NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

BENCHMARKING BAYESIAN DEEP LEARNING METHODS WITH MULTI-SPECTRAL SATELLITE IMAGERY

by

Benjamin R. Marsh

September 2021

Thesis Advisor: Marko Orescanin
Second Reader: Scott Powell

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2021	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE BENCHMARKING BAYESIAN DEEP LEARNING METHODS WITH MULTI-SPECTRAL SATELLITE IMAGERY	**5. FUNDING NUMBERS**
6. AUTHOR(S) Benjamin R. Marsh	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

The deep convolutional neural network (DCNN) is the current state-of-the-art approach for automatic image classification tasks. Historically, Bayesian deep learning methods have been applied to these models in narrow scopes. This thesis has created and tested several Bayesian deep learning models to perform classification on operational meteorological multi-spectral satellite data while quantifying the uncertainty in the model predictions. This large-scale dataset is used to compare the performance of Bayesian models against a DCNN and the current algorithm used by the National Aeronautics and Space Administration (NASA) to perform precipitation classification on the dataset. The use of a large-scale, operational dataset to benchmark Bayesian deep learning methods is the first application of its kind and represents a novel contribution to the fields of Bayesian deep learning and computer science. Several novel benchmarks were developed for use in this work. The best performing Bayesian model achieved 92 percent classification accuracy with demonstrated calibrated uncertainty on test data. All Bayesian models are shown to outperform current state-of-the-art DCNNs and the current operational algorithm. Furthermore, it is demonstrated that Bayesian model uncertainties can be used to screen uncertain predictions, and these uncertainties can be mapped spatially to identify specific regions of data that can be used to further improve the model performance.

14. SUBJECT TERMS artificial intelligence, machine learning, deep learning, image classification, Bayesian deep learning, uncertainty quantification, convolutional neural networks, deep neural networks, atmospheric science, convective type classification, meteorology, passive microwave satellite images, estimating uncertainty, aleatoric uncertainty, epistemic uncertainty, data science, statistics, deep convolutional neural network, DCNN, National Aeronautics and Space Administration, NASA	**15. NUMBER OF PAGES** 87
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

BENCHMARKING BAYESIAN DEEP LEARNING METHODS WITH MULTI-SPECTRAL SATELLITE IMAGERY

Benjamin R. Marsh
Captain, United States Marine Corps
BS, University of Illinois at Urbana-Champaign, 2015

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2021

Approved by: Marko Orescanin
Advisor

Scott Powell
Second Reader

Gurminder Singh
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The deep convolutional neural network (DCNN) is the current state-of-the-art approach for automatic image classification tasks. Historically, Bayesian deep learning methods have been applied to these models in narrow scopes. This thesis has created and tested several Bayesian deep learning models to perform classification on operational meteorological multi-spectral satellite data while quantifying the uncertainty in the model predictions. This large-scale dataset is used to compare the performance of Bayesian models against a DCNN and the current algorithm used by the National Aeronautics and Space Administration (NASA) to perform precipitation classification on the dataset. The use of a large-scale, operational dataset to benchmark Bayesian deep learning methods is the first application of its kind and represents a novel contribution to the fields of Bayesian deep learning and computer science. Several novel benchmarks were developed for use in this work. The best performing Bayesian model achieved 92 percent classification accuracy with demonstrated calibrated uncertainty on test data. All Bayesian models are shown to outperform current state-of-the-art DCNNs and the current operational algorithm. Furthermore, it is demonstrated that Bayesian model uncertainties can be used to screen uncertain predictions, and these uncertainties can be mapped spatially to identify specific regions of data that can be used to further improve the model performance.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AI	Artificial Intelligence
AI/ML	Artificial Intelligence/Machine Learning
AUROC	Area Under the Receiver Operator Characteristic
BCNN	Bayesian Convolutional Neural Network
BDL	Bayesian Deep Learning
BNN	Bayesian Neural Network
CNN	Convolutional Neural Network
COAMPS	Coupled Ocean/Atmosphere Mesoscale Prediction System
CRM	Cloud Resolving Models
DCNN	Deep Convolutional Neural Network
DNN	Deterministic Neural Network
DON	Department of the Navy
DPR	Dual-Frequency Precipitation Radar
ELBO	Evidence Lower Bound
FLOPS	Floating Point Operations
FPR	False Positive Rate
GMI	GPM Microwave Imager
GPCP	Global Precipitation Climatology Project
GPM	Global Precipitation Measurement

GPROF	Goddard Profiling Algorithm
GPU	Graphics Processing Unit
KL	Kullback-Leibler
LHS	Left Hand Side
MC	Monte Carlo
NASA	National Aeronautics and Space Administration
NAVGEM	Navy Global Environment Model
NEPTUNE	Navy Environmental Prediction System Utilizing the NUMA Core
OOD	Out of Distribution
PMW	Passive Microwave
RESNET	Residual Network
RHS	Right Hand Side
SAR	Synthetic Aperture Radar
SGD	Stochastic Gradient Descent
SGVB	Stochastic Gradient Variational Bayes
TBS	Brightness Temperature
TPR	True Positive Rate
VI	Variational Inference

CHAPTER 1:
Introduction

The second decade of the twenty-first century was a period of rapid technological innovation. The proliferation of powerful computers, fast internet speeds and smart devices and an exploding tech industry have combined to create a deluge of information that is unprecedented in its depth and scope. This data stream is too vast to be analyzed by humans alone. Advances in computer science and technology have unlocked the use of artificial intelligence (AI) to assist in this task. The Department of the Navy (DON) has recognized the utility of AI and has made investments in the applications of this technology. One of these applications is using AI, specifically deep learning and neural networks, to digest enormous data sets of satellite images to segment and classify the data for predictive purposes [1]. In the past, DON weather forecasters and intelligence analysts would interpret this data manually and make predictions. In the future, AI driven models could assist forecasters and analysts to provide predictions that are more accurate than a human analyst can produce alone.

To automate the extraction of information from satellite image datasets for classification and predictive purposes, current research has been focused on the use of supervised learning algorithms, such as deep convolutional neural networks (DCNNs). DCNNs have been applied with great success to the automatic classification of images [2]. In this application, images were input into a DCNN that extracted information from the channels in the image to predict what is contained within the image, and then provided on output a decision if content of an image belongs to a specific class (i.e., the image is a picture of a boat). This approach has been used to classify the weather in a given image of the outdoors [3]. This thesis proposes that the same approach can be used to classify satellite remote sensing images for meteorological purposes, such as classifying precipitation events as convective or stratiform, which are two categories of convection that describe typical vertical profiles of upward and downward motion and diabatic heating, processes that are critical to atmospheric phenomena at all spatial and temporal scales.

Bayesian deep learning, a well-developed field within the artificial intelligence/machine

learning (AI/ML) community, provides an alternative to classical deep learning methods by accounting for model uncertainty in the weight space and achieves balancing between model complexity and data fitting. Kendall and Gal [4] first demonstrated that Bayesian deep learning methods improved performance of neural networks while providing uncertainty estimation on predictions for computer vision tasks.

The focus of this work is on quantifying the uncertainty in deep learning models trained on multi-spectral satellite data, and builds upon previous work by Petković and Orescanin [5] by applying multiple Bayesian convolutional neural network (BCNN) architectures to the same task. Specifically, multi-spectral passive microwave (PMW) satellite data will be used as input data to several different types of BCNNs with the aim of providing synthetic convective type observations and associated uncertainties that may be assimilated into atmospheric forecast models. Additionally, the results obtained from the use of these Bayesian neural networks will be used as a benchmark for these various Bayesian Methods and will be analyzed to investigate the differences in output from each model given similar input data. The synthetic product produced, the usage of Bayesian neural networks, and the bench marking of the Bayesian methods used are novel for this application and are an original contribution to the fields of computer science and meteorology.

1.1 Problem Statement

There are two main problems that this thesis will address. First, the application of Bayesian deep learning to convective type prediction has not yet taken place. While Petković and Orescanin [5] applied conventional deep learning to this task with great success, there are several reasons why a Bayesian Deep Learning advancement may be advantageous. First, conventional deep learning models merely output a prediction, whereas a BCNN would not only make a prediction, but would output the model's uncertainty in that prediction. This uncertainty could be further used to determine if the prediction meets a confidence threshold for inclusion in decision making. These previous two reasons were demonstrated in research by Orescanin et al. in [6]. Additionally, modeling the predictive uncertainties allows a model to be trained to minimize those uncertainties and improve predictions based upon the different uncertainty types. Specifically, Kendall and Gal demonstrated that a BCNN can be trained to learn loss attenuation through modeling aleatoric uncertainty, which improved model accuracy by up to 3 percent over a non-Bayesian baseline model [4].

Developing a BCNN that can accurately classify precipitation events while providing uncertainty estimates could benefit the DON in a variety of ways. First, this task will provide information from a single source that previously (without a BCNN) required two instruments to measure. To elaborate, deriving convective type from PMW observations currently requires the use of nadir-pointing radar observations of the precipitation event in addition to PMW sensor data. With a BCNN, this tasks could be accomplished with a single PMW sensor instrument in conjunction with the BCNN. Additionally, demonstrating the ability to quantify the uncertainty in BCNN Satellite image classification predictions in a meteorological context may provide basis for BCNN research in other DON satellite-remote sensing applications. For example, automated convective and stratiform classification of satellite data will help constrain satellite-derived precipitation estimates since the microphysical and kinematic structure of convective and stratiform rainfall differ substantially. Such improved observations may ultimately help improve the representation of various atmospheric processes in DON numerical models such as the Navy Global Environment Model (NAVGEM), the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) and future generations of models such as the Navy Environmental Prediction System Utilizing the NUMA Core (NEPTUNE)

The second problem to be addressed concerns the benchmarking of various Bayesian Methods in regard to their raw performance and the quality of their uncertainty quantifications. In regard to these uncertainties, Kendall and Gal [4] demonstrated that it is possible to extract aleatoric and epistemic uncertainties in neural networks via BCNN architectures. Moreover, it is possible to estimate the aleatoric and epistemic uncertainties of the model. Modeling the aleatoric uncertainty in a BCNN allows the user to quantify the model uncertainty that is due to noise in the observations (i.e., the input data), while modeling the epistemic uncertainty quantifies the uncertainty that is due to the model bias relative to the true model. Epistemic uncertainty can be explained, and reduced, given enough data.

A developing trend in research in this area is to benchmark Bayesian methods in order to better understand the performance trade-offs inherent to each. As of this writing, Filos et al. [7] have benchmarked Bayesian models using clinical scans of patients' retinas to detect the presence of diabetic retinopathy. This thesis will build upon this research by developing novel Bayesian benchmarks using a large-scale, balanced, operational dataset of PMW images. Developing Bayesian benchmarks using this dataset could benefit the DON

by providing a real-world operational benchmarks that can be used to further future research and development in this space.

1.2 Research Objectives

There are two chief research tasks executed in this thesis. The first task is the classification of precipitation events as convective or stratiform from PMW multi-spectral satellite images. The second task is to provide performance benchmarks for the three Bayesian Methods that will be implemented.

Research objectives to be addressed in this thesis are the following:

1. Determine if Bayesian CNNs can accurately classify cloud convective classes more accurately than deterministic CNNs and the operational Goddard Profiling Algorithm (GPROF).
2. Determine which Bayesian CNN model architectures perform well on multispectral satellite data.
3. Investigate how modeling three different types of uncertainty and filtering by said uncertainties affects the classification performance of the Bayesian models.
4. Investigate model performance trade-offs of estimating and filtering by the three uncertainty types.
5. Develop novel Bayesian Model benchmarks for use on large scale datasets and evaluate the Bayesian model performance with them.

In order to meet these objectives, this thesis will propose and develop three BCNN models to classify cloud convective classes (stratiform vs convective) from passive microwave (PMW) satellite imagery. Additionally, the BCNN model performances will be compared against the current algorithm used by NASA for this task (GPROF) and against the performance of a deterministic CNN.

Next, the developed BCNN models will be analyzed for their performance along several standard metrics. Novel to this thesis will be the development of a series of benchmarks that quantify the uncertainty of the model predictions in a spatial context. These benchmarks will determine the spatial uncertainties in a total predictive uncertainty, aleatoric uncertainty, and epistemic uncertainty context. Lastly, the utility of providing aleatoric and epistemic

uncertainty estimates will be studied, and the model performance trade-offs of estimating total uncertainty over estimating both aleatoric and epistemic uncertainty will be explored.

1.3 Organization

This thesis will begin with an overview of Bayesian deep learning, past and current research in the space, and how this thesis will build upon previous work. Additionally, the derivations of the Bayesian methods will be detailed. Next, an in-depth discussion of the experimental methodology will take place. This section will include a discussion of the dataset used, the model architectures implemented, the metrics used and the implementations of the Bayesian model benchmarks. After the methodology section is a section detailing the results of the experiments. Finally, the research conclusions and future work will be detailed.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Background and Previous Work

The field of Bayesian deep learning is a rapidly expanding field that offers valuable insights into the uncertainty of deep neural network predictions. Currently, there are multiple Bayesian deep learning methods developed, and while work was conducted to compare these methods on smaller, experimental datasets [7], [8], the benchmarking of bayesian methods using a large dataset of multi-spectral images has yet to be completed. This section begins with an overview of Bayesian deep learning that details a main advantage over classical deep learning frameworks. Next, a survey of past and current research on Bayesian deep learning will be examined through a research review, and concludes with how this thesis will advance current knowledge.

2.1 Bayesian Deep Learning: Overview

Deep learning has seen a significant resurgence in recent years due to empirical advantages over other machine learning algorithms on many supervised learning tasks. This resurgence is due to advances in hardware accelerators allowing the construction and training of deep neural networks [9] and the ability to perform optimization on large datasets using stochastic gradient descent (SGD) [10]. While deep learning has spurred significant development in intelligent technologies across many different fields, such as medical imaging [11], there is a significant limitation to the method. Specifically, a deep neural network is unable to express the uncertainty inherent in its predictions, which is especially problematic when the model is confronted with data that was not a part of its training set [12]. Consider that when a deep neural network (DNN) is trained to classify images as part of a certain group, it can be trained to maximize the following equation:

$$Max[p] = Max[p(C|\theta, D)] \tag{2.1}$$

The above equation is an example of a likelihood equation which when maximized (through SGD in deep learning) determines the weight parameters θ of the neural network that

maximize the probability of the model outputting the correct image class C from an image in dataset D. Note that in this classical deep learning context, these weight parameters are fixed values, and that the images that comprise dataset D that is used to optimize the network weights are similar to the images that comprise the dataset used for inference. In other words, if images from classes A, B, and C are used for training, then images from classes A, B, and C are assumed to be input into the model when it is used for inference. If this assumption does not hold, however, an interesting scenario results. Since the network has been trained to classify images into one of three groups (A, B, C), it will continue to classify in this manner during inference with high confidence, even if the input image is actually part of a completely different class (class D) than the classes that the network was trained upon! This type of aberrant input into the network, called an out-of-distribution example (OOD), presents a significant problem for classical deep neural networks. While there are methods developed that can somewhat mitigate this issue [13], classical deep learning frameworks cannot inherently identify OOD examples. While this example specifically details the OOD problem in the context of a classification task, classical neural networks utilized in a regression framework suffer the same problem [12].

