_{ID}, S_{OOD})$
- $\ell_1 \leftarrow \lambda_1 \left| p_{ood} - 1/k \right|$ ▷ compute ℓ_1-regularisation for p_{ood}
- $\ell_2 \leftarrow \lambda_2 \left\| y_{id}\, p_{id} - \alpha \right\|$ ▷ compute ℓ_2-regularisation for p_{id}
- $\cos \leftarrow \frac{{p_{id}}^\top p_{ood}}{\|p_{id}\| \, \|p_{ood}\|}$ ▷ compute cosine for probabilities p_{id}, p_{ood}
- $L \leftarrow \max(0, \gamma + \cos) + \ell_1 + \ell_2$ ▷ compute the final ranking loss
- $\theta_{t+1} = \theta_t - \eta \, \nabla_\theta L$ ▷ compute gradient w.r.t params θ and backprop errors

end procedure

There are a number of hyperpameters $\{\gamma, \lambda_1, \lambda_2\}$ which can be tuned utilising a validation set, where γ denotes a margin reducing correlation between probabilities p_{id} and p_{ood} when the cosine similarity exceeds it, and $\lambda_1,\ \lambda_2$ refer to the regularisation strength. Algorithm 3 shows the implementation procedure of the Contrastive Ranking (ContRank) objective. Next, we describe the experiments to evaluate these objectives on synthetic and real datasets.

6.4 Baseline Methods Used in Evaluation Experiments

In the experiments described in this chapter the newly proposed objective functions described in the previous section are compared to baseline objectives for OOD detection proposed in the literature. This section describes each of these.

One of the early established baseline approaches for OOD detection is the maximum softmax probability (MSP) (Hendrycks and Gimpel, 2017) which states that correctly classified instances are usually assigned greater probability than erroneous or out-of-distribution instances, therefore OOD detection is based on AUC-ROC scores from MSP resembling binary classification between positive and negative instances, with positive instances representing predictions assigned greater probability than negative instances. A different approach to OOD detection is the Mahalanobis estimator (Lee et al., 2018a), measuring the Mahalanobis distance from outputs to the in-distribution data.

$$L(x, \mu_k, \mathbf{\Sigma}) = \max_k (f(x) - \mu_k)^\top \mathbf{\Sigma}^{-1} (f(x) - \mu_k). \tag{6.6}$$

A key assumption is that the estimator's penultimate layer output $f(x)$ conditioned on the categories y follows a multivariate Gaussian density $\mathbb{P}(f(x)|y = k) = \mathcal{N}(f(x)|\mu_k, \mathbf{\Sigma})$ with μ_k being the empirical mean for each category k and $\mathbf{\Sigma}$ the empirical covariance.

Next is the ODIN estimator (Liang et al., 2018) utilising temperature scaling τ in addition to adversarial input perturbations δ to produce calibrated softmax scores identifying ID data from OOD data exceeding some threshold T.

$$L(x, \delta, \tau) = \begin{cases} 1 & \text{if } p(y = k|x + \delta; \tau) \leq T \\ 0 & \text{if } p(y = k|x + \delta; \tau) > T. \end{cases} \tag{6.7}$$

The multiple semantic representation (MSRep) estimator (Shalev et al., 2018) introduces the idea of representing the corresponding word of each target category y with multiple word embedding representation estimators $g_1, \ldots, g_m$ such that $y = \{g_1(y), \ldots, g_m(y)\}$, and measuring the cosine similarity among the predicted target embeddings $f(x)$ and the embeddings from the multiple semantic representations $g(y)$ over all m different embeddings.

$$L(x, y) = \sum_{i=1}^{m} \cos(g_i(y), f_i(x)). \tag{6.8}$$

The outlier exposure (OE) estimator (Hendrycks et al., 2019a) builds on the idea of exposing the estimator to an auxiliary dataset of outliers disjoint from the in-distribution test dataset to improve OOD detection. The final objective formulation minimises the errors for both in- and out-of-distribution estimates. Notice that the outlier objective L_{OE} is task dependent resulting in different formulation for classification or density estimation.

$$L(x_{id}, y_{id}, x_{ood}) = \mathop{\mathbb{E}}_{S_{ID}} \left[\ln p(y_{id}|f(x_{id}))\right] + \lambda \mathop{\mathbb{E}}_{S_{OOD}} \left[L_{OE}(f(x_{ood}))\right]. \tag{6.9}$$

The energy based estimator (EnergyOOD) (Liu et al., 2020) utilises a calibrated energy score function $E(x) = -\tau \log \sum_y e^{-f(x)[y]/\tau}$ via temperature scaling with parameter τ parameterised by a neural network $f(x)$ for each category y. The overall objective is formulated as minimising the errors for the energy score function between in- and out-of-distribution data.

$$L(x_{id}, y_{id}, x_{ood}) = \mathop{\mathbb{E}}_{S_{ID}} \left[E(x_{id})/\tau - \ln \sum_k e^{E_k(x_{id})/\tau} \right] + \lambda(E(x_{id}) + E(x_{ood})). \tag{6.10}$$

If the energy score is large exceeding some threshold T then the estimates correspond to OOD inputs, and otherwise they correspond to in-distribution (ID) inputs.

6.5 Evaluation Experiments on Synthetic Data

In order to measure the efficacy of our proposed objectives for OOD detection we designed an experiment utilising synthetic data. We are interested primarily in verifying the hypothesis that our proposed objectives in combination with unlabelled data could improve (i.e. robustify) overall inference on OOD test data. The train data S_{ID} consists of instances sampled from three Gaussians with varying standard deviation σ, representing the different classes in a multi-class classification setting. The different subset splits *train* $\sim S_{ID}^{train}$, *test* $\sim S_{ID}^{test}$ and *test OOD* $\sim S_{OOD}$ over the synthetic dataset are depicted in Figure 6.1.

Figure 6.1: Synthetic dataset comprised of ID train data (a), ID test data (b), OOD test data (c), and finally the union of ID and OOD (yellow colour) test data (d).

6.5.1 Experiment Design

To evaluate the effectiveness of the proposed objective functions we utilised a multilayer perceptron (MLP) neural network, trained on the synthetic dataset to verify that it can attain high accuracy while at the same time maintaining high OOD detection score. Usually we would expect that a property leading to accurate estimators might also imply increased robustness during inference, although, this is not always the case for neural networks.

In this experiment the MLP estimator is trained on the synthetic dataset utilising different objectives with cross-entropy (CE) being the baseline, cross-entropy and MC-Dropout (CE+MCD) employing MC-Dropout during training and inference, and finally, with the newly proposed contrastive regularised objective (ContReg) and contrastive ranking objective (ContRank). During inference, we test each objective on the OOD test data S_{OOD} constructed of instances sampled from four Gaussians displaced at different locations compared to the ID test data S_{ID}^{test}.

To record the efficacy of each objective at detecting OOD data we first measure the regular accuracy of estimators trained using each objective on the ID test data, S_{ID}^{test}.

Additionally, we also utilise three different metrics including confidence, entropy, and mutual information in order to capture the ability of each objective function to detect OOD instances during inference. The main questions of interest are:

- Do our proposed objectives surpass the cross-entropy baseline at OOD detection?
- Does additional explicit regularisation like Dropout during training and inference provide any benefit in OOD detection?

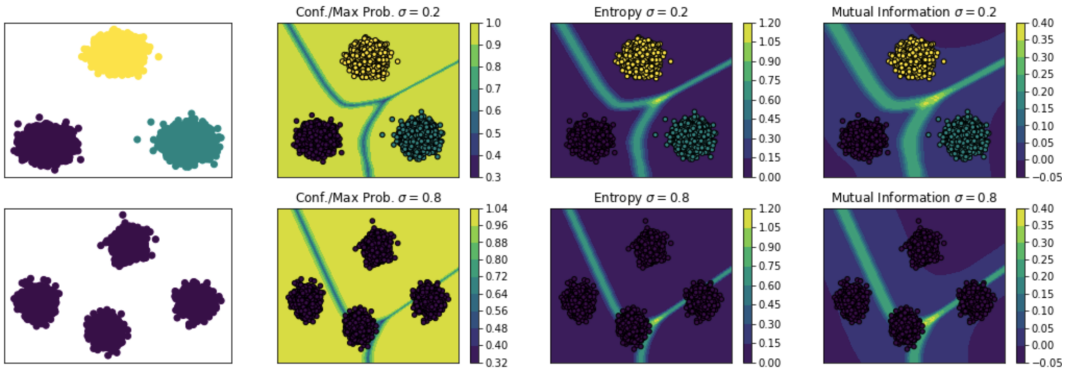
6.5.2 Results

In Table 6.1 we present the results of this experiment using accuracy and AUC-ROC scores based on confidence, entropy, and mutual information obtained from the different objectives: cross-entropy (CE), cross-entropy and MC-Dropout (CE+MCD), contrastive regularised objective (ContReg), and contrastive ranking objective (ContRank). Moreover, Figure 6.2 depicts the different decision boundaries learned using each objective function overlaid with ID test data (1st row) and OOD test data (2nd row). Notice that the colour bar indicates the OOD score of the estimator regarding its predictions, for instance, the confidence OOD score depicted in Figure 6.2 should be high for the ID data and low for OOD data, while entropy representing the predictive uncertainty in addition to mutual information should be low for ID data and high for OOD data.