Bayesian deep learning provides a solution to this OOD problem in addition to providing accurate estimations of model uncertainty. This property is inherent to the structure of a Bayesian Neural Network (BNN). BNNs replace the fixed weight values found in classical deep learning models with weight distributions.

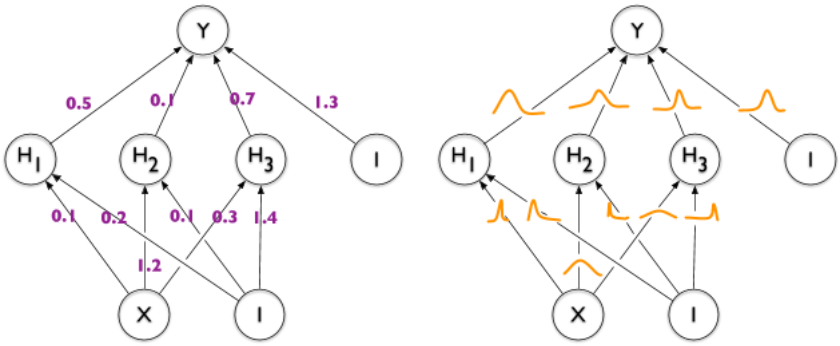


Figure 2.1. A deterministic neural network (DNN) with fixed weights is shown on the left, and a BNN with weight distributions is shown on the right. Source: [14].

Consider the previous classification task example. For the classical deep neural network

with fixed weights, if the same image is processed through the network multiple times, the same classification result will occur each time. Due to this property, classical deep neural networks are DNNs. For a BNN, however, the same input will produce slightly different outputs during each forward pass through the network because in each pass weights are sampled from a distribution.

This non-determinism provides an inherent solution to the OOD problem that plagues DNNs. By making repeated forward passes through the BNN and measuring the variance of the inferences, it is possible to quantity the uncertainty of the model's prediction. This model uncertainty, known as epistemic uncertainty [15], details how uncertain the BNN is in its predictions. OOD examples will often reveal themselves through a high amount of epistemic uncertainty [12].

2.2 Background: Variational Inference

Bayesian deep learning mainly concerns the application of Bayes Theorem [16] to deep neural networks. While Bayes Theorem has been utilized for centuries in the fields of statistics and data science, it was only relatively recently that a Bayesian variational inference algorithm was proposed for use in conjunction with neural networks [17].

Variational inference (VI) [18], also known as Bayesian inference, is a method that is used in probabilistic modeling (of which Bayesian deep learning is a part) to distill normally intractable probabilistic models into optimization problems that offer accurate solutions in practical amounts of user and computational time. Because of this utility, Bayesian variational inference is the foundational method from which other Bayesian deep learning methods are derived.

The derivations in this section were sourced from Broderick's ICML presentation [19] and Blei et al.'s review on Variational Inference [20].

To demonstrate the need for Variational Inference, consider a Standard Bayesian Probabilistic Model:

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)} \tag{2.2}$$

where θ represents the probabilistic model parameters, and Y represents the dataset being used. Subsequently, $p(\theta)$ is a prior distribution that represents what information (in this case, numeric information) is currently known about the model parameters, $p(Y|\theta)$ is a likelihood distribution that describes how the data interacts with the model parameters, and $p(Y)$ is an evidence distribution that describes the information known about the dataset itself.

The overall goal of this standard Bayesian probabilistic model is to learn the interaction between the parameters and the data in order to calculate the mean and co-variance of a posterior distribution $p(\theta|Y)$. If the parameters are known, it is possible to make accurate inferences with new data using the posterior distribution. Calculating the posterior requires calculating the evidence via the following equation where the data is integrated over the parameters:

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{\int p(Y,\theta)d\theta} \tag{2.3}$$

This calculation rapidly becomes intractable when the parameters θ have many dimensions, as the integral to solve for the evidence becomes high-dimensional and, as such, no longer possesses a closed form solution.

Variational inference provides a solution to this problem. Instead of attempting to calculate the true posterior distribution $p(\theta|Y)$ via Eq. 2.3, Variational inference approximates the posterior through optimizing a posterior distribution q^* from a family of distributions Q that possess qualities that are desirable, such as having an easily calculable mean and co-variance. This optimization begins with initializing an approximate q^* from the chosen distribution family Q, and then systematically adjusting the parameters of q^* to minimize the distance between q^* and the true posterior. This process is shown graphically below, where the set of all distributions is described in the gray circle. $p(\theta|Y)$ represents the target true posterior distribution, and q^* is the optimized variational distribution whose distance to the true posterior is the minimum among all of the distributions in the variational family Q.

To continue the process of tuning this variational distribution to fit the true posterior

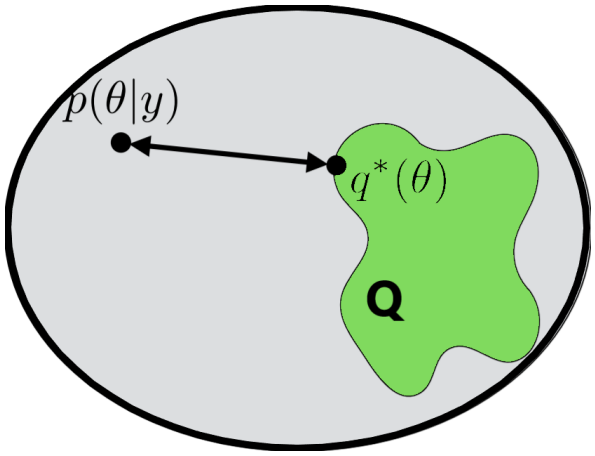


Figure 2.2. Q denotes the variational family, $q^*(\theta)$ is the optimized variational distribution with the minimal distance to the true posterior, $p(\theta|y)$, with the dataset given by y.

distribution can be mathematically modeled as an optimization problem. Consider:

$$q^*(\theta) = \underset{q^* \in Q}{\operatorname{argmin}}\, D[q^*||p(\theta|Y)] \tag{2.4}$$

where a new, optimized posterior approximation $q^*(\theta)$ is calculated by minimizing the distance (D) between the q^* and p distributions. In the equation above, the $argmin$ term represents the choice of q^* from variational family Q that minimizes the distance calculation between the q^* and p distributions. In variational inference, the Kullback-Leibler (KL) divergence [21] is used to calculate the difference between the two distributions. The KL divergence is given by the following formula:

$$KL[q^*||p^*] = \int p(\theta) log(q^*(\theta)/p(\theta)|Y)\, d\theta \tag{2.5}$$

where, identical to above, θ represents the model parameters, $q^*(\theta)$ represents the approximate variational distribution and $p(\theta|Y)$ represents the true posterior distribution. Immediately present is the problem where the true posterior distribution is required to solve for the KL divergence, but by assuming that the prior and posterior distributions are a part of the same probability distribution family (conjugate distributions), the KL divergence

equation can be written as such:

$$KL[q^*||p^*] = \int p(\theta) log(q^*(\theta)p(Y)/p(\theta), Y)\, d\theta \tag{2.6}$$

where the true posterior is no longer required for calculation. In this equation, the assumed conjugate posterior distribution is given by p^* By taking advantage of the properties of logarithms and rearranging terms, the KL divergence equation becomes:

$$KL[q^*||p^*] = log(p^*(Y)) - \int q^*(\theta)(log(p^*(\theta, Y)/q^*(\theta))\, d\theta \tag{2.7}$$

Since the KL divergence will always be greater than, or equal to, zero [22], the log of the evidence will always be greater than the value of the integral. Due to this relationship, the terms can be rearranged further:

$$ELBO = log(p^*(Y)) - KL[q^*||p^*] \tag{2.8}$$

This resulting equation is called the evidence lower bound, or ELBO. The ELBO provides two crucial benefits to the variational inference optimization problem. First, by eliminating the need to calculate the true posterior in the standard KL divergence equation, the ELBO makes stochastic optimization possible. Second, maximizing the ELBO of the variational distribution q^* is mathematically equivalent to minimizing the KL divergence between the variational and conjugate posterior distributions. Therefore, the optimization problem of tuning the variational distribution to the true posterior distribution becomes:

$$q^* = \underset{q^* \in Q}{\text{argmax}}\, ELBO[q^*] \tag{2.9}$$

where the optimized approximate posterior distribution is the distribution whose parameters maximize the ELBO. This procedure can be utilized in Bayesian deep learning through the application of the stochastic variational inference algorithm [23] so that Bayesian model weight distributions can be optimized through stochastic gradient descent to obtain a re-

sulting total approximate posterior that can produce accurate inferences and uncertainty quantification.

2.3 Previous Work

While a formal study of variational inference and BNNs as a whole began in the 1990s, research in the broader applications of BNNs didn't begin in earnest until the 2010s with research into probabilistic frameworks for historically deterministic models [24]. Notably, research in BNNs first coincided with the previously mentioned applications of SGD algorithms and the GPU accelerator breakthroughs. Advances continued in the realm of more computationally efficient variational inference algorithms with the publishing of two papers detailing new ways of implementing variational inference. The first paper detailed an algorithm called bayes by backprop [25], which details a specific variational inference algorithm that lends itself well to SGD-based networks. Next, a further efficiency in variational inference was developed, called reparameterization [26] that computes a more efficient gradient.

Next, Y. Gal and Z. Ghahramani [27] detailed an alternative way to approximate variational inference in BNNs using existing deep learning methods. This method, called Monte Carlo dropout [27], uses standard dropout layers in a novel way to approximate variational inference with less computational cost than a full variational inference implementation. Additional advancement in this area was made with the implementation of the flipout method, which, at the cost of additional computational complexity, computes a more robust gradient estimation than reparametrization [28].

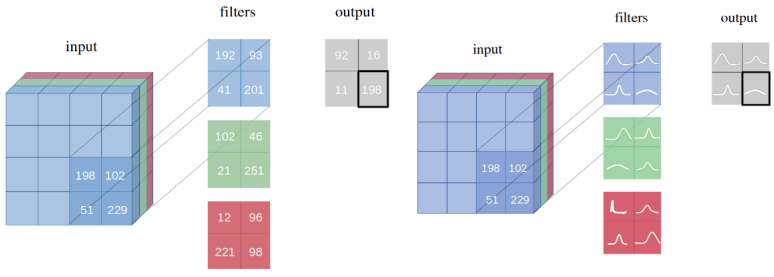
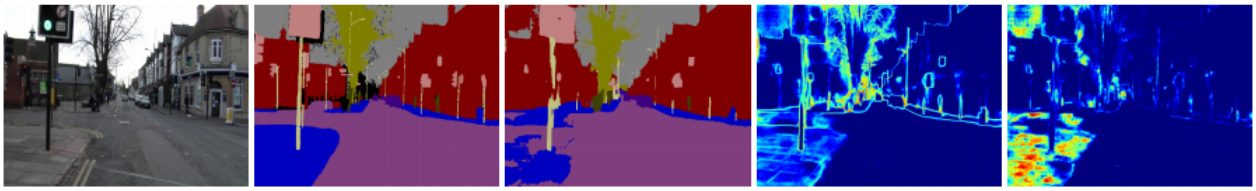


Figure 2.3. A comparison of a DCNN architecture on the left and the BCNN architecture on the right. DCNN convolutional kernels are fixed weights, whereas BCNN convolutional kernels are weight distributions. Source: [29].

The next advancement was the introduction of BCNNs [30]. Similar to the structure differences of DNNs and BNNs, a Bayesian convolutional neural network replaces the fixed-weight kernels found in deterministic convolutional neural networks with weight distributions. The algorithms used for BNNs are equally applicable for use in BCNNs.

This advancement allowed the application of Bayesian deep learning to computer vision and image processing tasks, allowing researchers to examine how Bayesian models express uncertainty in spatial terms across an image. Next, A. Kendall and Y. Gal demonstrated how the predictive uncertainty expressed by BCNNs can be further broken down into different types of uncertainty [4]. This paper described how the bulk predictive uncertainty can be expressed in terms of epistemic uncertainty, which describes uncertainty within the weights of the model, and aleatoric uncertainty, which describes uncertainty inherent within the dataset itself.



(a) Input Image (b) Ground Truth (c) Semantic Segmentation (d) Aleatoric Uncertainty (e) Epistemic Uncertainty

Figure 2.4. A comparison of epistemic and aleatoric uncertainties. An input image is shown on the far left, with the ground truth segmentation and model segmentation shown. Next, the aleatoric, or dataset, uncertainty is shown. On the far right, the epistemic, or model uncertainty, is shown. In this case, the model error in segmentation was due to high model uncertainty in its segmentation. Source: [4].

This paper showed that a Bayesian model can account for uncertainties present in the dataset and for the model's own bias. This additional granularity in uncertainty quantification has greatly increased the utility of BCNNs for computer vision tasks, as it is now possible to show spatial areas where the model is uncertain, and spatial areas where the dataset is uncertain, which can provide insight for improvements to data collection and sensor suites. Recent work in this area offered improvements in the implementation and extraction of aleatoric and epistemic uncertainties in image classification tasks [31] while providing a real-world application in identifying stroke lesions in medical images.

Current research in Bayesian deep learning can be broadly classified into two groups of research. One group focused upon applying Bayesian deep learning to novel datasets, and the other on improving uncertainty quantification and developing Bayesian deep learning benchmarks. Recent research in the first group includes many novel applications, showing the wide applicability of Bayesian deep learning to both computer vision and conventional tasks. For example, Bayesian neural networks have been applied to the problem of detecting cosmic background radiation [32], multi-task scene segmentation with aleatoric uncertainty [33], and seismic facies detection [34]. Recent research on improving uncertainty quantification includes recent work on using aleatoric uncertainty as an analogue for model re-calibration [35], and the effects of including a limited number of Bayesian layers in a standard deterministic model [36], along with the aforementioned stroke lesion use-case.

The rise in the number of applications of Bayesian deep learning (BDL) has driven a need for a suite of performance benchmarks that can accurately compare Bayesian and deterministic models using the same dataset. This benchmarking is especially important in the case of computer vision and other high-risk Bayesian deep learning applications, as each Bayesian implementation may perform and express its uncertainties in a different manner than other implementations. An example of current research in this area is a systematic comparison of various Bayesian and deterministic models using a medical-imaging application [7]. This example uses 512x512 RGB images of retinas as input to a suite of models, and then assesses the models' performance based upon metrics drawn from the medical domain.

Other metrics produced include graphical plots showing the area under the receiver operator characteristic (aucROC) curve and a graphical comparison of predictive uncertainty and predictive probability of classification of disease. Notably, this benchmark does not include spatial plots of the model uncertainties.

In contrast to the applications and research discussed previously, there has been little work done until recently to apply Bayesian deep learning to meteorological remote sensor data. Bayesian probability theory has been investigated for use in several meteorological applications, such as network probability models for weather prediction [37]. Notable examples include a 2018 paper using Bayesian deep learning to improve climate model predictions of severe weather events [38] and an investigation into integrating Bayesian deep learning into the Lorenz 84 weather forecasting system [39]. As previously discussed,

applying Bayesian deep learning to operational datasets, such as Passive Microwave (PMW) observations used for convective type predictions, shows promise due to recent successes in applying deterministic architectures [5].

Applications in remote sensing include active BDL learning tasks using synthetic aperture radar (SAR) data [40] and using hyperspectral data [41] to train classifiers using small quantities of labeled data while conducting reinforcement learning on new, unlabeled data.

In conclusion, Bayesian deep learning is an emerging field within the AI/ML community that has experienced numerous recent successes in improving the performance and reliability of neural network model predictions. Additionally, recent work in the meteorological space shows that Bayesian deep learning has the potential to advance current research of applying deterministic deep learning to meteorological applications, which would further DON research goals in this space while providing valuable test benchmarks for future research.

CHAPTER 3:
Methodology

This chapter explains the experimental methodology of this work. The main goals of this methodology are to benchmark three different implementation methods for Bayesian deep learning using a large-scale, operational data set. These Bayesian models will be used to predict convective-type, and to compare the Bayesian models' predictive performance against a deterministic convolutional neural network and the operational algorithm currently used by the National Aeronautics and Space Administration (NASA) for convective-type prediction. The Bayesian methods to be benchmarked are: reparameterization variational inference, flipout variational inference, and Monte Carlo dropout.

This chapter will comprehensively examine the experimental methodology, beginning with a full explanation of the data set that is being used. Next, the data pipeline code implementation will be discussed, followed by mathematical derivations of all three Bayesian methods. After these derivations, the model architectures used will be examined. After this, derivations of the three types of uncertainty that will be produced from the output of the Bayesian models will be shown, along with a discussion on the regularization scheme of the KL divergence hyperparameter that is used.

To continue, a full examination of the model training methodology will be conducted, and relevant hyperparameters will be explained and discussed. Next, the prediction and serving methodologies will be explained. Finally, a detailed examination and explanation of the performance benchmarks used to evaluate the methodology goals will be given, to include sample metrics and figures when applicable.

3.1 Data Set Description

The following section (3.1) is adapted from [6], previously published by the IEEE Geoscience and Remote Sensing Letters, ©2021 IEEE.[1,2]

3.1.1 Data Set Motivation and Operational Use

The dataset developed for use in this work is derived from PMW and dual-polarization radar data observed by the Global Precipitation Mission.

The Global Precipitation Measurement (GPM) mission is a joint international effort to provide accurate and timely global satellite observations of precipitation [6]. Relying on its core satellite to serve as a calibration standard, the GPM mission uses a constellation of passive microwave radiometers to offer a nearly global sampling of rain and snow rate estimates at 0.1° grid spacing and 30-min temporal resolution. The GPM core-observatory carries a passive microwave imager (GMI; [42]) and an advanced dual-frequency precipitation radar (DPR) system [43], which together collect measurements throughout the atmospheric column and build links between PMW brightness temperatures and radar-derived precipitation rates. Once generated, this link is employed by an enterprise retrieval ([44]) to serve each of the constellation's passive microwave radiometers (more affordable instruments relative to radar), to estimate surface precipitation over large areas.

This unique setup provides accurate global estimates of precipitation rates over long time periods. However, a lack of accurate information on the variability in precipitation system morphology (i.e., convective and stratiform rainfall) often results in region-specific biases

[1]Reprinted, with permission, from Orescanin et al., "Bayesian deep learning for Passive Microwave Precipitation Type Detection," IEEE Geoscience and Remote Sensing Letters, 2021. This publication is a work of the U.S. government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States. IEEE will claim and protect its copyright in international jurisdictions where permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

[2]In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of the Naval Postgraduate School's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

([5]) in precipitation rate. Convective rainfall is usually associated with stronger vertical motions and heavier rainfall than stratiform precipitation ([45]). Using space-borne radar, the vertical structure and horizontal distribution of radar reflectivity factor is leveraged to adequately classify vertical columns of DPR echo ([46]).

However, successfully classifying precipitating regions using PMW radiances is challenging, particularly because highly resolved data in the vertical is not available to detect radar brightbands that are prevalent in stratiform precipitation. Nonetheless, recent deep learning studies have demonstrated that PMW signals indeed contain sufficient information content on precipitation system morphology ([47], [5]), but estimating the uncertainty of those estimates has remained a challenge.

3.1.2 Data Collection and Labeling

The performance of deep learning methods, mainly accuracy and ability to generalize to new inputs, is driven by the quality and quantity of the dataset. The approach of Petkovic et al. [5] is followed with some adjustments. First, two independent 12-month periods of GMI and DPR co-located observations were created. Model training and validation relied on data collected during 2017, while performance tests of the trained model were performed on 2018 observations. The GMI brightness temperatures (TBS) observed at 13 microwave channels (10.65H/V, 18.7H/V, 23.8V, 36.5H/V, 89.0H/V, 166V/H, and 183.3±3/7V GHz) stored in publicly available (e.g., https://storm.pps.eosdis.nasa.gov) GPM level-1 standard product [GPM_BASEGPMGMI_XCAL - V05; GPM Science Team 2016] were used to construct model training features.

This thesis chooses the GMI product to define the atmospheric state (i.e., an observation vector) over an area of approximately 125 km × 125 km centered on the observing Field of View (FOV, hereinafter referred to as pixel). Given the GMI's scanning geometry [42], such an area corresponds to a patch of 25×9 individual pixels. Collecting TBS at each of 13 GMI channels, the resulting training feature elements are stored into 9×25×13 arrays, where the three dimensions reflect the number of GMI scans, pixels and channels, respectively, to form an input data set for the model. The input dataset was normalized by scaling each TBS with its channel's maximum value, subtracting the channel's mean and dividing by the channel's standard deviation, a procedure known as a z-score scaling [48].

Data was labeled using the output from the DPR radar, specifically the GPM_2ADPR standard product and its precipitation rate and type-flag. Two precipitation categories, convective and stratiform, were considered when defining the label for each individual GMI pixel. Using DPR observations falling within GMI's 18 GHz channel field of view, a convective fraction is calculated by applying Gaussian weighting to DPR-observed precipitation rates. This has ensured accurate matching between DPR- and GMI-viewing geometry ([5]). Once available, the convective fraction of precipitation was used to assign a label to each individual GMI pixel. A convective flag was assigned to all pixels with the fraction of 50% or more; otherwise, the pixel was labeled as stratiform. Observations containing any missing or non-classified data (less than 5% of total data) were excluded from the training dataset to ensure minimal noise. Upon labeling, the dataset used for model development was balanced so that an equal representation of both precipitation classes is preserved. In total, ~14 million samples were collected and were further split into training/validation/test data with an 80/10/10 ratio respectively.

3.2 Bayesian Methods

This section derives and details the specific Bayesian deep learning methods used. Each subsection will focus on a specific method with mathematical details and implementation considerations in Tensorflow 2.0. The three methods evaluated in this work: reparameterization, flipout, and Monte Carlo dropout, represent the three most commonly utilized methods in Bayesian deep learning. Further, we discuss trade-offs between different methods. In this thesis we benchmark the quality of the uncertainties generated by these various methods. By comparing the outputs of each method given the same robust dataset, it will be possible to effectively compare and contrast the performance of each of these methods as well as the quality of their uncertainties.

3.2.1 Bayesian Methods: Reparameterization

While the variational inference optimization described in the previous section describes a general optimization problem for variational inference, utilizing this general form in a stochastic gradient descent context will produce a gradient exhibiting very high variance that renders it impractical for use with large data sets [49]. One method for dealing with

this high variance is to reparameterize the ELBO in order to yield a gradient with lower variance.

The derivations in this section were sourced from Kingma and Welling's paper on Auto-Encoding Variational Bayes [50].

Recall the ELBO equation from the previous section on variational inference:

$$ELBO = log(p^*(Y)) - KL[q^*||p^*] \tag{3.1}$$

where the ELBO is defined as the log evidence of the model given the dataset Y minus the Kullback-Liebler Divergence (KL) of the approximate posterior p^* and prior q^*. To demonstrate the motivation for reparameterization, consider a true posterior distribution in a deep learning context, $q_\theta(z|Y)$, whose model parameters θ and latent variables z given the dataset Y are unknown. Similar to the previous section, the desired approximate posterior in this context is given as $q_\phi(z|Y)$. From these two pieces, the ELBO can be re-written as:

$$ELBO(\theta, \phi, Y^{(i)}) = -KL(q_\theta(z)||q_\phi(z|Y^{(i)})) + E_{q_\phi(z|Y^{(i)})}[log(p_\theta(Y^{(i)}|z)] \tag{3.2}$$

where the left hand side (LHS) of the equation denotes the ELBO calculated for a given data point i in dataset Y, model parameters θ and variational parameters ϕ. The right hand side (RHS) of the equation gives the KL divergence between the variational posterior $q_\phi(z|Y$ with variational parameters ϕ and the true posterior $p_\theta(z)$ with model parameters θ, and the expectation of the true posterior with respect to the approximate posterior, all for a given data point i in the data set Y.

In order to optimize the lower bound with respect to both the variational parameters ϕ and model parameters θ, the previous equation must be differentiated with respect to both of these variables. However, the standard Monte Carlo gradient estimator of the variational lower bound with respect to the variational parameters ϕ, exhibits a level of variance that is too high to be practical for deep learning purposes [49].The solution to this issue is to reparameterize $q_\phi(z|Y)$ so that the samples generated from the reparameterized approximate posterior yield a lower variance.

To do this, let z be a continuous random variable, and have z $q_\phi(z|Y)$ be a conditional distribution. Under certain conditions [50], it is possible to reparameterize z using a differentiable transformation $g_\phi(\epsilon|Y)$ of a noise variable $\epsilon \approx p(\epsilon)$. Once this reparameterization is made, Monte Carlo estimates of a function $f(z)$ with respect to the approximate posterior is possible via:

$$E_{q\phi}(z|Y^{(i)})[f(z)] = E_p(\epsilon)[f(g_\phi(\epsilon, Y^{(i)})] \frac{1}{L}\sum_{l=1}^{L} f(g_\phi(\epsilon^{(l)}, Y^{(i)})) \tag{3.3}$$

where $\epsilon^{(l)} \approx p(\epsilon)$. Applying this to the ELBO:

$$ELBO(\theta, \phi; Y^{(i)}) = \frac{1}{L}\sum_{l=1}^{L} log(p_\theta(Y^{(i)}, z^{(i,l)})) - log(q_\phi(z^{(i,l)}|Y^{(i)})) \tag{3.4}$$

where $z^{(i,l)} = g_\phi(\epsilon^{(i,l)}, Y^{(i)})$ and $\epsilon^{(l)} \approx p(\epsilon)$. The above equation is a generic stochastic gradient variational Bayes (SGVB) estimator for the ELBO. Furthermore, given a dataset Y with N datapoints, the generic SGVB estimator can be manipulated to estimate the lower bound on the entire dataset given a minibatch of size M:

$$ELBO^{Minibatch}(\theta, \phi; Y) = \frac{N}{M}\sum_{i=1}^{M} L(\theta, \phi; y^{(i)}) \tag{3.5}$$

yielding a SGVB minibatch estimator. Taking derivatives of this estimator yields gradients that can be used in conjunction with stochastic gradient descent in a deep learning context given in the following reparameterization SGVB Algorithm:

SGVB Minibatch Estimator $ELBO^{Minibatch}(\theta, \phi; Y) = \frac{N}{M}\sum_{i=1}^{M} L(\theta, \phi; y^{(i)})$;

Step 1: Initialize parameters θ and ϕ
Repeat until convergence of parameters θ and ϕ:
Draw random minibatch M from full dataset Y
Draw random samples from noise distribution ϵ
Compute gradients of SGVB minibatch estimator

Update parameters θ and ϕ using gradients of SGVB minibatch estimator
Return parameters θ and ϕ

In Tensorflow 2.0, this method is implemented using the reparameterization layers provided by the Tensorflow Probability library. These layers implement the reparameterization analogue to dense and/or convolution layers by assuming the kernel and/or the bias are drawn from distributions which can be specified by the user [51]. By default, these distributions are assumed to be mean field normal distributions, and the layers implement a stochastic forward pass via sampling from the kernel and bias posteriors [51].

3.2.2 Bayesian Methods: Flipout

The next method utilized in this work is the flipout method. While the previous reparameterization method presents an improvement over standard variational inference, the flipout method, given several assumptions and a trade-off in computational complexity, can yield a gradient of the ELBO that, theoretically, exhibits less variance than the gradient computed via reparameterization. The flipout method takes advantage of the fact that computations on a mini-batch can be computed via matrix multiplications, which can be implemented efficiently with GPUs.

The derivations in this section are sourced from Yeming et al. [28].

To illustrate the flipout method in comparison to reparameterization, consider drawing a sample from a Gaussian weight distribution W. Let $f(X, W)$ represent the output of a Bayesian neural network with weight distribution W and input X. A sample from a Gaussian weight distribution can be described in terms of perturbations:

$$W \approx (\bar{W}, \sigma^2) \tag{3.6}$$

where W represents the weight sample, $\bar{W}$ represents the average of the weight distribution, and σ^2 represents the standard deviation of the distribution. Using reparameterization [50], the above Gaussian perturbation can be rewritten as:

$$W = \bar{W} + \sigma\epsilon \tag{3.7}$$

where $\epsilon \approx (0, 1)$ [50]. Notice from the previous section, however, that the reparameterization SGVB algorithm only samples from the noise distribution ϵ once per minibatch. This is due to the fact that it is very expensive to compute and store a separate weight perturbations for each example in the minibatch. The drawback to this shortcut, however, is that the reparameterization SGVB algorithm gradient estimates suffer from increased variance over a fully independent weight sample perturbation method because all of the training examples in the minibatch share the same perturbation, which introduces correlations between the gradients in the minibatch that cannot be eliminated by averaging [28].

The flipout method reduces this variance by introducing a computationally efficient method of perturbing the weights quasi-independently within a minibatch. To do this, two assumptions must be made: First, that all weight perturbations are independent of the others, and second, that the distribution of perturbations is symmetric around zero. These assumptions are nontrivial, but, importantly, they are allowable for the families of distributions utilized in Bayesian neural networks (notably the Gaussian family of distributions). Notably, under these assumptions, the distribution of perturbations is invariant to element-wise multiplication by a random sign matrix (matrix whose elements are +/- 1) [28].

To demonstrate, let q^* be an approximate prior distribution that follows the above assumptions, let $\Delta\bar{W} \approx q^*$, and let E be a random sign matrix that is independent of $\Delta\bar{W}$. Let $\square$ denote element-wise matrix multiplication.

From the previous paragraphs, we know that Δ W = $\Delta\bar{W}\square E$ is identically distributed to $\Delta\bar{W}$, and that the gradients computed with ΔW are identically distributed to those computed using $\Delta\bar{W}$.