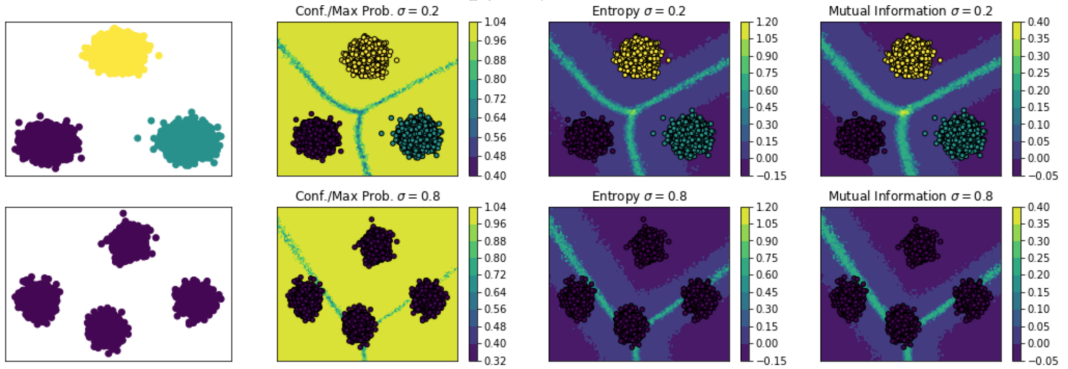
Table 6.1: Accuracy and AUC-ROC scores for each objective function and metric represented in percentages (%).

Objectives	Accuracy	AUC-ROC scores		
		Confidence	Entropy	Mutual Information
CE	100	61.64	61.61	63.62
CE+MCD	100	75.14	73.63	73.56
ContReg (ours)	**100**	**99.99**	**99.99**	**99.99**
ContRank (ours)	**100**	**99.99**	**99.99**	**99.99**

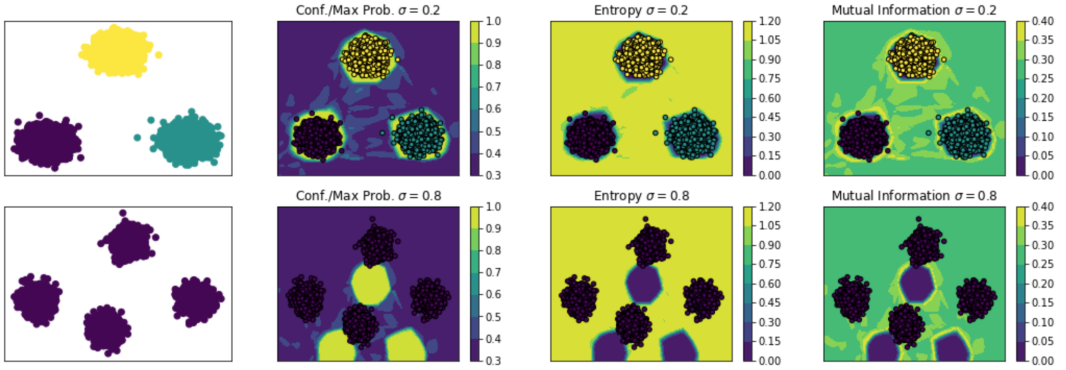
As it is shown in Table 6.1 even though all objective functions achieve the best possible accuracy on the ID test data, nevertheless, only our proposed methods achieve near optimal OOD detection when presented with ambiguous test data. For instance, from Figure 6.2 it is evident that an estimator trained with cross-entropy on a classification problem behaves equivalently to a max-margin estimator whereas our recommended objectives behave equivalently to a density estimator. This indicates that the choice of objective and regularisation denote a crucial role contributing towards the final behaviour of the estimator.



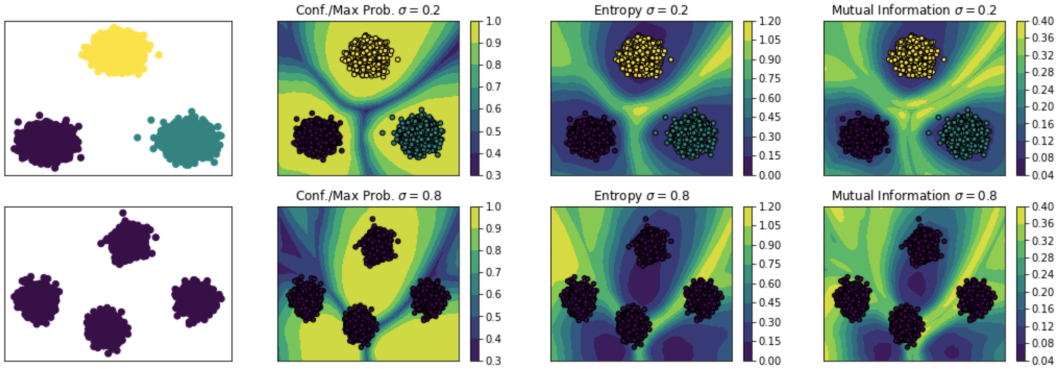
(a) CrossEntropy objective as baseline.



(b) CrossEtnropy+MCD objective.



(c) ContReg objective (ours).



(d) ContRank objective (ours).

Figure 6.2: Decision boundaries on objectives for ID (1st row) & OOD (2nd row) test data.

6.6 Evaluation Experiments on Real Data

Based on their strong performance on the synthetic datasets, to further evaluate the performance of the proposed objective functions we designed an experiment using five well-known image classification datasets: *CIFAR-10* (Krizhevsky, 2009), *CIFAR-100* (Krizhevsky, 2009), *SVHN* (Netzer et al., 2011), *FashionMNIST* (Xiao et al., 2017), and *LSUN* (Yu et al., 2015) (utilising the *bedroom* scene as OOD data for inference). Additionally, every dataset was split into three distinct sets train, validation, and test with augmentations such as random mirroring and cropping applied on the train set.

6.6.1 Experiment Design

We use a 28 layers wide and 10 layers deep WideResNet (Zagoruyko and Komodakis, 2016) as the DNN estimator, which we trained for 300 epochs on each objective using a validation set for hyperparameter tuning and rolling back to the best network to avoid overfitting. The optimiser was Stochastic Gradient Descent (SGD) (Kiefer and Wolfowitz, 1952; Robbins and Monro, 1951) with momentum set to 0.9 and weight decay in the range $[3e^{-4}, 5e^{-4}]$.

To comprehensively explore the performance of the different estimators we perform experiments using each dataset for training while retaining the remaining datasets for measuring the ability of the estimator to perform OOD detection (Malinin and Gales, 2018; Morningstar et al., 2021; Sastry and Oore, 2020). The ability of each estimator to perform the in-distribution image classification task it was trained for, was first evaluated using the corresponding in-distribution test set associated with each dataset. All datasets have balanced category distributions, therefore we use classification accuracy to measure their performance.

To measure the ability of the estimators to recognise OOD instances we follow the same procedure outlined in Chapter 4 obtaining prediction estimates for both the test portion of the in-distribution dataset and also the test portion of the other three remaining out-of-distribution datasets: *CIFAR-100*, *SVHN*, and *LSUN*. This means that for each training set we have three different evaluations of OOD detection effectiveness. For instance, when *SVHN* is used as the in-distribution training set, *CIFAR-10*, *LSUN*, and *CIFAR-100* are used as the out-of-distribution test sets. One of the available out-of-distribution datasets is selected for use during training for the following methods DPN, DNN+ℓ_1, ContReg, and ContRank which require this extra data. These datasets are marked with an asterisk (*) in Table 6.3. Only the training set of this dataset is used for this purpose, while the test set are used for evaluation.

The estimated predictions provided by the underlying estimators are converted into OOD scores using confidence, entropy, and mutual information. To avoid having to set detection thresholds on these scores we measure the separation between the scores generated for instances on the ID and OOD test sets using the area under the curve (AUC-ROC). We do this individually for each approach in order to generate the AUC-ROC scores.

6.6.2 Results

First we consider whether our approach performs better at OOD detection compared to the DNN baseline. According to Table 6.3 our proposed objectives ContReg, and ContRank outperform the DNN baseline by a large margin (see also Figure 6.3), except for when *CIFAR-100* is utilised as the in-distribution data S_{ID} and *CIFAR-10*, and *LSUN* as the out-of-distribution data S_{OOD}. This observation is interesting since it suggests that the value of auxiliary information from S_{OOD} might reduce when S_{ID} is a superset of S_{OOD}. With the term superset we refer to the fact that the ID data S_{ID} might represent a broader set of covariates compared to OOD data S_{OOD}. Thus, training an estimator with a small subset of the ID data as OOD might not be beneficial since no additional information is presented to the estimator because the covariates from S_{ID} and S_{OOD} overlap with each other. In Table 6.2 we present the classification performance of the estimators on the in-distribution test datasets. Observe from the highlighted scores that our methods perform equally well to those attaining the highest scores with only a small reduction in the final accuracy of the in-distribution test datasets.

Table 6.2: Accuracy of estimators on the in-distribution data classification task.

Estimators	CIFAR-10	SVHN	FashionMNIST	CIFAR100
DNN	95.06	96.67	95.27	77.44
DPN	88.10	90.10	93.20	**79.34**
MCD	96.22	96.90	**95.40**	78.39
SWAG	**96.53**	**97.06**	93.80	78.61
JEM	92.83	96.13	83.21	77.86
DNN+ℓ_1	90.66	95.34	93.89	62.30
DNN+ℓ_1+MCD	90.33	94.85	91.37	60.35
ContReg (ours)	90.76	**95.25**	**93.68**	**72.78**
ContReg+MCD (ours)	90.31	94.75	93.01	64.04
ContRank (ours)	89.01	94.97	93.40	64.32
ContRank+MCD (ours)	**91.96**	82.34	93.13	60.43

Table 6.3: Out-of-distribution experiment results. Scores are Entropy based AUC-ROC in percentage. The values in bracket are % improvement of the corresponding algorithmwrt. DNN, taken as a baseline. An ↑ indicates improvement and ↓ degradation wrt. the baseline (DNN). The asterisks (*) next to each dataset indicates out-distribution datasets used to train DPN.

Data		(baseline)	Entropy AUC-ROC score (% gain wrt. baseline)									
S_{ID}	S_{OOD}	DNN	DPN	MCD	SWAG	JEM	DNN + ℓ_1	DNN + ℓ_1 + MCD	ContReg	ContReg + MCD	ContRank	ContRank + MCD
CIFAR-10	CIFAR-100*	86.27	85.60 (↓0.78%)	89.92 (↑4.23%)	91.89 (↑6.51%)	87.35 (↑1.25%)	95.74 (↑10.97%)	95.99 (↑11.26%)	92.74 (↑7.50%)	93.62 (↑8.52%)	94.92 (↑10.03%)	96.60 (↑11.97%)
	SVHN	89.72	98.90 (↑10.23%)	96.25 (↑7.28%)	98.62 (↑9.92%)	89.22 (↓0.56%)	96.72 (↑7.80%)	97.51 (↑8.68%)	99.33 (↑10.71%)	98.36 (↑9.63%)	95.40 (↑6.33%)	96.96 (↑8.07%)
	LSUN	88.83	83.30 (↓6.23%)	92.04 (↑3.61%)	95.12 (↑7.08%)	89.84 (↑1.14%)	95.31 (↑7.29%)	96.45 (↑8.57%)	93.11 (↑4.82%)	93.62 (↑5.39%)	95.21 (↑7.18%)	95.61 (↑7.63%)
SVHN	CIFAR-100	93.19	99.10 (↑6.34%)	94.33 (↑1.22%)	95.97 (↑2.98%)	92.34 (↓0.91%)	99.87 (↑7.17%)	99.85 (↑7.15%)	99.86 (↑7.16%)	99.78 (↑7.07%)	99.93 (↑7.23%)	99.20 (↑6.45%)
	CIFAR-10*	94.58	99.60 (↑5.31%)	94.97 (↑0.41%)	96.03 (↑1.53%)	92.85 (↓1.83%)	99.99 (↑5.72%)	99.91 (↑5.64%)	99.97 (↑5.70%)	99.93 (↑5.66%)	99.98 (↑5.71%)	99.22 (↑4.91%)
	LSUN	92.97	99.70 (↑7.24%)	93.31 (↑0.37%)	95.71 (↑2.95%)	91.82 (↓1.24%)	99.90 (↑7.45%)	99.91 (↑7.46%)	99.91 (↑7.46%)	99.87 (↑7.42%)	99.97 (↑7.53%)	99.26 (↑6.77%)
FashionMNIST	CIFAR-100	91.20	99.50 (↑9.10%)	93.75 (↑2.80%)	96.19 (↑5.47%)	62.79 (↓31.15%)	99.98 (↑9.63%)	99.96 (↑9.61%)	99.92 (↑9.56%)	99.91 (↑9.55%)	99.97 (↑9.62%)	99.95 (↑9.59%)
	CIFAR-10*	94.59	99.60 (↑5.30%)	96.06 (↑1.55%)	94.28 (↓0.33%)	64.76 (↓31.54%)	99.99 (↑5.71%)	99.98 (↑5.70)%	99.97 (↑5.69%)	99.97 (↑5.69%)	99.99 (↑5.71%)	99.96 (↑5.68%)
	LSUN	93.34	99.80 (↑6.92%)	97.40 (↑4.35%)	99.05 (↑6.12%)	65.38 (↓29.96%)	99.99 (↑7.12%)	99.98 (↑7.11)%	99.97 (↑7.10%)	99.96 (↑7.09%)	99.99 (↑7.12%)	99.96 (↑7.09%)
CIFAR-100	CIFAR-10	78.25	85.15 (↑8.82%)	80.70 (↑3.13%)	84.92 (↑8.52%)	77.64 (↓0.78%)	68.74 (↓12.15%)	66.93 (↓14.46%)	74.04 (↓5.38%)	68.25 (↓12.77%)	69.56 (↓11.10%)	69.47 (↓11.22%)
	SVHN*	81.52	92.64 (↑13.64%)	85.59 (↑4.99%)	94.16 (↑15.51%)	81.22 (↓0.37%)	99.98 (↑22.64%)	99.71 (↑22.31%)	99.76 (↑22.37%)	98.98 (↑21.41%)	99.95 (↑22.61%)	99.79 (↑22.41%)
	LSUN	77.22	86.38 (↑11.86%)	76.58 (↓0.83%)	87.22 (↑12.95%)	77.54 (↑0.41%)	63.67 (↓17.54%)	62.31 (↓19.30%)	71.39 (↓7.54%)	64.21 (↓16.84%)	61.47 (↓20.39%)	61.32 (↓20.59)%
Avg % improvement			(↑6.48%)	(↑2.76%)	(↑6.60%)	(↓7.96%)	(↑5.15%)	(↑4.98%)	(↑6.26%)	(↑4.82%)	(↑4.80%)	(↑4.90%)

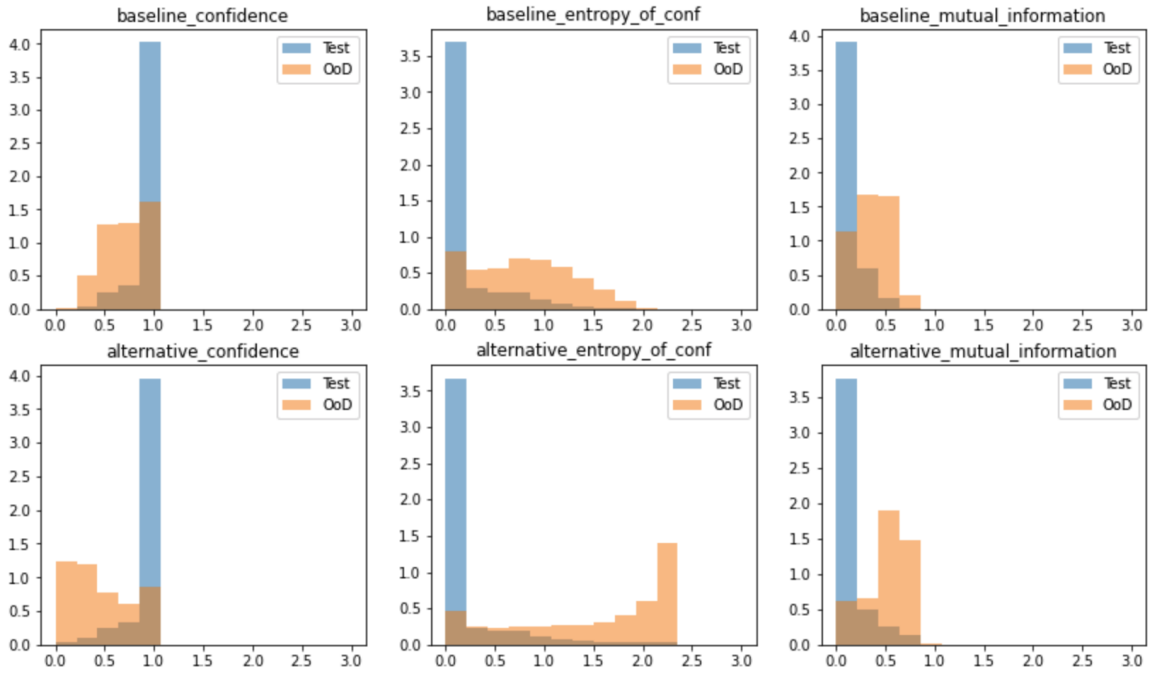


Figure 6.3: Comparison of DNN baseline with cross-entropy against the alternative proposed objective ContReg on three metrics *confidence*, *entropy*, and *mutual information* with respect to a WideResNet28x10 architecture.