The flipout method utilizes this relationship by using a base weight perturbation $\Delta\bar{W}$ that is shared by every example in the minibatch and multiplied by a unique sign matrix per minibatch example:

$$\Delta W = \Delta\bar{W}\square(r_n s_n)^T \tag{3.8}$$

where n denotes a minibatch example, and r_n and s_n are vectors whose entries are sampled from +/-1. From previously, we know that the marginal distribution over the gradients computed from the above equation will be identical to those computed using a shared weight perturbation (i.e., reparameterization method). Therefore, the flipout method yields unbiased, de-correlated gradients that allow a Gradient Descent algorithm to achieve much lower variance updates when averaging over a minibatch than the reparameterization method [28].

Computationally, this efficiency comes at a cost. Flipout requires twice as many floating point operations (FLOPS) per update as reparameterization [26], [28], but for applications where a large minibatch size can be used, flipout provides a significant reduction in the variance of the computed gradients of the Bayesian neural network [28].

In Tensorflow 2.0, this method is implemented using the flipout layers provided by the Tensorflow Probability library. These layers implement the flipout analogue to dense and/or convolution layers by assuming the kernel and/or the bias are drawn from distributions which can be specified by the user [52]. Identically to the reparameterization layers described previously, these distributions are assumed to be mean field normal distributions, and the layers implement a stochastic forward pass via sampling from the kernel and bias posteriors [52].

3.2.3 Bayesian Methods: Monte Carlo Dropout

The final method to be discussed, Monte Carlo dropout, is a significant shift in implementation from the previous two methods. When implemented in Tensorflow, the reparameterization and flipout methods, by default, approximate each weight with a Gaussian posterior distribution. In effect, this implementation doubles the weights of a Bayesian neural network over its exact deterministic counterpart since each weight in the Bayesian neural network possess both a mean (μ) and a standard deviation term (σ) versus a point value given in a deterministic neural network. While the effectiveness of the previous methods in fitting a BNN was established in the previous sections, the large number of weights present in these types of BNNs can be problematic when applied in situations with computational constraints. Fortunately, Monte Carlo dropout is able to provide an approximation of variational inference at a much lower computational cost.

Dropout, as a concept, was first introduced in 2014 as a way to prevent over-fitting in deep neural networks [53]. Dropout is a layer that, when added to a neural network, sets a random number of neurons in the previous layer to zero during the training of the neural network. These neurons, along with the connection weights that begin at each selected neuron, are effectively "dropped" from the neural network during that specific update cycle of the network. The resultant effect of this dropout is to generalize the neural network in order to prevent over-fitting to a dataset [53]. Originally proven to be effective when used in conjunction with fully-connected layers, dropout has since been proven to generalize convolutional layers as well by dropping feature maps instead of activations [54].

Dropout is implemented in Tensorflow as a layer that is added after a given weight layer in order to implement dropout in the previous layer. The dropout percentage (the percentage of randomly selected weights to drop from the previous layer per update cycle) is given as a parameter that is input by the user. The higher the dropout percentage, the greater the generalization effect of the dropout layer [53].

In order to extend dropout to become a Bayesian deep learning method, the user fixes the dropout layer to be active in testing, as well as training. By extending the dropout layers in this fashion, a deterministic DNN can approximate a BNN without doubling the weights of the neural network [27].

To continue, Monte Carlo dropout approximates weight distributions with a simpler distribution than other Bayesian methods. While other Bayesian methods (such as reparameterization and flipout) utilize a Gaussian weight distribution, Monte Carlo dropout weight distributions are modeled as a distribution tuned by the dropout percentage parameter:

In Bayesian neural networks, the predictive distribution of the network is given by:

$$p(\theta|x, Y) = \sum_i p(\theta|x, w_i) * p(w_i|Y) \tag{3.9}$$

from [27]. Where $p(\theta|x, Y)$ is the total predictive distribution for a given test value, x from dataset Y, and it is calculated by averaging over all the weight distributions in the BNN given in the right hand side of the equation [12].

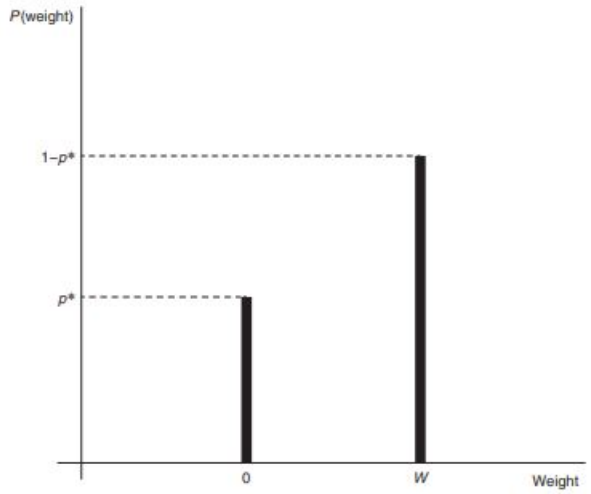


Figure 3.1. A weight distribution using Monte Carlo dropout. Dropout probability for a given weight p and weight value w define the weight distribution. The only values drawn from this distribution are w and 0. Source: [12].

When using Monte Carlo dropout, the dropout layers are active during testing. By making multiple predictions per each test input, the input samples a different sample of active weights in the network. By averaging these predictions, a Monte Carlo predictive distribution is created:

$$p(\theta|x, Y) = \frac{1}{T}\sum_{t=1}^{T} p(\theta|x, w_t) \tag{3.10}$$

where T is the number of predictions for a given input, and w_t is the specific weight constellation [12] of the neural network that is active for the given prediction.

When implemented with a loss function that utilized the ELBO equation (described in follow-on sections), the predictive distribution calculated from T samples is theoretically approximate to the predictive distributions calculated using other Bayesian methods (such as reparameterization and flipout) [27].

3.3 Model Architectures

The following section (3.3) is adapted from [6], previously published by the IEEE Geoscience and Remote Sensing Letters, ©2021 IEEE[3],[4]

This section details the specific model architectures utilized in this work. To begin, deterministic and Bayesian configurations of deep residual networks (ResNeT) model [55], [56] architectures were chosen to be benchmarked in this work. The ResNet model was specifically chosen due to their high performance on ImageNet dataset classification tasks. Additionally, since ResNet model architectures perform best in conjunction with large datasets, it is uniquely suited for the dataset used in this study. For comparison, ImageNet is comprised of 14.1 million images, versus the 14.4 million images in the PMW GPM dataset used in this work.

3.3.1 Model Architectures: Deterministic and Bayesian ResNets

Past research has shown that convolutional neural networks are the best performing models for large scale image recognition tasks [2]. Deep neural networks, however, suffer from a Vanishing Gradient Problem [57] where the weights in lower layers receive smaller and smaller updates until the error gradient vanishes completely, leading to diminished returns with deeper networks. A solution to this problem was introduced in 2016 by He et al. [55] that allowed deeper networks to be trained, resulting in large performance gains. In this work, we utilized residual network (ResNet) architectures to train and develop both deterministic and Bayesian deep learning models for classification using the previously described data set.

[3]Reprinted, with permission, from Orescanin et al., "Bayesian deep learning for Passive Microwave Precipitation Type Detection," IEEE Geoscience and Remote Sensing Letters, 2021. This publication is a work of the U.S. government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States. IEEE will claim and protect its copyright in international jurisdictions where permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

[4]In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of the Naval Postgraduate School's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

In order to maximize classification performance with the computational resources available, 38-layer ResNet V2 [56] model architectures were used.

Additionally, the input size for each image was changed from previous work completed with this dataset by Petkovic et al. [5], instead of using smaller, 3 by 5 pixel 13-channel fields, the input data is pre-processed to provide pixel-level TBS (brightness temperatures) across a much wider 9 by 25 pixel domain. By utilizing these wider fields, each input image is expanded to account for an approximately 125 by 125 km area. These enhanced radiometric fields offer increased resolution for spatially-preserved signatures for a wide range of systems morphologies and offer a spatial-resolution enhancement over previous work. The model architectures used, therefore, are given an opportunity to distinguish between small-scale (e.g., isolated summer convection) and mesoscale (e.g., squall lines or mesoscale convective systems) features known to possess distinct properties.

Reference ResNet implementations support inputs from $299 \times 299 \times 3$ inputs in contrast to presented dataset that has input features from TBS that are $9 \times 25 \times 13$ in dimension. One difference between presented implementation and the original architectures for deterministic ResNet is in the size of the average pooling. The average pooling window before the activation is changed from 8×8 in the original implementation [55] to 2×2 due to the difference in input sizes and propagation of the dimensions through the architecture. Additionally, for Bayesian ResNet architectures probabilistic versions of layers were adopted in identical configuration as deterministic architectures. This was achieved by following approach in [58]. The number of trainable parameters for each model architecture used is given in the following figure:

Model Type:	Number of Trainable Parameters
Deterministic ResNet	1,124,322
Flipout ResNet	2,223,890
Reparameterization ResNet	2,223,890
Monte Carlo Dropout ResNet	1,124,322

Figure 3.2. The number of trainable parameters for each model architecture used in this work.

The Monte Carlo dropout layers were added following activation layers before the following

convolutional layer in accordance with the approach in [30]. Dropout rate was set to 10%. Note that the number of trainable parameters in the reparameterization and flipout implementations are double the number of trainable parameters in the deterministic and Monte Carlo dropout models. This is due to each weight distribution possessing two trainable values: the mean and standard deviation of the weight distribution.

3.4 Quantifying Model Uncertainties

While previous sections have described methods of implementing Bayesian deep learning, this section will describe methods of quantifying the specific model uncertainties that are present in Bayesian deep learning models. In Bayesian deep learning models, there are two main types of uncertainty that can be quantified: aleatoric uncertainty and epistemic uncertainty [59]. Aleatoric uncertainty represents the noise that is inherent within the dataset itself, such as noise due to the mechanism of the sensors generating the observations. Due to this property, aleatoric uncertainty cannot be reduced via improvements to the dataset [4]. Epistemic uncertainty, also known as model uncertainty, represents the uncertainty inherent within the model parameters [4], which means that epistemic uncertainty can be reduced given enough data. First, a discussion of how to extract the aleatoric and epistemic Uncertainties is presented, followed by how the total predictive uncertainty is generated.

3.4.1 Modeling Epistemic and Aleatoric Uncertainty

The method used in this thesis for extracting the epistemic and aleatoric uncertainties from the bulk predictive uncertainty follows the method used by Kwon et. al [31]. To derive this extraction, consider that the bulk uncertainty is the sum of aleatoric and epistemic components. From this, Kwon et. al construct this breakdown of the total predictive uncertainty into its aleatoric and epistemic components:

$$\frac{1}{T}\sum_{t=1}^{T} diag(\hat{p}_t) - \hat{p}_t^{\otimes}2 + \frac{1}{T}\sum_{t=1}^{T}(\hat{p}_t - \bar{p}_t)^{\otimes}2 \tag{3.11}$$

where $\frac{1}{T}\sum_{t=1}^{T} diag(\hat{p}_t) - \hat{p}_t^{\otimes}2$ represents the aleatoric component, $\frac{1}{T}\sum_{t=1}^{T}(\hat{p}_t - \bar{p}_t)^{\otimes}2$ represents the epistemic component, $\bar{p} = \sum_{t=1}^{T}\hat{p}/T$, and $\hat{p}_t$ represents the model output after it

is processed through the softmax function [31].

Extracting the uncertainties in this manner avoids the use of additional model layers as outlined in [4], and provides a more accurate estimate of the aleatoric and epistemic uncertainties than [4] as shown in [31].

In this thesis, the extraction was performed by implementing the pseudocode provided below [31]:

```
epistemic = np.mean(p_hat**2, axis=0) - np.mean(p_hat, axis=0)**2
aleatoric = np.mean(p_hat*(1-p_hat), axis=0)
```

Figure 3.3. epistemic and aleatoric extraction. Source: [31].

3.4.2 Quantifying Total Predictive Uncertainty

To quantify the total predictive uncertainty, a Bayesian neural network is constructed using one any one of the various Bayesian methods previously discussed. By utilizing a Bayesian neural network, the variational posterior distribution that the trained BNN produces captures the plausible set of model parameters, given the data. By taking N forward pass samples from the BNN, the total predictive uncertainty in a classification task is approximated via Monte Carlo integration by [4]:

$$p(y = c|x, X) \approx \frac{1}{N} \sum_{n=1}^{N} Softmax(f^{(W_n)}(x)) \tag{3.12}$$

where $p(y = c|x, X)$ represents the probability vector of the predicted class $y = c$ given a datapoint x from dataset X, and $f^{(W_n)}(x)$ is the logit result of a forward pass n through the BNN for the datapoint. The total predictive uncertainty is given by the summation of the aleatoric and epistemic uncertainties discussed previously [31]

$$T = \frac{1}{T} \sum_{t=1}^{T} diag(\hat{p}_t) - \hat{p}_t^{\otimes} 2 + \frac{1}{T} \sum_{t=1}^{T} (\hat{p}_t - \bar{p}_t)^{\otimes} 2 \tag{3.13}$$

where T represents the raw uncertainty output of a Bayesian neural network. In this thesis, this total uncertainty value is benchmarked across all three methods.

3.5 Training Methodology

The following section (3.5) is adapted from [6], previously published by the IEEE Geoscience and Remote Sensing Letters, ©2021 IEEE[5],[6]

The following section details the training methodology, hyperparameters, and loss functions used for the model training in this work. Architecture weights were initialized for training following He et al. [55]. An Adam optimizer was used with a starting learning rate of 0.001. A batch size of 128 images was used. Learning rate annealing was employed [60] via monitoring of validation loss such that the learning rate was reduced by a factor of 10 if the validation loss was not decreasing for 10 consecutive epochs. An early stopping strategy was utilized to regularize for overfitting [48]. The evaluation metrics used for the deterministic models were binary accuracy and binary cross-entropy loss.

Due to computational resource constraints, training was terminated at 600 epochs, which for a Bayesian model required about 3 weeks to train on a single NVIDIA RTX 8000 48GB GPU. Both deterministic and Bayesian models were trained with the same strategy for the fairness of benchmarking. No additional hyperparameters were tuned between the runs that would affect performance comparison between the model architectures with the exception of loss functions and KL Regularization affecting the Bayesian models only, which will be addressed in follow-on sections.

[5]Reprinted, with permission, from Orescanin et al., "Bayesian deep learning for Passive Microwave Precipitation Type Detection," IEEE Geoscience and Remote Sensing Letters, 2021. This publication is a work of the U.S. government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States. IEEE will claim and protect its copyright in international jurisdictions where permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

[6]In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of the Naval Postgraduate School's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

3.5.1 Negative Log-Likelihood Loss

The loss function used for the Bayesian models in this work consists of a code implementation of the ELBO. Recall the ELBO equation derived previously:

$$ELBO = log(p^*(D)) - KL[q^*||p^*] \tag{3.14}$$

where the first term is the log evidence of the approximate posterior and the second term is the KL divergence between the approximate prior and approximate posterior. The goal is to formulate this ELBO as a loss so that it can be minimized through a Gradient Descent Algorithm. Maximizing the first term of the ELBO finds the approximate posterior that provides the best predictions, which is equivalent to minimizing the negative log likelihood of the model predictions [61].

The KL term in the equation is calculated via the sum total KL distance between the approximate prior posterior distributions in each reparameterization and flipout layer in the models of those types. In the implementation in this work, the KL losses for each layer are calculated using Tensorflow Probability's method to calculate the KL loss between a given layer's prior and posterior distributions [62]. This KL loss is then added to each layer as a non-trainable weight value that propagates the model loss through the model output. In this manner, the KL loss is captured through the negative log-likelihood loss, and is able to be regularized as a hyperparameter. From this, negative log likelihood Loss is mathematically formalized as:

$$Loss = -\sum_{j=1}^{J} y_j log(\hat{y}_j) + R \sum_{n=1}^{N} KL[w_n(q^*)||w_n(p^*)] \tag{3.15}$$

where the first term is the negative log likelihood for J classes, y_j is the probability distribution of the correct class, $\hat{y}_j$ is the probability distribution of the predicted class, and the second term depicts the summation of the KL loss between each weight w layer n KL loss between the layer's approximate posterior p^* and prior q^* distributions, with a regularizer term R managing the effect of KL loss.

The KL loss regularization term R is used as a hyperparameter for training in this work in

order to tune the model for a trade-off between ELBO accuracy and model performance. Recall that the ELBO is merely a lower bound on the model uncertainty. By regularizing KL, the model is biased towards accurate predictions at the cost of reducing the bias towards reducing the KL distance between the approximate prior and posterior.

In this work, KL regularization is accomplished via two different schemes. The first scheme is setting the regularization term to zero in order to train models that are completely biased towards predictive accuracy. The second scheme, which is detailed in previous work [31] called batch annealing, is to train the models while reducing the KL loss regularization within each batch from $R = 1$ to $R = 0$.

In the code base, this scheme is implemented using a KL loss scheduler class as a model callback [63]. This class adjusts the KL loss weight values of each Bayesian layer according to the annealing scheme, and outputs the KL loss values to the user at the end of each epoch. In this work, the reparameterization and flipout models are each trained according to the two KL loss regularization schemes in order to study the effect that this regularization has on model performance and uncertainty quality.

3.6 Testing Methodology

After training, the models were evaluated on the test dataset containing $\sim$ 1.4 million TBS feature vectors. To compare to the results of Petkovic et al. 2019 [5], ResNet38 V2s both in deterministic [55], [56] and Bayesian reparameterization, flipout, and Monte Carlo dropout implementations were trained using the same training dataset and evaluated on the held-out test dataset. For the Bayesian models, predictive distributions were obtained via multiple stochastic forward passes through the network. Bayesian model uncertainties were then extracted from these predictive distributions. Bayesian models are presented with ensembles of T=25 stochastic forward passes. Deterministic models utilized a single forward pass.

In addition to the evaluated model performances, this work utilizes the Goddard Profiling Algorithm (GPROF) to compare deterministic and Bayesian model performance to operational precipitation type retrieval. GPROF is the current operational rainfall algorithm used in the Global Precipitation Climatology Project (GPCP) [64] and GPM [65] rainfall

products. GPROF estimates both the rainfall rate and the precipitation type by matching observed TBS to hydrometeor profiles using a Bayesian approach. TBS were computed at the observation frequencies using a one-dimensional Eddington approximation [66], [44] [67].

A small portion of the overall dataset containing a GPM orbit over Hurricane Lane near peak intensity on 11 August 2018 was utilized as a case study and serving set to observe performance comparisons over a continuous spatial region of interest and to demonstrate quantifying Bayesian model uncertainty to improve model predictive performance. This portion of the dataset was held out of the training and test sets.

3.7 Performance Benchmarks

This next section details the specific performance benchmarks used to evaluate all of the Bayesian methods implemented in this work. The Bayesian models were compared across all of the performance benchmarks utilizing the same standards for uncertainty quantification in order to maintain a homogeneous comparison. Additionally, the uncertainty quantification benchmarks are used to determine the scale and quality of the total predictive uncertainty, epistemic, and aleatoric uncertainties expressed by the Bayesian models. Furthermore, the time required to train and predict using each model is expressed as an analogue for the Bayesian model computational complexity.

In addition to the uncertainty quantification and computational complexity benchmarks used, a suite of standard deep learning performance benchmarks are used to compare the Bayesian model outputs to the deterministic Model and the operational GPROF algorithm.

3.7.1 Uncertainty Screening

To measure the effects of uncertainty on Bayesian model performance, the effect of screening predictions that registered an uncertainty value above a certain threshold was examined. Each measure of uncertainty was tested in this manner to determine what the effects of screening each type had on the Bayesian model performances. All screening values used were held consistent across the different model implementations such that 80% of the prediction set was maintained.

Total predictive uncertainty $U(p)$ was screened by first measuring it via the summation of

the calculated aleatoric and epistemic uncertainties (see Uncertainties section), and then removing predictions from the output array such that 80 percent of the dataset remained containing the predictions with the least amount of total predictive uncertainty. aleatoric $A(p)$ and epistemic $E(p)$ uncertainties were screened in a similar fashion. Each Bayesian model was evaluated on the performance benchmarks in two modalities: uncertainty-screened and non-screened. In this manner, it is possible to view the effects of uncertainty screening on model results and benchmark each Bayesian model on the quality of uncertainty output.

3.7.2 Benchmark Metrics

As stated previously, all models used in this study are benchmarked using a comprehensive suite of established and novel methods. All benchmarks were calculated in the same manner for both the deterministic Models and Bayesian models. Additionally, the operational GPROF algorithm was also evaluated with the applicable benchmarks on the same test dataset.

The first benchmark metric used to evaluate raw model performance is predictive accuracy drawn from the scikit-learn python library [68] where the predicted labels are compared with the true labels. For the Bayesian models, the predictive labels were generated by applying the softmax function to the average of the predicted logits.

The next benchmark metric utilized is the Area Under the Receiver Operating Characteristic (aucROC) score [69] and plotted curve. This is a measure of how well each model is able to distinguish between the different output classes. This measures how well each model is able to predict a correct classification of stratiform rain as stratiform rain versus how well it can predict a correct classification of convective rain as convective rain. A score of 1.0 signifies an ideal classifier that can make perfect predictions, whereas a score of 0.50 represents the performance of a random classification prediction.

In addition to the aucROC score metric, Precision, Recall, and F1-score metrics are also calculated for each model. Precision is calculated as $P = TP/TP + FP$ where FP represents false positive classifications, and measures the rate at which a predicted class is actually the correct class (i.e., predicted stratiform rain is actually stratiform rain) [70].

Recall is calculated as $R = TP/TP + FN$ and measures the rate at which a correct class

is predicted (convective rain is predicted by the model to be convective rain) [70]. F1-score is the weighted average between precision and recall given mathematically by $F1 = 2(precision * recall / precision + recall)$ [70].

Next, benchmark plots will detail the quantification and quality of the Bayesian model Uncertainties. The first plots generated are histograms of a specified model uncertainty (Figure 4.1), where the log base 10 count of the number of example in each bin is given on the y-axis, and the uncertainty bins are given on the x-axis. Additionally, a cumulative distribution of the predictions is overlaid over this histogram so that the percentage of data that falls at or below an uncertainty bin can be determined. Histograms are developed for each Bayesian model for total predictive uncertainty, epistemic, and aleatoric uncertainties and are overlaid in the same plot.

Finally, spatial plots of uncertainty (Figures 4.2 - 4.9) will be shown for the region of interest. These plots detail a swath of a region of interest given in stratiform and convective pixel outputs from (in order from left to right) the operational GPROF prediction, the DPR-derived true labels, the given Bayesian model prediction, and the spatial plot of a Bayesian model's particular uncertainty metric. These spatial plots detail what regions the model experiences the most uncertainty, broken down across the three uncertainty types measured.

In summary, three different Bayesian model implementations (flipout, reparameterization, Monte Carlo dropout) with two different levels of KL-divergence term regularization will be benchmarked for predictive performance and uncertainty quality on a large-scale, multi-spectral satellite image dataset against the operational GPROF algorithm and a deterministic CNN of the same type. The benchmarking metrics and plots detailed previously will be used to comprehensively examine the performance of the Bayesian models while comparing the quality of their expressed uncertainties.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results

The reported results analyze and benchmark the Bayesian model performances across two separate datasets. The first dataset analyzed consists of the held-out test dataset containing 1.4 million feature vectors. The second dataset is a region of interest that consists of a swath of Hurricane Lane Southeast of Hawaii in August 2018 that contains 1800 feature vectors. The test dataset was used to benchmark performance metrics and expressed model uncertainties across the presented Bayesian modalities, and the second dataset was used to serve the data to generate spatial maps of the Bayesian model outputs and the corresponding uncertainties expressed by the models. These datasets enabled a comprehensive benchmark of the Bayesian models and provided exhaustive data towards an analysis of the quality of the Bayesian model performance and quality of uncertainty expressed by them.[7,8]

4.1 Test Dataset Results

This section analyzes results for the model predictions conducted on the test dataset. The Bayesian models (Flipout ResNet38, Reparameterization ResNet38, Flipout ResNet38 with KL annealing, Reparameterization ResNet38 with KL Annealing, MC Dropout) performed $T = 25$ Monte Carlo predictions per feature vector on the test dataset and were compared with the deterministic model (ResNet38) and GPROF algorithm predictions.

[7]Chapter 4 reprinted, with permission, from Ortiz et al., "A Systematic Evaluation of Bayesian Deep Learning on Satellite Imagery for Classification," IEEE, 2021. This publication is a work of the U.S. government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States. IEEE will claim and protect its copyright in international jurisdictions where permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

[8]In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of the Naval Postgraduate School's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Table 4.1. Unfiltered performance metrics on the test dataset for each model architecture.

Architecture	**Acc** [%]	**aucROC**	**Prec**	**Rec**
GPROF	74.3	0.743	0.83	0.74
Det. ResNet38 V2	86.0	0.94	0.86	0.86
Flipout ResNet38 V2, T=25	92.7	0.977	0.927	0.927
Reparam ResNet38 V2 T=25	92.0	0.974	0.92	0.92
MC Dropout ResNet38 V2 DR=0.10, T=25	86.0	0.941	0.86	0.86
Flipout ResNet38 V2 KL Annealing, T=25	86.6	0.946	0.87	0.87
Reparam ResNet38 V2 KL Annealing, T=25	86.4	0.944	0.86	0.86

Table 4.1 details the raw performance results for the models. All of the deep learning models significantly outperformed the operational GPROF algorithm in all calculated metrics, and all of the Bayesian model modalities outperformed the deterministic model with the exception of the MC Dropout model. The non-KL annealing modalities (annotated as Flipout and Reparam in the table) significantly outperformed the rest of the Bayesian models in all calculated performance metrics.

Table 4.2. Total predictive uncertainty-filtered performance metrics on the test dataset, T=25.

Architecture	**Acc** [%]	**aucROC**	**Prec**	**Rec**
Flipout ResNet38 V2, U > 0.18	97.2	0.988	0.97	0.97
Reparam ResNet38 V2 U > 0.23	97.2	0.989	0.97	0.97
MC Dropout ResNet38 V2 DR=0.10 U > 0.41	92.6	0.97	0.93	0.93
Flipout ResNet38 V2 KL Annealing, U > 0.40	93.1	0.974	0.93	0.93
Reparam ResNet38 V2 KL Annealing, U > 0.405	92.9	0.972	0.93	0.93

Table 4.2 shows how the Bayesian models performed when their predictions are filtered based upon total predictive uncertainty values. In this table, the total predictive uncertainty

values were chosen so that 80 percent of the dataset is retained. In other words, the bottom 20 percent of predictions in terms of predictive uncertainty were removed from the corpus before performance metrics were calculated. The values indicated by the $U >$ show the uncertainty threshold to retain 80 percent of the dataset. By doing this, it is possible to see how each of the Bayesian modalities numerically expresses total predictive uncertainty.

Every Bayesian modality's total predictive uncertainty-filtered performance exceeded the performance of their unfiltered counterpart. Notably, the non-KL annealing modalities achieved near-ideal performance in all metrics. The KL-annealing modalities and the MC Dropout modalities performed similarly and achieved near-identical performance increases across all metrics which exceeded the performance increases seen by the non-KL annealing modalities. The uncertainty threshold between the reparameterization and flipout models differed by a significant margin, suggesting that the flipout model expresses lower uncertainty overall. Additionally, the quantity of uncertainty necessary to remove 20 percent of the dataset in the KL-annealing and Monte Carlo modalities was roughly double the amount required by the non-KL annealing models. Taken together, these two results indicate that the KL-annealing and MC dropout models express both a higher amount of total predictive uncertainty and a higher quality of uncertainty, as these models achieved a greater performance from filtering it out.

Table 4.3. Epistemic uncertainty-filtered performance metrics on the test dataset, T=25.

Architecture	**Acc** [%]	**aucROC**	**Prec**	**Rec**
Flipout ResNet38 V2, E > 2E-6	96.6	0.988	0.97	0.97
Reparam ResNet38 V2, E > 5E-6	96.5	0.988	0.96	0.96
MC Dropout ResNet38 V2 DR=0.10, E > 9E-3	91.0	0.968	0.91	0.91
Flipout ResNet38 V2 KL Annealing, E > 9E-4	91.3	0.971	0.91	0.91
Reparam ResNet38 V2 KL Annealing, E > 6.3E-3	91.4	0.970	0.91	0.91

Table 4.3 details how the Bayesian models performed when their predictions were filtered based upon epistemic uncertainty values. In an identical fashion to the total predictive uncertainty, the bottom 20 percent of predictions in terms of epistemic uncertainty were removed prior to the calculation of the performance metrics. The values indicated by the

$E >$ show the epistemic uncertainty threshold to retain 80 percent of the dataset.

In a similar result to the total predictive uncertainty, the Bayesian models received a performance boost from filtering the epistemic uncertainty. This performance boost, however, was not as significant as the one that resulted from filtering the total predictive uncertainty. The non-KL annealing models continued to perform better than the other modalities, but expressed epistemic uncertainty thresholds that were several orders of magnitude lower than the other modalities. Furthermore, the trend from the total predictive uncertainty screening continues with the flipout model expressing a threshold that is less than half that of the Reparameterization model. Additionally, all of the models expressed epistemic uncertainty thresholds that were extremely small. These small epistemic uncertainty values mean that all of the models' parameter distributions did not express much uncertainty in comparison to the total predictive uncertainty value, meaning that the models themselves were quite confident in their predictions and that their parameter distributions converged to a solution that may not be significantly improved by increasing the size of the dataset.

Table 4.4. Aleatoric uncertainty-filtered performance metrics on the test dataset, T=25.