Another factor that impairs OOD detection is the presence of label noise (Mitros et al., 2020b), (e.g. *CIFAR-10* vs. *CIFAR-100*), which has been identified with the term *near-OOD* vs. *far-OOD* in subsequent work (Winkens et al., 2020). One can think of *near-OOD* similarly to the case of hard-triplets presented earlier, while *far-OOD* to the case of easy-triplets. A natural question arising from this observation is whether we can identify the inflection point between target noise and OOD detection which would constitute an interesting direction for future work.

The second evaluation we perform investigates how the OOD detection estimators trained using the newly proposed objective functions compare to state-of-art OOD detection methods from the literature. The findings of the comparison are presented in Table 6.4 where we report confidence-based AUC-ROC scores tested on three OOD datasets. Due to computational constraints and the fact that existing published pre-trained estimators from existing methods complementing our experimental approach were only available for the *CIFAR-10* and *CIFAR-100* datasets we only use these in this experiment. The results for baseline approaches are based on those presented in publications describing them (cited below). Table 6.4 shows a comparison of the OOD detection perfromance of our propsed approaches and existing baselines.

Our methods ContReg, and ContRank outperform in the majority of cases *MSP* (Hendrycks and Gimpel, 2017), *ODIN* (Liang et al., 2018), and *EnergyOOD* (Liu et al., 2020) while at the same time provide equivalent results with *Mahalanobis* (Lee et al., 2018a), *MSRep* (Shalev et al., 2018), and *OE* (Hendrycks et al., 2019a). This is interesting since our goal is not to explicitly construct an OOD detector similar to *Mahalanobis*, *MSRep* and *OE*, but rather train a normal estimator on a classification problem and as consequence exhibit robust behaviour on ambiguous inputs.

Table 6.4: Comparison of our proposed methods with related work based on published results in the literature directly corresponding to our trained estimators.

Data		Entropy AUC-ROC scores							
S_{ID}	S_{OOD}	MSP	Mahalanobis	ODIN	MSRep	OE	EnergyOOD	ContReg(ours)	ContRank(ours)
CIFAR-10	CIFAR-100*	86.15	93.90	85.59	91.23	93.30	92.60	92.23	**94.23**
	SVHN	89.60	97.62	91.96	**99.48**	98.40	90.96	**99.18**	95.40
	LSUN	88.54	96.30	90.35	96.05	**97.60**	94.24	92.44	94.77
CIFAR-100	CIFAR-10	73.41	81.34	74.54	**81.49**	75.70	76.61	72.94	68.89
	SVHN*	71.44	86.01	67.26	87.42	86.66	73.99	**99.68**	**99.95**
	LSUN	75.38	**93.90**	78.94	79.05	79.71	79.23	70.50	62.17

Often ambiguous inputs can be categorised as either natural or artificial corruptions. Natural corruptions could refer to artefacts introduced either due to measurement imprecision or due to inconsistent exposure conditions of the underlying object. Instead, artificial corruptions refer to specific corruptions due to an adversary or a process manipulating an outcome. Natural corruptions are prevalent in realistic applications leading to obscure failures of predictive estimators. In that regard it would be appealing to investigate whether our proposed objectives provide any protection against such common corruptions. This is the focus of the next evaluation experiment.

In order to evaluate whether our method is robust against common corruptions we utilised *CIFAR10-C* and *CIFAR100-C* including corruptions such as snow, fog, frost, contrast, etc. The underlying estimator architecture is a WideResNet28x10 trained on in-distribution *CIFAR-10* and tested on corrupted OOD datasets *CIFAR10-C* and *CIFAR100-C*. Similar to (Hendrycks and Dietterich, 2019) we report the mean corruption error (mCE) in Table 6.5.[1] Instead of the usual accuracy and AUC-ROC score metrics we shifted to the mCE score since it is more appropriate in identifying the robustness strengh of an estimator against common corruptions. Observe that our objective ContReg attains the smallest mCE on *CIFAR10-C* while DNN+ℓ_1 achieves the lowest score on *CIFAR100-C* with ContRank second best. Smaller values indicate the underlying estimator is robust against corrupted inputs implying that the proposed

[1]We do not adjust for the varying corruption difficulties by dividing the average corruption error with that of a baseline estimator.

objectives not only can improve OOD detection, but they still manage to handle these common corruptions of in-distribution data.

Table 6.5: Evaluating objectives on common corruptions against CIFAR10-C and CIFAR100-C measured in average corruption error (mCE).

Estimators	mCE	
	CIFAR10-C	CIFAR100-C
DNN	161.14	717.04
DNN+MCD	120.91	536.63
DNN+ℓ_1	144.02	**247.78**
DNN+ℓ_1+MCD	140.96	285.04
ContReg (ours)	**119.98**	337.94
ContReg+MCD (ours)	129.20	269.52
ContRank (ours)	149.46	**258.73**
ContRank+MCD (ours)	167.30	306.42

6.7 Impact of Regularisation

As well as evaluating the OOD detection ability of estimators trained using different objectives, the experiments described in the previous section also provide interesting insights into the role of regularisation in OOD detection. Observe in Table 6.6 that even though explicit regularisation (i.e. Monte-Carlo Dropout (MCD)) overall is beneficial compared to no regularisation, on the contrary, stronger regularisation might instead deteriorate OOD detection.

Table 6.6 is split in two parts with the first part introducing estimators without explicit MCD regularisation in the learning process, and the second part involving the same estimators with explicit MCD regularisation in the learning process. The symbol ✓ indicates whether MCD was also introduced in the inference process as an approximate sampling procedure transforming the underlying estimators to Bayesian neural networks, whereas the symbol ✗ indicates that MCD was not introduced in the inference process interpreting the underlying estimators as point estimate deep neural networks.

Figure 6.4 illustrates the error surface of different estimators trained with and without explicit regularisation (MCD). Each row indicates two error surface diagrams, with the left corresponding to an estimator trained without explicit regularisation and the right corresponding to an estimator trained with explicit regularisation. The y-axis depicts the loss for a randomly selected test sample plotted against the gradient direction of that sample in addition to a random direction. Notice that in Figures 6.4 (b) and (f) explicit regularisation indeed improves the test error by reducing it compared to no

explicit regularisation in Figures 6.4 (a) and (e). Instead, in Figure 6.4 (d) we observe that additional (i.e. stronger) explicit regularisation (MCD) in conjunction with ℓ_1 regularisation can lead to a degradation of the test error compared to Figure 6.4 (c).

The excessive use of explicit MCD regularisation is also evident in Figure 6.4 (c) and Figure 6.4 (d), where integrating ℓ_1 regularisation with MCD leads to a degradation of estimation on OOD data. An ongoing inquiry is to formally characterise and identify the necessary and sufficient conditions of regularisation in order to robustify estimators against ambiguous and corrupted inputs.

Similar conclusions supporting our claims on the impact of explicit regularisation have been also identified in prior work (Khani and Liang, 2020, 2021; Rice et al., 2020; Sagawa et al., 2020; Wei et al., 2020). What is still not evident at this point is why does explicit regularisation improve OOD detection? To answer this question we suggest a potential connection between Dropout (Gal and Ghahramani, 2016), Mixup (Zhang et al., 2017) and Randomised smoothing (Cohen et al., 2019), with these methods acting as a decision boundary thickening approach (Yang et al., 2020).

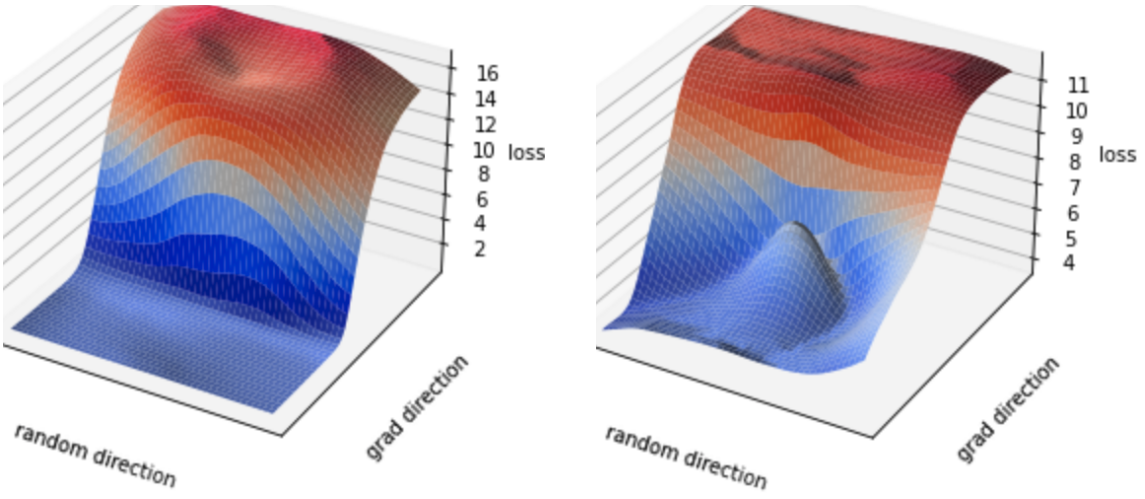
Comparing the decision boundaries in Figure 6.2 (a) and Figure 6.2 (b) we would observe that indeed Dropout regularisation acts equivalently to a decision boundary thickening technique. These observations lead us to the following interesting questions to which we do not yet know the answers:

- What is the role of objective function, regularisation and hypothesis class in OOD detection tasks?
- How to optimally utilise and construct auxiliary information (e.g. OOD training data) in order to mitigate high confidence predictions for ambiguous inputs?