Architecture	**Acc** [%]	**aucROC**	**Prec**	**Rec**
Flipout ResNet38 V2, A > 0.18	97.7	0.99	0.98	0.98
Reparam ResNet38 V2, A > 0.23	97.2	0.989	0.97	0.97
MC Dropout ResNet38 V2 DR=0.10, A > 0.41	92.1	0.968	0.92	0.92
Flipout ResNet38 V2 KL Annealing, A > 0.40	93.0	0.973	0.93	0.93
Reparam ResNet38 V2 KL Annealing, A > 0.40	92.8	0.972	0.93	0.93

Table 4.4 shows how the Bayesian models performed when their predictions were filtered based upon aleatoric uncertainty values to retain 80 percent of the dataset in a fashion identical to the previous filtering methods. The retention of 80 percent of the dataset was chosen to demonstrate the filtering method and is able to be adjusted depending upon the needs of the user. The values indicated by $A >$ show the aleatoric uncertainty threshold needed to retain 80 percent of the dataset.

The results from the aleatoric screening show notable differences from the previous epistemic uncertainty analysis. First, the non-KL annealing Bayesian modalities achieved performance increases that met or, in the case of the non-KL annealing flipout model, exceeded the performance increase seen from screening based upon total predictive uncertainty. The non-KL annealing flipout model achieved near-optimal performance based upon the auROC score, and its classification accuracy was half a percentage point better than the accuracy seen when total predictive uncertainty screening was used. Next, the KL annealing modalities and the Monte Carlo dropout model achieved similar performance increases to filtering with total predictive uncertainty, with threshold values that were on par with or, in the case of the KL annealing reparameterization model, slightly lower than the total predictive uncertainty screening values.

Notably, the scale of the aleatoric uncertainty thresholds are nearly identical to the total predictive uncertainty thresholds previously discussed. This result means that the majority of the total predictive uncertainty is a result of uncertainty inherent to the dataset itself, not the uncertainty inherent in the model parameters. This result further suggests that improving model performance may require changes to how the data is collected, such as remote senor improvements, as the aleatoric uncertainty cannot be reduced by the inclusion of additional training data [4]. The differences in the expressed aleatoric uncertainty thresholds also reinforce the trend that the non-KL annealing models express a lower uncertainty quality and quantity than the KL annealing and Monte Carlo dropout models, as those models express uncertainty values that are not only significantly larger than the non-KL annealing values, but are also consistent across the MC dropout and KL annealing modalities.

Table 4.5. Average uncertainties over the test dataset. Table 4.5 is adapted from [71], previously published by the IEEE ©2021.

Architecture	**Mean Aleatoric**	**Mean Epistemic**
Flipout ResNet38 V2	9.475e-02	1.360e-06
Reparam ResNet38 V2	1.054e-01	4.340e-06
MC Dropout ResNet38 V2 DR=0.10	2.000e-01	5.113e-03
Flipout ResNet38 V2 KL Annealing	1.861e-01	5.843e-04
Reparam ResNet38 V2 KL Annealing	1.905e-01	3.666e-03

Next, Table 4.5[9,10] details the average aleatoric and epistemic uncertainty values calculated over the entire dataset for each model. This table further reinforces previously discussed trends. First, the non-KL annealing models express much lower values of both aleatoric and epistemic uncertainties than their KL annealing modalities and the MC Dropout model, with the non-KL annealing flipout model expressing the least amount of uncertainty overall. At the other end of the spectrum, the MC dropout model expressed the highest amount of aleatoric and epistemic uncertainty.

Across all of the models, however, these results indicate that the models are quite certain in their predictions. This indicates that efforts to reduce the uncertainty and to increase the performance of these models should be focused upon reducing the aleatoric uncertainty, or the uncertainty due to the dataset itself. This result is consistent with previous research [31], [4] that suggests that dataset improvements are the most effective way to improve the performance of well-converged Bayesian models.

[9]Reprinted, with permission, from Ortiz et al., "A Systematic Evaluation of Bayesian Deep Learning on Satellite Imagery for Classification," IEEE, 2021. This publication is a work of the U.S. government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States. IEEE will claim and protect its copyright in international jurisdictions where permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

[10]In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of the Naval Postgraduate School's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Table 4.6. Training time per epoch [TPE] vs. model size.

Architecture	**TPE [s]**	**Model Size (params)**
Deterministic ResNet38 V2	1874	1124322
Flipout ResNet38 V2	3311	2223890
Reparam ResNet38 V2	1815	2223890
MC Dropout ResNet38 V2 DR=0.10	2371	1124322
Flipout ResNet38 V2 KL Annealing	4108	2223890
Reparam ResNet38 V2 KL Annealing	2886	2223890

Table 4.6 describes the average time each training epoch took to complete for each of the models trained on the overall test dataset and lists the number of trainable parameters in each model. The non-KL annealing reparameterization modality was the fastest model to train, despite having twice the trainable parameters as the deterministic model. This is a surprising result, given the additional calculations that are necessary to run the reparameterization model. This suggests that the training time was weighted towards loading and moving data into and out of main memory for processing by the model, and the additional calculations required by the reparameterization model to calculate and process the negative log-likelihood loss did not adversely affect the training time.

Next, the MC dropout modality required approximately 500 seconds longer on average for each epoch to complete. This result makes sense in the context of the model's construction, as it contained 36 dropout layers. Each dropout layer requires additional calculation to eliminate a 10 percent of the previous layer's connection during every forward pass through the network, so the additional training time required per epoch makes sense given the greater computational burden on the hardware. Next, the KL annealing reparameterization ResNet required approximately 500 more seconds per epoch than the MC dropout model. The calculation of the KL loss between the approximate prior and posterior distributions for each Bayesian model layer accounts for this difference, as it requires an additional set of calculations per forward pass. The flipout non KL annealing and KL annealing models took the longest time to train. Again, this result makes sense in the context of the

calculations that take place to compute the flipout method. As demonstrated previously in this work, the flipout method requires twice the number of floating point operations than the reparameterization method in order to calculate a more precise gradient. In theory, this should translate to a more accurate model over the other Bayesian methods, but the performance gains did not manifest themselves, which reduces the utility of the flipout method due to the much-greater training time of the flipout modalities.

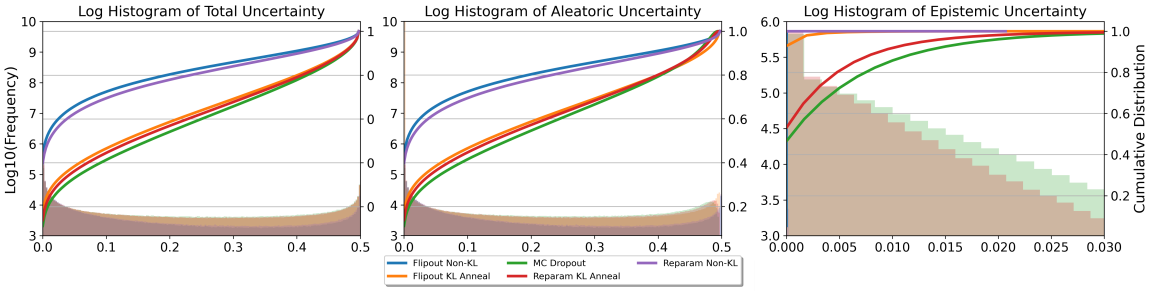


Figure 4.1. Log histogram and CDF of expressed uncertainties for all Bayesian model modalities

Figure 4.1 details the log-histogram and CDF of the uncertainties expressed by the Bayesian models. From this figure, it can be observed that the non KL annealing modalities express a much lower amount of uncertainty than the other models. This shows that removing the calculation of the KL divergence term in the loss function reduces the effectiveness of the models in expressing uncertainty, and leads to an over-confidence in their predictions as expressed by the CDFs in the epistemic uncertainty portion of the plot. The non KL modalities express extremely low amounts of epistemic uncertainty while also expressing a lower amount of aleatoric uncertainty. When comparing the two non KL modalities, the flipout method expresses lower uncertainty than the reparameterization model.

In contrast, the KL annealing models (with the notable exception of the KL annealing flipout model) express a much higher amount of total, aleatoric, and epistemic uncertainty than the non-KL annealing modalities. The flipout and reparameterization models all expressed less uncertainty than the MC dropout model, and the MC dropout model expressed a much higher amount of epistemic uncertainty. The KL-annealing flipout and reparameterization models expressed nearly identical amounts of aleatoric uncertainty, but the KL-annealing flipout model expressed extremely low amounts of epistemic uncertainty in line with the

amount of epistemic uncertainty expressed by the non KL annealing models.

4.2 Region of Interest Dataset Results

This section analyses results for the model predictions conducted on the serving dataset on a specific region of interest. The Bayesian models (Flipout ResNet38, Reparameterization ResNet38, Flipout ResNet38 with KL annealing, Reparameterization ResNet38 with KL Annealing, MC Dropout) performed $T = 25$ Monte Carlo predictions per feature vector on the serving dataset and were compared with the deterministic model (ResNet38) and GPROF algorithm predictions.

The serving dataset is a region of interest that consists of a swath of Hurricane Lane as it reached peak intensity southeast of Hawaii in August 2018 that contains 1800 feature vectors. This serving swath was chosen due to the rarity of its presentation, as the coincidence of the GPM satellites, DPR sensor, and an intense tropical cyclone can be classified as a statistical rarity. As such, this serving dataset provides an analysis of model performance on data that is likely a very small fraction of the training dataset. While a rain-type classification task on the inner core of a tropical cyclone is not dynamically meaningful from a meteorological perspective, the comparison with GPROF and DPR is nonetheless a useful demonstration of Bayesian model performance on classifying remote sensing data of the atmosphere.

Table 4.7. Model performance metrics for the region of interest, T=25.

Architecture	**Acc** [%]	**aucROC**	**Prec**	**Rec**	**F1**
GPROF	69.5	0.715	0.73	0.70	0.70
Det. ResNet38 V2	83.2	0.848	0.95	0.76	0.84
Flipout ResNet38 V2	81.4	0.902	0.83	0.82	0.81
Reparam ResNet38 V2	81.7	0.913	0.84	0.82	0.82
MC Dropout ResNet38 V2 DR=0.10	82.4	0.930	0.85	0.82	0.83
Flipout ResNet38 V2 KL Annealing	83.4	0.934	0.85	0.83	0.84
Reparam ResNet38 V2 KL Annealing	83.2	0.932	0.85	0.83	0.83

Table 4.7 shows the unfiltered Bayesian model performances on the serving dataset. Ad-

ditionally, the performances of the operational GPROF algorithm and a deterministic ResNet38 V2 model are also shown. All models performed worse on the serving dataset in comparison to the test dataset. As discussed in the introduction to this section, this is likely due to the pseudo out-of-distribution nature of the data. Notably, the Bayesian models that performed the best on the test dataset, the non KL Annealing flipout and reparameterization models, showed the greatest performance decrease on the serving dataset. This result indicates that these models have overfit to the training dataset, as the other Bayesian modalities suffered a 3 percent accuracy decrease against the test dataset, versus a performance decrease of over 10 percent for the non KL annealing models. Notably, the deterministic model performed on par with the KL annealing Flipout and reparameterization models, with the MC dropout model only outperforming the non KL annealing models and the operational GPROF algorithm.

Table 4.8. Model performance metrics for total predictive uncertainty-filtered region of interest, T=25.

Architecture	**Acc** [%]	**aucROC**	**Prec**	**Rec**	**F1**
Flipout ResNet38 V2, U > 0.045	87.8	0.934	0.89	0.99	0.88
Reparam ResNet38 V2, U > 0.07	88.9	0.937	0.90	0.89	0.89
MC Dropout ResNet38 V2, U > 0.35	89.2	0.960	0.91	0.89	0.89
Flipout ResNet38 V2 KL Anneal, U > 0.34	89.2	0.963	0.91	0.89	0.89
Reparam ResNet38 V2 KL Anneal, U > 0.34	89.2	0.962	0.91	0.89	0.89

Table 4.8 above details the Bayesian model performance on the region of interest when the model predictions are filtered based upon total predictive uncertainty such that 80 percent of the dataset is retained. In congruence with the analysis on the test dataset, the non KL annealing modalities express a smaller performance gain with this filter than the KL annealing and MC dropout modalities. Additionally, the scale of the uncertainty values reinforce previous analysis, as the non KL models express total predictive uncertainty values that are far lower than the other models. This reinforces the analysis that these models overfit to the training set, as their performance suffered while the models' confidence in their predictions was higher than the confidence expressed by the other Bayesian modalities.

Table 4.9. Performance metrics for epistemic-filtered region of interest, T=25.

Architecture	**Acc** [%]	**aucROC**	**Prec**	**Rec**	**F1**
Flipout ResNet38 V2, E > E-7	87.0	0.929	0.88	0.87	0.87
Reparam ResNet38 V2, E > 1E-6	88.7	0.938	0.90	0.89	0.89
MC Dropout ResNet38 V2, E > 1E-2	89.7	0.960	0.91	0.90	0.90
Flipout ResNet38 V2 KL Anneal, E > 1E-3	88.9	0.957	0.90	0.89	0.89
Reparam ResNet38 V2 KL Anneal, E > 6E-3	88.6	0.956	0.90	0.89	0.89

The analysis of the epistemic uncertainty-filtered results in Table 4.9 reinforce prior analysis. The non KL annealing modalities expressed epistemic uncertainty filters that were approximately $\frac{1}{6}$ and $\frac{1}{4}$ the values expressed on the test dataset for the reparameterization and flipout models, respectively. This is in contrast to the epistemic uncertainty filter values expressed by the KL annealing and MC dropout models that were either orders of magnitude greater on the serving set (for MC dropout and flipout) or on par with (in the case of reparameterization) the epistemic uncertainty filter values expressed by the test dataset. The results expressed by these Bayesian modalities are consistent with prior research on out-of-distribution (OOD) results with Bayesian models on smaller datasets that show that Bayesian models express higher epistemic uncertainty in regions of data that are not represented in the training data corpus. Taken together, this further reinforces the notion that ignoring the KL term in the negative log-likelihood loss function for Bayesian models results in overfitting to the training data in a manner that is similar to deterministic models.

Table 4.10. Model performance metrics for aleatoric-filtered region of interest, T=25.

Architecture	**Acc** [%]	**aucROC**	**Prec**	**Rec**	**F1**
Flipout ResNet38 V2, A > 0.045	87.8	0.934	0.88	0.87	0.87
Reparam ResNet38 V2, A > 0.07	88.9	0.937	0.90	0.89	0.89
MC Dropout ResNet38 V2, A > 0.34	89.0	0.960	0.91	0.89	0.89
Flipout ResNet38 V2 KL Anneal, A > 0.33	89.6	0.964	0.91	0.90	0.90
Reparam ResNet38 V2 KL Anneal, A > 0.32	89.5	0.963	0.91	0.89	0.90

The aleatoric uncertainty filtered results in Table 4.10 also reinforce the trends seen previously on the test dataset. While the aleatoric uncertainty filters are lower for all Bayesian models on the serving set, the KL annealing and MC dropout models produce similar amounts of aleatoric uncertainty, reinforcing the idea that these models produce well-calibrated uncertainties that are consistent across different types of models. Additionally, the reduced aleatoric uncertainty filter values produced by the Non KL modalities also reinforces the notion that, similar to previous analysis, ignoring the KL divergence between the prior and posterior distributions during loss calculation greatly reduces the calibration and utility of the uncertainties produced by those types of models.