These questions would also constitute potential directions for further investigation in future work.

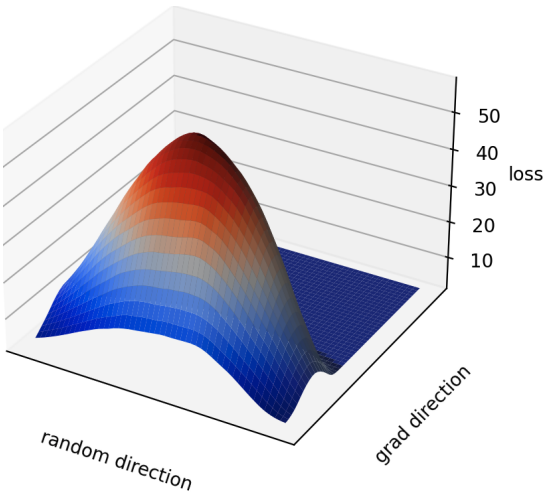
Table 6.6: Comparison of our proposed methods with and without MCD during train and inference ✓.

Data		Entropy AUC-ROC scores in %											
S_{ID}	S_{OOD}	DNN + ℓ_1		ContReg (ours)		ContRank (ours)		DNN + ℓ_1+MCD		ContReg+MCD (ours)		ContRank+MCD (ours)	
		✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓
CIFAR-10	CIFAR-100*	95.74	**96.37**	92.74	92.66	94.92	**94.96**	95.99	**96.11**	93.62	**93.86**	96.60	**96.79**
	SVHN	96.72	**97.35**	99.33	**99.49**	95.40	**95.79**	97.51	97.51	98.36	**98.80**	96.96	**97.18**
	LSUN	95.31	**95.99**	93.11	93.06	95.21	94.79	96.45	**96.51**	93.62	**94.08**	95.61	**95.96**
CIFAR-100	CIFAR-10	68.74	**69.15**	74.04	72.83	69.56	69.28	66.93	**67.97**	68.25	**68.95**	69.47	69.15
	SVHN*	99.98	**99.99**	99.76	99.74	99.95	99.95	99.71	**99.87**	98.98	99.28	99.79	**99.83**
	LSUN	63.67	**65.27**	71.39	71.26	61.47	**62.74**	62.31	**63.56**	64.21	**65.35**	61.32	**62.03**
SVHN	CIFAR-100	99.87	**99.88**	99.86	99.81	99.93	99.89	99.85	**99.86**	99.78	**99.80**	99.20	**99.45**
	CIFAR-10*	99.99	99.93	99.97	99.96	99.98	99.98	99.91	**99.93**	99.93	99.93	99.22	**99.48**
	LSUN	99.90	**99.92**	99.91	99.87	99.97	99.96	99.91	**99.92**	99.87	99.87	99.26	**99.48**
FashionMNIST	CIFAR-100	99.98	99.97	99.92	**99.93**	99.97	99.97	99.96	**99.97**	99.91	99.87	99.95	**99.97**
	CIFAR-10*	99.99	99.99	99.97	99.97	99.99	99.99	99.98	99.98	99.97	99.94	99.96	**99.98**
	LSUN	99.99	99.99	99.97	99.97	99.99	99.90	99.98	99.98	99.96	99.95	99.96	**99.98**

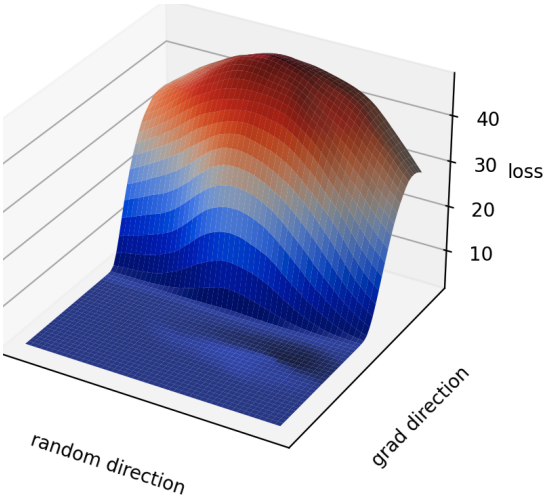


(a) DNN.

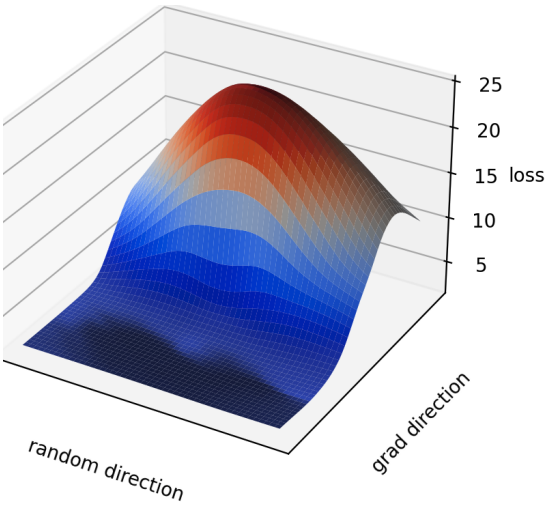
(b) DNN+MCD.



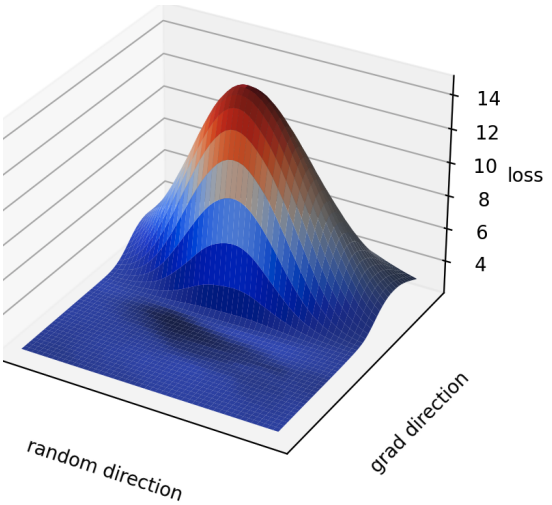
(c) DNN+ℓ_1.



(d) DNN+ℓ_1+MCD.



(e) ContReg.



(f) ContReg+MCD.

Figure 6.4: Comparison of different objectives trained on ID CIFAR-10 and tested on OOD CIFAR-100 with (1st column) and without (2nd column) explicit regularisation (MCD: Monte-Carlo Dropout).

6.8 Conclusion

In this chapter we presented two novel objective functions with the goal of being utilised in a classification setting while at the same time improving OOD detection and exhibiting a level of robustness against common corruptions and ambiguous inputs. We demonstrate that our approach outperforms half of the baseline methods and performs competitively to the remaining ones. In addition, these objectives can ideally represent uncertainty for ambiguous inputs outside the support of the in-distribution train data resembling density estimators, thus avoiding the pitfall of max-margin estimators (e.g. cross-entropy) assigning relatively high confidence equally for near and far from the decision boundary instances. Moreover, they can be utilised for both classification and density estimation whenever there is an absence of target categories in the train data.

Compared to the *Mahalanobis* estimator these objectives do not require estimating expensive covariance matrices for each layer of a DNN in addition to adding adversarial noise to each test instance prior to OOD detection, or require multiple estimators and multiple world embeddings to represent each target category like *MSRep*. Contrary to *OE* the objectives presented in this chapter do not rely on a subset of the ImageNet dataset as auxiliary data in order to represent a broader set of covariates for OOD detection, and potentially can ameliorate the need for unlabelled auxiliary data by substituting it with random augmentations which is an idea we elaborate further in the future work section. Furthermore, we present the efficacy of our method against common corruptions measured in mCE compared to competitive alternative methods.

Finally, we identify the importance of auxiliary information in addition to the role of regularisation in OOD detection, followed by a number of questions in essence asking to identify the role of bias in the choice of objective function L, hypothesis class $\mathcal{F}$, and learning algorithm $\mathcal{A}$ when considering designing robust estimators for out-of-distribution detection.

CHAPTER

SEVEN

CONCLUSION

7.1 Summary

This dissertation described an investigation into the following research questions:

- **RQ1**: *Are Bayesian neural networks better calibrated than point estimate neural networks?*
- **RQ2**: *Can Bayesian neural networks quantify uncertainty of ambiguous inputs better than point estimate neural networks?*
- **RQ3**:*Are Bayesian neural networks capable of detecting OOD inputs, and how do they compare with point estimate neural networks for this task?*
- **RQ4**: *Do Bayesian neural networks exhibit any robustness against adversarial inputs by default, and, if not can we make them more robust?*
- **RQ5**: *Is it possible to devise custom objectives in order to improve OOD detection ability in neural networks utilising auxiliary information as a form of regularisation?*

In this dissertation, we studied the problem of out-of-distribution detection in Bayesian neural networks from the perspective of calibration, uncertainty estimation, and adversarial noise reduction addressing these research questions. We demonstrated that Bayesian neural networks are better calibrated than point estimate deep neural networks while providing a principled approach allowing the underlying estimator to be informed about the uncertainty in their parameters leading to overall better uncertainty estimation over their predictions. Moreover, we showed that Bayesian neural networks outperform point estimate deep neural networks at the task of out-of-distribution detection, but they cannot withstand against common corruptions and adversarial noise without additional underlying assumptions regarding the nature of

out-of-distribution data or possible mitigation approaches to improve their efficacy. Furthermore, we showed that there seem to be no strong evidence of consistent correlation across datasets between calibration of an estimator and its ability to perform out-of-distribution detection. Finally, we proposed two objectives to improve out-distribution detection performance which are independent of any architecture or underlying training methodology, and therefore can be incorporated equivalently both in point estimate deep neural networks and Bayesian neural networks.

The contributions claimed in this dissertation in response to the research questions are summarised below:

We show that Bayesian neural networks by default are better calibrated than point estimate deep neural networks while at the same time providing sensible uncertainty estimation for novel inputs. Furthermore, we show that there seems to be absence of strong evidence of correlation across datasets between calibration and uncertainty estimation ability of the underlying estimator overall implying that a calibrated estimator might not necessarily improve uncertainty estimation on such novel instances. This contribution addresses **RQ1** and **RQ2** and the corresponding claims stated here are supported in Chapter 3.

In addition, we show that Bayesian neural networks are indeed effective at uncertainty estimation and, additionally, can outperform point estimate deep neural networks in out-of-distribution detection. This contribution addresses **RQ3** and its claim is supported by the experiments outlined in Chapter 4 showing that among the estimators evaluated three Bayesian neural networks (DPN, MC-Dropout, SWAG) performed best, surpassing the DNN baseline. It should be noted that DPN which is the best performing Bayesian estimator for out-of-distribution detection is sensitive to the choice of hyperparameters leading to an arduous training process in addition to requiring access to auxiliary training data.

Moreover, we show that adversarial defence techniques improve detection for out-of-distribution inputs in Bayesian neural networks while also withstanding against adversarial noise. This contribution addresses **RQ4** and is supported by the experiments in Chapter 5 showing that adversarial defence techniques like Randsmooth and MMLDA overall provide the best defence against adversarial inputs in terms of accuracy on the corrupted test set with adversarial noise. Instead, for out-of-distribution detection on the adversarially corrupted test set Top-k provides the best results approximately similar among all estimators, and, finally, for out-of-distribution detection on ambiguous inputs not explicitly corrupted with adversarial noise MMLDA and Top-k provide the best results approximately similar for both DNN and MC-Dropout. No-

tice that although adversarial defence techniques overall improve out-of-distribution detection they might degrade accuracy on the in-distribution test dataset.

Finally, we propose two custom objectives that can extend any estimator to improve out-of-distribution detection while utilising auxiliary information as an additional form of regularisation. We demonstrate that estimators trained using these objectives outperform a number of current approaches while maintaining competitive performance with the rest. This contribution addresses **RQ5** and is supported by the experiments illustrated in Chapter 6 showing that the proposed objectives ContReg and ContRank outperform MSP, ODIN, EnergyOOD in out-of-distribution detection while maintaining competitive performance with Mahalanobis, MSRep, and OE.

7.2 Reflections

The field of uncertainty estimation is vast, from abstentions (De Stefano et al., 2000; Geifman and El-Yaniv, 2019), and conformal predictions (Papadopoulos et al., 2007) to ensembles (Choi and Jang, 2018; Lakshminarayanan et al., 2017). Bayesian neural networks provide a simple and principled approach to estimate uncertainty. Unfortunately they pose their own set of challenges. The two main quantities of interest in a Bayesian neural network are the prior and the posterior. The posterior might not be a faithful representation of the observed data and parameters due to approximation errors introduced while obtaining it or due to a misspecified estimator. The prior, which should express our belief of the governing distribution of parameters prior to observing any data, does not accurately reflect how to appropriately incorporate domain knowledge. This is one of the reasons why the isotropic Gaussian distribution is still the default choice of prior in Bayesian neural networks. Nevertheless, the Bayesian literature is rich with alternative priors (horseshoe, Wishart, etc.) but to the best of our knowledge none of the existing proposed priors translates efficiently in tasks with high-dimensional and noisy data (e.g. images) rendering uncertainty estimation challenging in this application setting.

The challenges of uncertainty estimation are exacerbated in out-of-distribution and adversarial settings. Particularly you might notice that the out-of-distribution detection performance of the estimators might vary depending on the underlying metric utilised in retrieving the AUC-ROC scores. This indicates that there might be a correlation between out-of-distribution detection performance and the underlying metrics obscuring the distinction of whether the observed increase in performance was due to the metrics or the actual choice of the estimator.

Moreover, defending against malicious adversaries is extremely challenging due to the large pixel space of high dimensional images introducing a combinatorial number of possible pixel perturbations from which the adversary can choose to pollute the clean image. A surprising finding of the out-of-distribution detection experiment described in Section 5.3 of Chapter 5, is that most adversarial defence techniques improve performance in out-of-distribution detection approximately similarly across estimators but only two (RandSmooth, MMLDA) provide some protection against adversarially corrupted inputs. It would be interesting to understand if the same pattern is evident while varying the underlying estimator architectures since different architectures inherit different biases which translate in the final results.

In addition, we acknowledge that calibration measures can be sensitive with respect to the size of the evaluation dataset, the number of bins and the norm used to compare accuracy to confidence, potentially leading to underestimation or overestimation of the true calibration error (Nixon et al., 2019).

Finally, although the proposed objectives in Chapter 6 provide good out-of-distribution detection performance nevertheless they require access to additional unlabelled data during training whose availability might vary depending on the application setting. This might be considered as a drawback in scenarios without available access to unlabelled data, for which we ameliorate by proposing a simple fix in the future work section.

7.3 Future Work

Even though Bayesian neural networks exhibit desirable properties of calibration and uncertainty estimation there are still interesting open questions that need to be addressed. In particular, we present two directions at the level of priors and at the level of posteriors, interested in further investigating in our future work, in addition to a potential improvement for our proposed objectives removing their reliance on external unlabelled data sources.

Often the choice of priors is a critical part of the Bayesian inference workflow, but recent Bayesian neural networks have fallen back on vague priors such as utilising isotropic Gaussian priors as the de facto standard. An interesting phenomenon observed in this setting of vague priors is that Bayesian neural networks seem to perform an increasing sparsification of their parameters with depth increase, establishing an alternative form of connection between regularisation and Bayesian neural networks (Vladimirova et al., 2019). Additional compelling research directions of establishing

priors for Bayesian neural networks include exploring the difference between prior distributions on parameter space versus function space (Pillai et al., 2007; Terenin and Draper, 2017), or, establishing an approach to learn priors directly from data (Fortuin et al., 2020; Ulyanov et al., 2020).

Expressing a prior belief in the form of a distribution over the estimator's parameters prior to observing any data can be challenging. Instead, one could inform a prior belief based on the observed data and utilise that as their prior in the final construction of the estimator. This idea was proposed by Ulyanov et al. (2020) in the context of inverse problems for generative estimators, but the same concept can be applied also in the Bayesian framework. The idea consists of learning low level statistics from data (e.g. images) prior to any actual learning with a randomly initialised neural network g. Let S be a dataset and S' be a fixed but noisy dataset different from S, then one could train an estimator $g_\theta(S'; S)$ minimising the error between S' and S. Finally, the obtained parameters θ from estimator g can represent our prior for the estimator f when trained in a classification problem on dataset S. This way the parameters θ express our prior belief in the form of a distribution invariant to noise over the estimator f.

In recent work Wenzel et al. (2020) also found that approximate inference in Bayesian neural networks could be surpassed by tuning the temperature of the posterior on a validation set, often referred to as the cold posterior effect. This finding could be related to earlier work of Foong et al. (2019) drawing attention to the poorly understood approximations due to computational tractability upon which Bayesian neural networks rely, indicating that common approximations such as the factorised Gaussian assumption might lead to inaccurate estimates of predictive posterior uncertainty. For instance, posteriors obtained via mean field variational inference (MFVI) (Blei et al., 2017) might be unable to represent uncertainty between different data categories but have no issue representing uncertainty outside those categories.