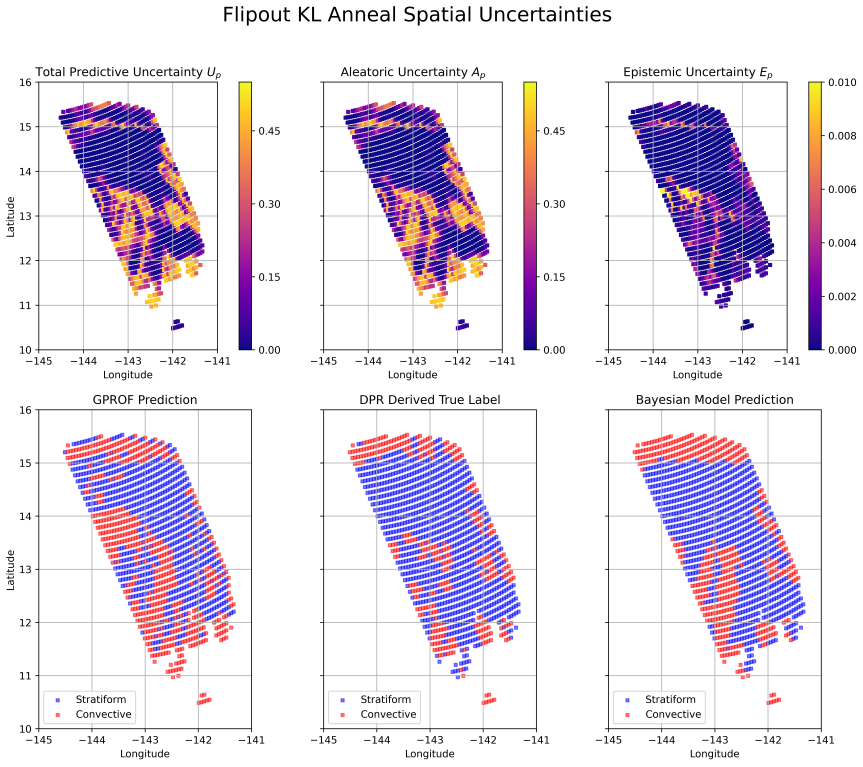


Figure 4.2. Combined spatial uncertainty and prediction plot comparison for the KL annealing flipout model. Uncertainties are given in the top row, and model outputs are given in the bottom row.

Figure 4.2 details the spatial results for the KL annealing flipout model. The spatial predictions show that the model is superior at detailing storm structure than the GPROF model, especially in regions where convective precipitation is bordered by stratiform precipitation. The spatial uncertainties detail precisely which regions of the hurricane within which the model experiences epistemic and aleatoric uncertainty. Those regions coincide with areas of incorrect classification. Notably, the regions expressing large aleatoric uncertainty coincide with regions expressing epistemic uncertainty. This shows that the epistemic uncertainty is tied spatially to regions where the dataset expresses noise. However, information gain may be possible through training the model with data that expresses epistemic uncertainty values that are higher than the test dataset average. Since the highlighted areas in the epistemic uncertainty plot express values orders of magnitude greater than the test dataset average, the model performance can likely be improved by including data on hurricane structures to

the training data corpus.

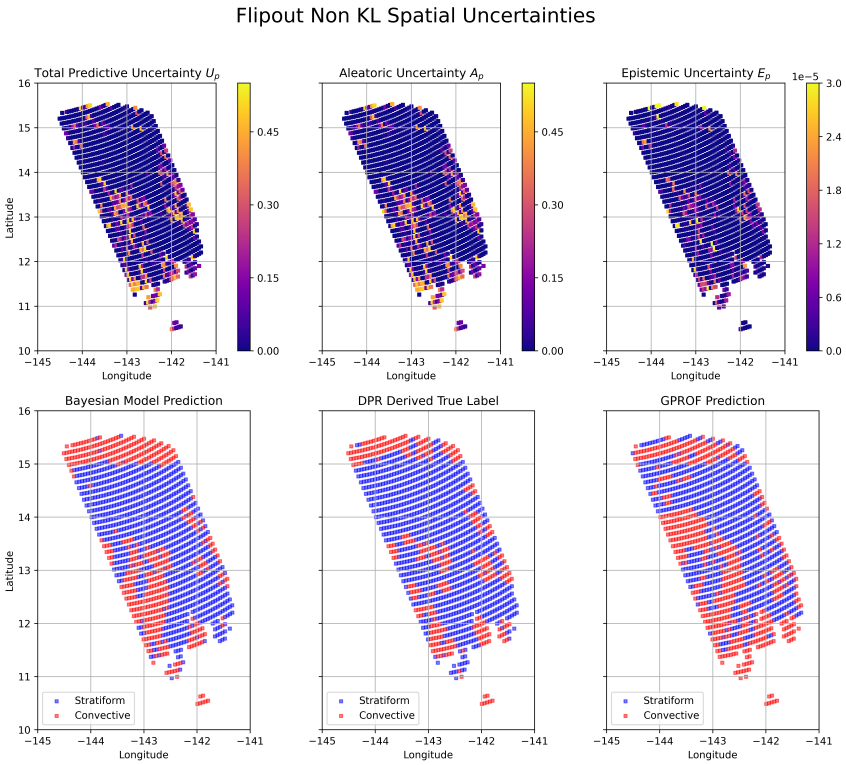


Figure 4.3. Combined spatial uncertainty and prediction plot comparison for the non KL annealing flipout model.

In contrast to the previous result, the non KL annealing flipout modality does not express quality spatial uncertainty (Figure 4.3). The expressed uncertainties are unclear in comparison to the uncertainties expressed by the KL annealing model that were tied to specific storm structures. Subsequently, the model over-classifies convection when compared to its KL annealing counterpart. Due to this lack of uncertainty quality, it is difficult to ascertain regions within which the dataset and model results could be improved.

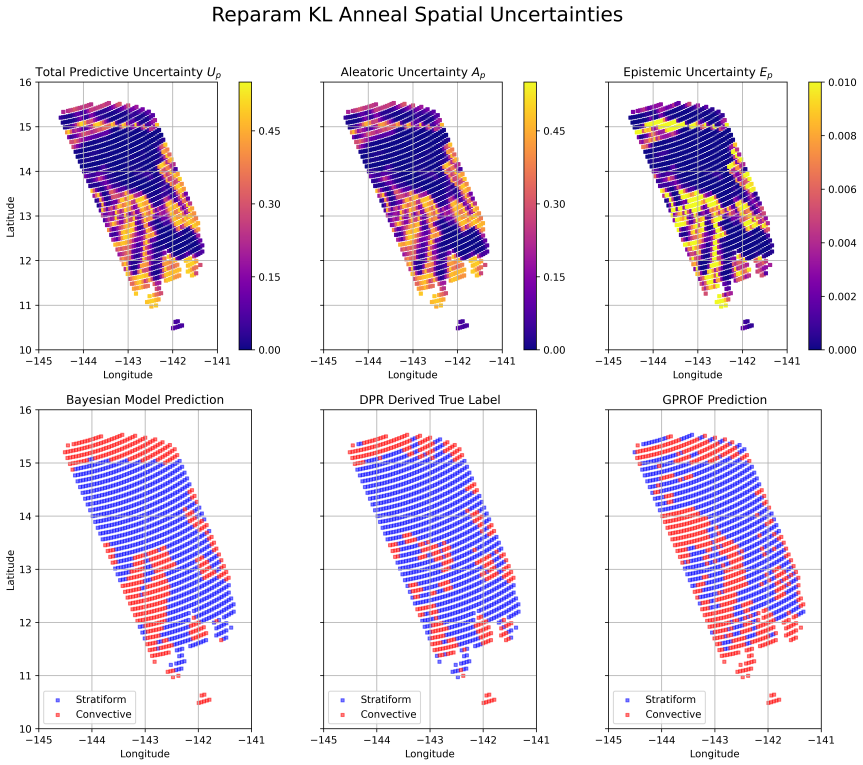


Figure 4.4. Combined spatial uncertainty and prediction plot comparison for the KL annealing reparameterization model.

Similarly to the KL annealing flipout model, the KL annealing reparameterization model (Figure 4.4) expresses detailed spatial uncertainties that are strongly correlated with the hurricane's structure. In line with Table 4.9, the KL annealing reparameterization model expresses higher epistemic uncertainties than the KL annealing flipout model. In a similar fashion, these higher epistemic uncertainties over the test dataset baseline demonstrate possible information gain.

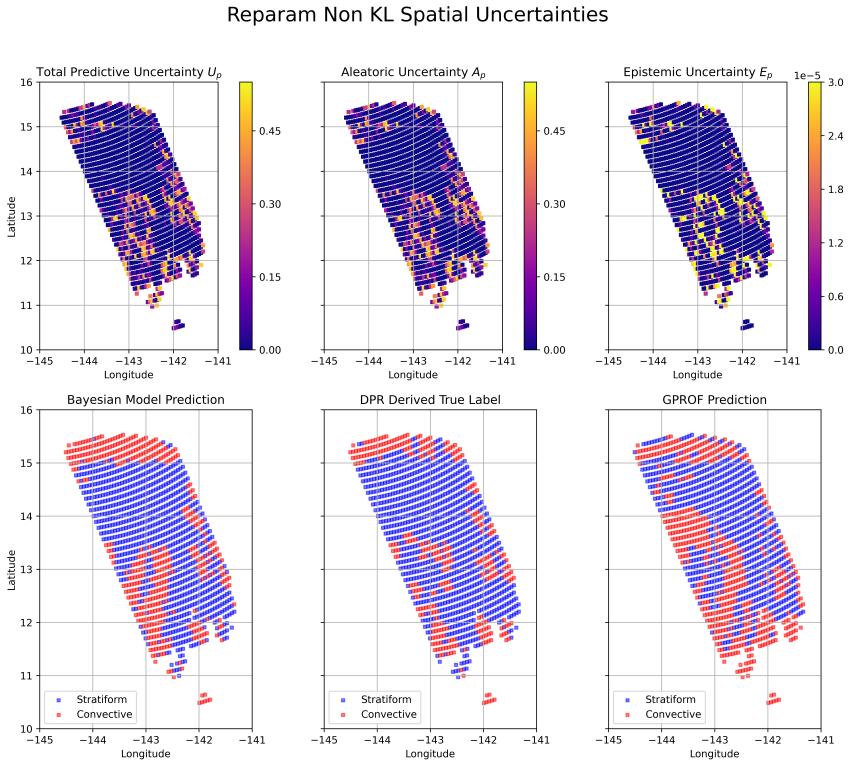


Figure 4.5. Combined spatial uncertainty and prediction plot comparison for the non KL annealing reparameterization model.

In contrast to the KL annealing reparameterization modality, the non KL reparameterization model (Figure 4.5) expresses uncertainties in a similar fashion to the non KL annealing flipout model (Figure 4.3). The regions of uncertainty are similar in shape and in scope, and are greater in magnitude than the non KL flipout model, but they also lack detail. It is not clear from these spatial uncertainties where the model and dataset could be improved, as the uncertainties do not detail specific storm structures nor are they strongly correlated to specific regions where the model performed incorrect classifications.

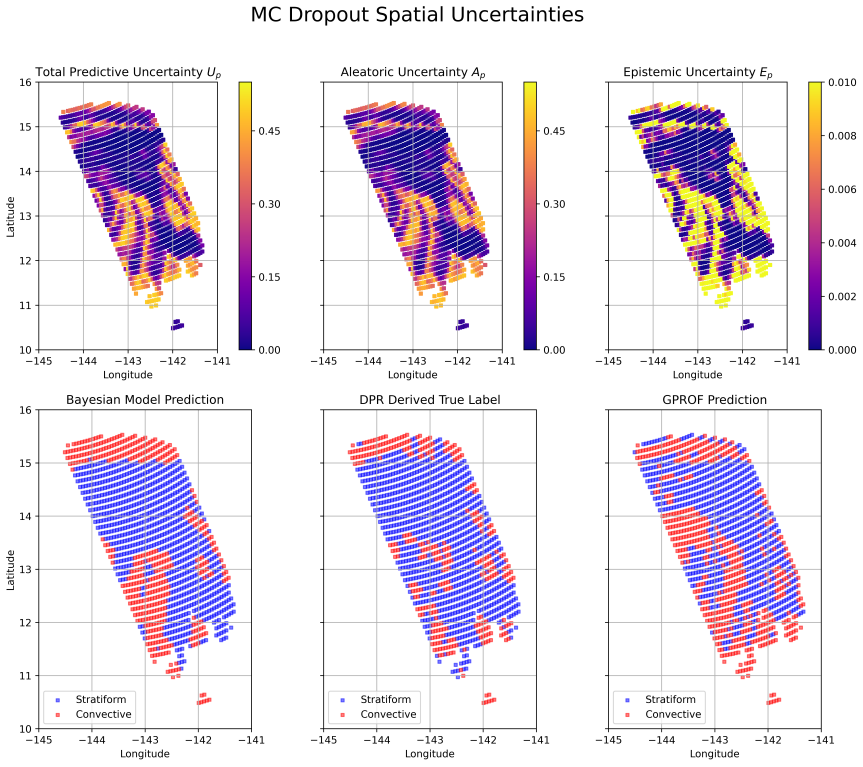


Figure 4.6. Combined spatial uncertainty and prediction plot comparison for the MC dropout model. Figure 4.6 is adapted from [71], previously published by the IEEE ©2021.

The MC dropout spatial results agree with Table 4.9 that show that the MC dropout model expressed higher epistemic uncertainty on the serving dataset than the other Bayesian models (Figure 4.6[11,12]). The spatial uncertainties expressed reinforce the notion that the

[11]Reprinted, with permission, from Ortiz et al., "A Systematic Evaluation of Bayesian Deep Learning on Satellite Imagery for Classification," IEEE, 2021. This publication is a work of the U.S. government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States. IEEE will claim and protect its copyright in international jurisdictions where permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

[12]In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of the Naval Postgraduate School's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada

main source of the uncertainty (and incorrect classifications) lies with noise inherent to the dataset itself. This provides an important advantage over the deterministic model and GPROF results, as it is possible with the Bayesian models to determine where (and if) it is possible to improve the model through the inclusion of more data, and where the dataset itself can be improved through sensor and data collection improvements.