Unfortunately, the implications of approximate inference in Bayesian neural networks in terms of uncertainty estimation and in comparison with exact inference to identify their discrepancies are still poorly understood. A naive first step in this direction would have been to perform a large scale ablation study comparing and identifying which approximate inference methods perform best on a number of different tasks and to what extent they deviate or converge to the: (i) true Bayes posterior, and (ii) point estimates obtained from SGD. As a matter of fact Wenzel et al. (2020) raised numerous hypothesis in effort of identifying the source of cold posteriors in approximate inference. Among the studied hypothesis was also the comparison between Hamiltonian Monte Carlo (HMC) and Stochastic Gradient Markov chain Monte Carlo (SG-MCMC) concluding that both methods are in agreement with each other as expressed by the KL divergence

on the obtained predictive distributions from each method, although, we would argue that the experiment size was quite small utilising only multilayer perceptron (MLP) and a more comprehensive study including convolutional estimators might overall be beneficial in providing additional insights into the poorly understood approximate inference.

Another potentially interesting direction for exploration in our future work would be the investigation of compatibility and feasibility of the current methodologies presented in this work with regards to the classification problem as they are being transferred to the equivalent regression problem along with its applications. This will provide further insights into which methodologies are robust enough in order to be assigned to either the classification or regression setting without demur.

Finally, we present a simple approach for future work to ameliorate the need of additional unlabelled data for the objectives proposed in Chapter 6 based on augmentations or adversarial noise corruptions. A disadvantage of our suggested objectives is that they rely on additional unlabelled data to improve OOD detection, but this can be challenging in specific applications where there is a scarcity of such data, for instance in medical applications. In order to resolve this issue we propose to replace the additional unlabelled data S_{OOD} by augmentations or adversarial noise to induce an artificial distribution shift. Let $T \sim \mathcal{T}$ represent an operation inducing a distribution shift on a dataset S such that $S_{OOD} = T(S)$. This way we have a created an artificial OOD dataset from our original dataset S without the need of additional unlabelled data. The training process can now proceed as before with S_{OOD} representing distributionally shifted copies of S and not unlabelled real dataset. These are only some directions representing a small subset of interesting challenges in Bayesian neural networks towards which we would like to contribute in the near future.

BIBLIOGRAPHY

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

Alemi, A. A., Fischer, I., and Dillon, V. J. (2018). Uncertainty in the variational information bottleneck. *CoRR*, abs/1807.00906.

Alquier, P. (2020). Approximate bayesian inference. *Journal of Entropy and Information Studies*, 22.

Amaral, S., Allaire, D., and Willcox, K. (2014). A decomposition-based approach to uncertainty analysis of feed-forward multicomponent systems. *International Journal for Numerical Methods in Engineering*, 100.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *CoRR*, abs/1907.02893.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. (2006). Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems, (NeurIPS)*. MIT Press.

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

Carlini, N. and Wagner, D. A. (2017). Adversarial examples are not easily detected: Bypassing ten detection methods. In *Workshop on Artificial Intelligence and Security, (AISec)*. Association for Computing Machinery.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning, (ICML)*, volume 119. PMLR.

Choi, H. and Jang, E. (2018). Generative ensembles for robust anomaly detection. *CoRR*, abs/1810.01392.

Christian, P. R. (2014). Approximate bayesian computation: A survey on recent results. In *Monte Carlo and Quasi-Monte Carlo Methods, (MCQMC)*, volume 163 of *Springer Proc. in Mathematics and Statistics*.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. (2019). Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning, (ICML)*, volume 97. PMLR.

Cullina, D., Bhagoji, A. N., and Mittal, P. (2018). Pac-learning in the presence of adversaries. In *Advances in Neural Information Processing Systems, (NeurIPS)*, volume 31.

Damianou, A. C. and Lawrence, N. D. (2013). Deep gaussian processes. In *International Conference on Artificial Intelligence and Statistics, (AISTATS)*, volume 31. JMLR.

Dawid, A. P. (1982). The well-calibrated bayesian. *Journal of the American Statistical Association*, 77.

De Stefano, C., Sansone, C., and Vento, M. (2000). To reject or not to reject: that is the question-an answer in case of neural classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 30.

Depeweg, S., Hernandez-Lobato, J.-M., Doshi-Velez, F., and Udluft, S. (2018). Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning, (ICML)*, volume 80. PMLR.

Der Kiureghian, A. and Ditlevsen, O. (2009). Aleatory or epistemic? does it matter? *Structural Safety*, 31. Risk Acceptance and Risk Communication.

Foong, A. Y. K., Burt, D. R., Li, Y., and Turner, R. E. (2019). Pathologies of factorised gaussian and mc-dropout posteriors in bayesian neural networks. *CoRR*, abs/1909.00719.

Fortuin, V., Garriga-Alonso, A., Wenzel, F., Ratsch, G., Turner, R. E., van der Wilk, M., and Aitchison, L. (2020). Bayesian neural network priors revisited. In *NeurIPS Workshops*.

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

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning, (ICML)*, volume 48. JMLR.

Geifman, Y. and El-Yaniv, R. (2019). Selectivenet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning, (ICML)*, volume 97. PMLR.

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

Goyal, P., Mahajan, D., Gupta, A., and Misra, I. (2019). Scaling and benchmarking self-supervised visual representation learning. In *IEEE/CVF International Conference on Computer Vision, (ICCV)*.

Grathwohl, W., Wang, K., Jacobsen, J., Duvenaud, D., Norouzi, M., and Swersky, K. (2020). Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations, (ICLR)*. OpenReview.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning, (ICML)*, volume 70. PMLR.

Gupta, C. and Ramdas, A. (2021). Distribution-free calibration guarantees for histogram binning without sample splitting. In *International Conference on Machine Learning*, volume 139. PMLR.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. In *European Conference of Computer Vision, (ECCV)*, volume 9908. Springer.

Henaff, O. (2020). Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning, (ICML)*, volume 119. PMLR.

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

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, (ICLR)*. OpenReview.net.

Hendrycks, D., Mazeika, M., and Dietterich, T. G. (2019a). Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations, (ICLR)*. OpenReview.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. (2019b). Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems, (NeurIPS)*.

Hill, B. (2019). Confidence in belief, weight of evidence and uncertainty reporting. In *International Symposium on Imprecise Probabilities: Theories and Applications, (ISIPTA)*, volume 103. PMLR.

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

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. W. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(1).

Huang, H., van Amersfoort, J., and Gal, Y. (2021). Decomposing representations for deterministic uncertainty estimation. In *Advances in Neural Information Processing Systems Workshops, (NeurIPS)*, volume 34.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, P. D., and Wilson, G. A. (2018). Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence, (UAI)*. AUAI Press.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37.

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, (NeurIPS)*.

Khani, F. and Liang, P. (2020). Feature noise induces loss discrepancy across groups. In *International Conference on Machine Learning, (ICML)*, volume 119. PMLR.

Khani, F. and Liang, P. (2021). Removing spurious features can hurt accuracy and affect groups disproportionately. In *ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery.

Kiefer, J. and Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Kuleshov, V., Fenner, N., and Ermon, S. (2018). Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning, (ICML)*, volume 80. PMLR.

Kuleshov, V. and Liang, P. (2015). Calibrated structured prediction. In *Advances in Neural Information Processing Systems, (NeurIPS)*, volume 28.

Kumar, A., Liang, P., and Ma, T. (2019). Verified uncertainty calibration. In *Advances in Neural Information Processing Systems, (NeurIPS)*.

Kurakin, A., Goodfellow, J. I., and Bengio, S. (2017). Adversarial machine learning at scale. In *International Conference on Learning Representations, (ICLR)*.

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

Lazo, A. V. and Rathie, P. N. (1978). On the entropy of continuous probability distributions. *IEEE Transactions on Information Theory*, 24.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. (2006). A tutorial on energy-based learning. *Predicting Structured Data*, 1.

Lee, H., Lee, H., Na, D., Kim, S., Park, M., Yang, E., and Hwang, S. J. (2020). Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. In *International Conference on Learning Representations, (ICLR)*. OpenReview.

Lee, K., Lee, H., Lee, K., and Shin, J. (2018a). Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations, (ICLR)*. OpenReview.net.

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, (NeurIPS)*.

Li, J., Zhou, P., Xiong, C., and Hoi, S. (2021). Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations, (ICLR)*.

Li, Y. and Vasconcelos, N. (2020). Background data resampling for outlier-aware classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*.

Liang, S., Li, Y., and Srikant, R. (2018). Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations, (ICLR)*. OpenReview.net.

Liu, J., Paisley, J., Kioumourtzoglou, M.-A., and Coull, B. (2019). Accurate uncertainty estimation and decomposition in ensemble learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Liu, S., Garrepalli, R., Dietterich, G. T., Fern, A., and Hendrycks, D. (2018). Open category detection with PAC guarantees. In *International Conference on Machine Learning, (ICML)*, volume 80. PMLR.

Liu, W., Wang, X., Owens, J., and Li, Y. (2020). Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations, (ICLR)*.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems, (NeurIPS)*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Malinin, A. and Gales, J. F. M. (2018). Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems, (NeurIPS)*.

Marin, J., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). Approximate bayesian computational methods. *Statistics and Computing*, 22.

Masana, M., Ruiz, I., Serrat, J., van de Weijer, J., and López, A. M. (2018). Metric learning for novelty and anomaly detection. In *British Machine Vision Conference, (BMVC)*. BMVA Press.

Matthew, D. H. and David, M. B. (2015). Stochastic structured variational inference. In *International Conference on Artificial Intelligence and Statistics, (AISTATS)*, volume 38 of *Proc. of JMLR Workshop and Conference*. JMLR.

Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., and Lucic, M. (2021). Revisiting the calibration of modern neural networks. In *Advances in Neural Information Processing Systems, (NeurIPS)*, volume 34.

Mitros, J. and Mac Namee, B. (2019). On the validity of bayesian neural networks for uncertainty estimation. In *Irish Conference on Artificial Intelligence and Cognitive Science, (AICS)*, volume 2563.

Mitros, J. and Mac Namee, B. (2021). On the importance of regularisation and auxiliary information in ood detection. In *International Conference on Neural Information Processing, (ICONIP)*, volume 1517. CCIS series Springer.

Mitros, J., Pakrashi, A., and Mac Namee, B. (2020a). A comparison of bayesian deep learning for out of distribution detection and uncertainty estimation. In *International Conference of Machine Learning Workshops, (ICML)*, volume 119.

Mitros, J., Pakrashi, A., and MacNamee, B. (2020b). Ramifications of approximate posterior inference for bayesian deep learning in adversarial and out-of-distribution settings. In *European Conference of Computer Vision Workshops, (ECCV)*, volume 12535. Springer International Publishing.

Mohseni, S., Pitale, M., Yadawa, J. B. S., and Wang, Z. (2020). Self-supervised learning for generalizable out-of-distribution detection. In *International Conference on Artificial Intelligence, (AAAI)*.

Morningstar, W., Ham, C., Gallagher, A., Lakshminarayanan, B., Alemi, A., and Dillon, J. (2021). Density of states estimation for out of distribution detection. In *International Conference on Artificial Intelligence and Statistics, (AISTATS)*, volume 130. PMLR.

Murphy, A. H. and Epstein, E. S. (1967). Verification of probabilistic predictions: A brief review. *Journal of Applied Meteorology and Climatology*, 6.

Naeini, M. P., Cooper, G., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *International Conference on Artificial Intelligence, (AAAI)*.

Nalisnick, E. T., Matsukawa, A., Teh, Y. W., Görür, D., and Lakshminarayanan, B. (2019). Do deep generative models know what they don't know? In *International Conference on Learning Representations, (ICLR)*. OpenReview.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer-Verlag.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *Workshop on Deep Learning and Unsupervised Feature Learning (NeurIPS)*.

Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*.

Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *International Conference on Machine Learning, (ICML)*. Association for Computing Machinery.

Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., and Tran, D. (2019). Measuring calibration in deep learning. In *Conference on Computer Vision and Pattern Recognition Workshops, (CVPR)*.

Oord, v. d. A., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv:1807.03748*.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems, (NeurIPS)*, volume 32.

Ozdag, M. (2018). Adversarial attacks and defenses against deep neural networks: A survey. *Procedia Computer Science*, 140.

Paisley, J. W., Blei, D. M., and Jordan, M. I. (2012). Variational bayesian inference with stochastic search. In *International Conference on Machine Learning, (ICML)*, volume 29.

Pang, T., Xu, K., Dong, Y., Du, C., Chen, N., and Zhu, J. (2020). Rethinking softmax cross-entropy loss for adversarial robustness. In *International Conference on Learning Representations, (ICLR)*. OpenReview.

Papadopoulos, H., Vovk, V., and Gammerman, A. (2007). Conformal prediction with neural networks. In *IEEE International Conference on Tools with Artificial Intelligence, (ICTAI)*, volume 2.

Pillai, N. S., Wu, Q., Liang, F., Mukherjee, S., and Wolpert, R. L. (2007). Characterizing the function space for bayesian kernel models. *Journal of Machine Learning Research*, 8.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances In Large Margin Classifiers*. MIT Press.

Polyak, B. and Juditsky, A. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimisation*, 30.

Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. (2019). On variational bounds of mutual information. In *International Conference on Machine Learning, (ICML)*, volume 97. PMLR.

Qiao, F., Zhao, L., and Peng, X. (2020). Learning to learn single domain generalization. In *Conference on Computer Vision and Pattern Recognition, (CVPR)*. Computer Vision Foundation / IEEE.

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. In *Advances in Neural Information Processing Systems, (NeurIPS)*.

Rice, L., Wong, E., and Kolter, Z. (2020). Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning, (ICML)*, volume 119. PMLR.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22.

Ruppert, D. (1988). Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2020). Distributionally robust neural networks. In *International Conference on Learning Representations, (ICLR)*.

Sastry, C. S. and Oore, S. (2020). Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning, (ICML)*, volume 119. PMLR.

Schulam, P. and Saria, S. (2019). Can you trust this prediction? auditing pointwise reliability after learning. In *ternational Conference on Artificial Intelligence and Statistics, (AISTATS)*, volume 89. PMLR.

Shalev, G., Adi, Y., and Keshet, J. (2018). Out-of-distribution detection using multiple semantic label representations. In *Advances in Neural Information Processing Systems, (NeurIPS)*.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27.

Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations, (ICLR)*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. In *Journal of Machine Learning Research*, volume 15.

Sun, X., Yang, Z., Zhang, C., Ling, K.-V., and Peng, G. (2020). Conditional gaussian distribution learning for open set recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*.

Tack, J., Mo, S., Jeong, J., and Shin, J. (2020). Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems, (NeurIPS)*, volume 33. Curran Associates, Inc.

Terenin, A. and Draper, D. (2017). A noninformative prior on a space of distribution functions. *Entropy*, 19.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2019). Robustness may be at odds with accuracy. In *International Conference on Learning Representations, (ICLR)*.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. S. (2020). Deep image prior. *International Journal of Computer Vision*, 128.

Vaicenavicius, J., Widmann, D., Andersson, C. R., Lindsten, F., Roll, J., and Schön, T. B. (2019). Evaluating model calibration in classification. In *International Conference on Artificial Intelligence and Statistics, (AISTATS)*, volume 89. PMLR.

Vladimirova, M., Verbeek, J., Mesejo, P., and Arbel, J. (2019). Understanding priors in bayesian neural networks at the unit level. In *International Conference on Machine Learning, (ICML)*, volume 97. PMLR.

Vyas, A., Jammalamadaka, N., Zhu, X., Das, D., Kaul, B., and Willke, T. L. (2018). Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *European Conference of Computer Vision, (ECCV)*, volume 11212. Springer.

Wald, A. (1945). Statistical decision functions which minimize the maximum risk. *Annals of Mathematics*, 46.

Wang, H. and Yeung, D.-Y. (2016). Towards bayesian deep learning: A framework and some existing methods. *IEEE Transactions on Knowledge and Data Engineering*, 28.

Wei, C., Kakade, S., and Ma, T. (2020). The implicit and explicit regularization effects of dropout. In *International Conference on Machine Learning, (ICML)*, volume 119. PMLR.

Wenzel, F., Roth, K., Veeling, B. S., 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 *International Conference on Machine Learning, (ICML)*, volume 119. 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, (NeurIPS)*.

Winkens, J., Bunel, R., Roy, A. G., Stanforth, R., Natarajan, V., Ledsam, J. R., MacWilliams, P., Kohli, P., Karthikesalingam, A., Kohl, S., Cemgil, A. T., Eslami, S. M. A., and Ronneberger, O. (2020). Contrastive training for improved out-of-distribution detection. *CoRR*, 2007.05566.

Xiao, C., Zhong, P., and Zheng, C. (2020). Enhancing adversarial defense by k-winners-take-all. In *International Conference on Learning Representations, (ICLR)*. OpenReview.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747.

Yang, Y., Khanna, R., Yu, Y., Gholami, A., Keutzer, K., Gonzalez, J. E., Ramchandran, K., and Mahoney, M. W. (2020). Boundary thickness and robustness in learning models. In *Advances in Neural Information Processing Systems, (NeurIPS)*, volume 33. Curran Associates, Inc.

Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. (2015). LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365.

Zadrozny, B. and Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *International Conference on Machine Learning, (ICML)*. Morgan Kaufmann Publishers Inc.

Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery.

Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. In *Proceedings of the British Machine Vision Conference, (BMVC)*. BMVA Press.

Zhang, H., Cissé, M., Dauphin, N. Y., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations, (ICLR)*, volume 15.