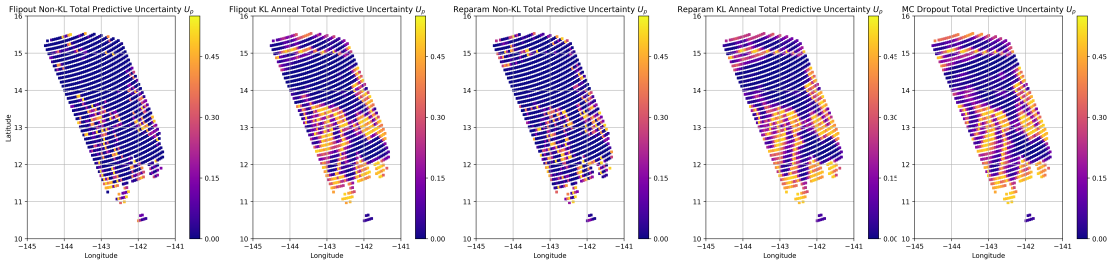


Figure 4.7. Combined total predictive uncertainty spatial comparison of the Bayesian models.

Figure 4.7 compares the spatial total predictive uncertainty expressed by the Bayesian models. This spatial plot shows that the non KL annealing models express total predictive uncertainty that is of a much lower quality than the KL annealing and MC dropout models. Notably, the KL annealing and MC dropout models are consistent in the location and scale of the total predictive uncertainty, whereas the non KL annealing models show significant differences in the location of their total predictive uncertainty, leading to an inconclusive spatial uncertainty quantification for those models when compared to the modalities that include KL divergence in their calculations.

may supply single copies of the dissertation.

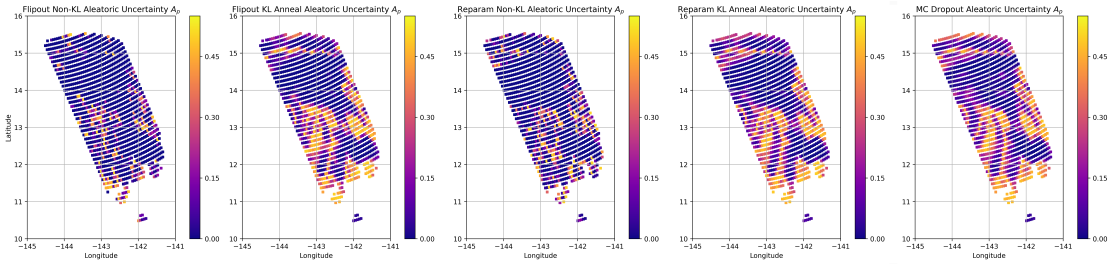


Figure 4.8. Combined aleatoric uncertainty spatial comparison of the Bayesian models.

The spatial aleatoric uncertainty comparison reinforces the previous analysis while providing additional insight into the specific behavior of the different Bayesian models. Specifically, the non KL annealing models express markedly less spatial aleatoric uncertainty than the KL annealing models (Figure 4.8). Next, the KL annealing flipout and reparameterization models express different spatial aleatoric uncertainty. The reparameterization model expresses more aleatoric uncertainty in the northern and southwestern regions of the plot. Finally, the MC dropout model expresses the most aleatoric uncertainty overall, with slightly more aleatoric uncertainty present in all regions of the plot than the reparameterization model.

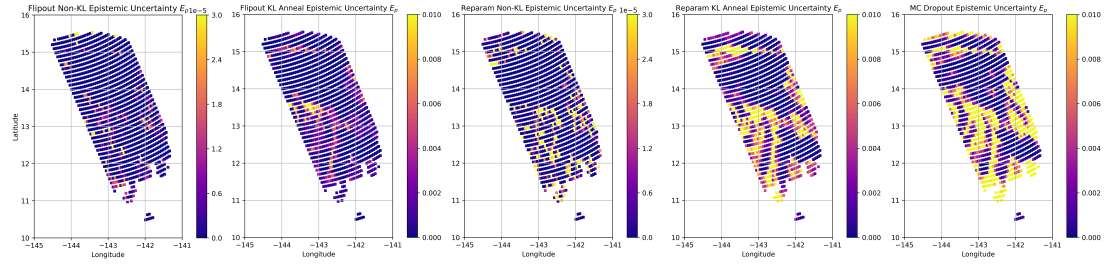


Figure 4.9. Combined epistemic uncertainty spatial comparison of the Bayesian models.

Figure 4.9 reinforces previous analysis that shows the lack of uncertainty quality that is present in the non KL annealing models. Additionally, this spatial comparison allows a more detailed analysis of the difference in the quality of the epistemic uncertainty expressed by the

KL annealing and MC dropout models. Notably, the KL annealing flipout model expresses the lowest amount of epistemic uncertainty and is the most certain in its predictions. Next, the reparameterization model expresses a greater amount of epistemic uncertainty in both magnitude and spatial distribution than the flipout model.

This marks a notable difference in the expressed uncertainties between the KL annealing flipout and reparameterization models. While the reparameterization model is congruent with the MC dropout model (which expresses the most uncertainty overall), the flipout model lacks epistemic uncertainty in regions where the other two model possess it. This suggests that the flipout model expresses a lower amount of epistemic uncertainty than the reparameterization and Monte Carlo Dropout models. This makes sense given the more precise calculations that take place with the flipout method, as it calculates a more precise result that contains less uncertainty by design than the reparameterization and MC dropout methods, but this spatial analysis, when taken in context with Tables 4.9 and 4.10, confirms the mathematical theory that the flipout model expresses a lower amount of uncertainty than other Bayesian modalities in both a spatial and magnitude context.

CHAPTER 5:
Conclusions and Future Work

To review, the research objectives explored in this work were as follows:

1. Determine if Bayesian CNNs can classify cloud convective classes more accurately than deterministic CNNs and the operational GPROF algorithm.
2. Determine which Bayesian CNN model architectures perform well on multispectral satellite data.
3. Investigate how modeling three different types of uncertainty and filtering by said uncertainties affects the classification performance of the Bayesian models.
4. Investigate model performance trade-offs of estimating and filtering by the three uncertainty types.
5. Develop novel Bayesian Model benchmarks for use on large scale datasets and evaluate the Bayesian model performance with them.

In this final chapter, each one of the research objectives will be investigated in order, and conclusions for each will be stated based upon the research and results conducted earlier in this work. Finally, a future work section will detail additional research questions developed from this thesis.

5.1 Conclusions

The first research objective was met by the results of this work. Bayesian model performance met or exceeded deterministic model performance in all metrics on the unfiltered test dataset, and far exceeded the operational GPROF algorithm in all metrics. When the test dataset predictions were filtered to remove samples with high aleatoric uncertainty, the Bayesian models achieved, at worst, a 50 percent error reduction when compared to the deterministic model. The optimal test dataset results were achieved by the flipout and reparameterization models, which achieved an 80 percent error reduction over the deterministic model. All Bayesian modalities expressed an auROC score within the 0.97-0.99 range, which is close to the maximum score of 1.0. Furthermore, the increased computational complexity of these models is a reasonable trade-off when viewed in the context of the performance advantages

and the ability to identify regions of model (epistemic) uncertainty and dataset (aleatoric) uncertainty. These results confirm that Bayesian models possess performance on large, operational datasets that can exceed the performance of state-of-the-art deterministic neural networks.

Next, all Bayesian modalities performed well on classification with the multi-spectral satellite dataset. The conclusion to this research objective can be further refined once performance on OOD data is considered. The KL annealing and MC dropout modalities achieved superior performance metrics on the OOD dataset than the non KL annealing models, and expressed increased ability to generalize to new, unseen OOD data than the non KL annealing models. The best performing model overall was the MC dropout model due to superior quantification and quality of uncertainty, performance on the test and OOD dataset, and reduced computational complexity compared to the other Bayesian models. The next best performing modalities were the KL annealing reparameterization and KL annealing flipout Bayesian models. While performance metrics for those models were similar to the MC dropout model on the testing and serving datasets, their increased computational complexity may limit their utility in certain applications, and the objective quality of their uncertainties, evidenced by the filtering results in Tables 4.8 - 4.10 and spatial results in Figures 4.7 - 4.9, was less than that of the MC dropout model. Finally, the non KL annealing Bayesian models performed the worst out of the Bayesian models. Their reduced computational complexity compared to their KL annealing counterparts does not make up for the lack of quality of their uncertainties, and these models overfit to the training data as evidenced by their comparatively reduced performance on the OOD data.

The third research objective to investigate how filtering by the three uncertainty types affected Bayesian model performance was met. Filtering by all three types of uncertainty greatly improved the performance of all of the Bayesian models. On the test dataset, filtering by each uncertainty type successfully increased the model performance by similar amounts. On the OOD serving dataset on the region of interest, however, filtering by aleatoric uncertainty provided an increased performance improvement than by filtering with the other uncertainty types.

Next, the research objective to investigate the performance trade-offs of estimating and filtering by the different uncertainty types was also met. To begin, all of the Bayesian

models expressed very low epistemic uncertainty values. This shows that all of the models were highly accurate and confident in their predictions. The KL annealing and MC dropout Bayesian models also expressed higher epistemic uncertainty values on the OOD serving dataset on the region of interest. This result confirms, on a far larger and operational dataset, past research that shows increased epistemic uncertainty values on OOD data. On the other hand, all of the Bayesian models expressed aleatoric uncertainty values that were several orders of magnitude greater than the epistemic uncertainty.

Furthermore, as shown on the spatial uncertainty plots, these aleatoric uncertainties are strongly correlated with regions of epistemic uncertainty. This confirms that Bayesian models can be used to screen noisy data samples from prediction sets, as data with high aleatoric uncertainty is not ideal for prediction due to noise in the data causing epistemic uncertainty. Filtering for epistemic uncertainty that is not tied to aleatoric uncertainty can be used to identify data that should be included in a future dataset to improve the model performance, since data with high epistemic uncertainty plus low aleatoric uncertainty presents the greatest opportunity for model information gain.

Next, the epistemic uncertainty maps detail where the KL annealing and MC Dropout Bayesian models are producing incorrect classifications. When the spatial predictions disagree with the DPR true labels, the epistemic uncertainty identifies the incorrect classifications. This phenomenon can be seen most clearly in the southwest, northern, and eastern portions of Figures 4.2, 4.4, and 4.6 where the KL annealing and MC Dropout Bayesian models incorrectly classify convection. Both the Bayesian and GPROF models failed to properly classify these regions, but the epistemic uncertainty map was able to recover the correct classifications, whereas the GPROF and deterministic models do not provide a way to recover the correct classifications.

Crucially, the chief advantages of filtering data based upon uncertainty are confirmed in this work. Past research in this area utilized small datasets that were curated for use by Bayesian models, whereas this work generalizes this approach to large scale datasets fit for operational use. By using a large datasets, Bayesian models were able to be trained to be highly accurate and specific in their uncertainty quantification, allowing potential errors in data collection and processing to be identified while identifying sections of data that should be pursued for further representation in the training data in order to improve model

performance.

Finally, benchmarks were developed in this work for use in future research. First, novel methods of quantifying and filtering by different uncertainty types were developed for use on large datasets in this work. Second, uncertainty histogram and CDF presentations were developed that allow for visual representation of the variance and scale of calculated uncertainties. Third, methods were developed to calculate the average uncertainty of a given type across an entire prediction set and to compare the computational and memory cost of each model. Finally, a novel method of spatially plotting model uncertainties was developed that allows a researcher to understand where, and at what scale, uncertainties occur in model predictions. These benchmarks will allow future researchers to compare and contrast Bayesian model performance across large datasets in a reproducible manner.

In conclusion, Bayesian deep learning methods express performance characteristics on large-scale datasets that are compelling for potential use in operational settings. Bayesian models are shown to outperform deterministic models and operational algorithms while producing uncertainties that can be used to further inform dataset improvement and prediction decisions at a marginal computational cost increase over conventional methods.

5.2 Future Work

To conclude, several additional areas of inquiry were raised during the writing of this work. These areas of inquiry are presented as research objectives in the following list:

1. Develop and benchmark Bayesian CNNs on PMW GPM data that is observed over land and compare performance against deterministic CNNs and the operational GPROF algorithm.
2. Develop and benchmark Bayesian CNNs on a large scale, multispectral satellite meteorological dataset on a regression task, such as rainfall prediction.
3. Investigate the potential relationship between expressed uncertainties by Bayesian CNNs on PMW GPM data that seem to identify storm structures that are not identified by other methods.
4. Develop additional uncertainty screening methods for use in constructing refined datasets for future Bayesian model training.

List of References

[1] M. Pritt and G. Chern, "Satellite image classification with deep learning," *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pp. 1–7, 2017.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, p. 84–90, May 2017. Available: https://doi.org/10.1145/3065386

[3] M. Elhoseiny, S. Huang, and A. Elgammal, "Weather classification with deep convolutional neural networks," 09 2015.

[4] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" *CoRR*, vol. abs/1703.04977, 2017. Available: http://arxiv.org/abs/1703.04977

[5] V. Petković, M. Orescanin, P. Kirstetter, C. Kummerow, and R. Ferraro, "Enhancing pmw satellite precipitation estimation: Detecting convective class," *Journal of Atmospheric and Oceanic Technology*, vol. 36, no. 12, pp. 2349 – 2363, 2019. Available: https://journals.ametsoc.org/view/journals/atot/36/12/jtech-d-19-0008.1.xml

[6] P. M. H. Orescanin, Petkovi´c, "Bayesian deep learning for passive microwave precipitation type detection," *IEEE Geoscience and Remote Sensing Letters*, p. 5, 2021.

[7] A. Filos, S. Farquhar, A. N. Gomez, T. G. Rudner, Z. Kenton, L. Smith, M. Alizadeh, A. de Kroon, and Y. Gal, "A systematic comparison of bayesian deep learning robustness in diabetic retinopathy tasks," *arXiv preprint arXiv:1912.10481*, 2019.

[8] J. M. Hernández-Lobato and R. P. Adams, "Probabilistic backpropagation for scalable learning of Bayesian Neural Networks," 2015.

[9] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.

[10] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, Y. Lechevallier and G. Saporta, Eds. Physica-Verlag HD, pp. 177–186.

[11] F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer, "Efficient deep network architectures for fast chest x-ray tuberculosis screening and visualization," vol. 9,

no. 1, p. 6268, number: 1 Publisher: Nature Publishing Group. Available: https://www.nature.com/articles/s41598-019-42557-4

[12] O. Duerr, B. Sick, and E. Murina, *Probabilistic Deep Learning: With Python, Keras and TensorFlow Probability*. Manning Publications, 2020. Available: https://books.google.com/books?id=-bYCEAAAQBAJ

[13] J. Chen, Y. Li, X. Wu, Y. Liang, and S. Jha, "Robust out-of-distribution detection for neural networks," 2020.

[14] S. Thakur, "The very basics of Bayesian Neural Networks," 2018. Available: "https://sanjaykthakur.com/2018/12/05/the-very-basics-of-bayesian-neural-networks/"

[15] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," 2020.

[16] T. Bayes, Rev., "An essay toward solving a problem in the doctrine of chances," *Phil. Trans. Roy. Soc. Lond.*, vol. 53, pp. 370–418, 1764.

[17] G. E. Hinton and D. van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proceedings of the Sixth Annual Conference on Computational Learning Theory* (COLT '93). New York, NY, USA: Association for Computing Machinery, 1993, p. 5–13. Available: https://doi.org/10.1145/168304.168306

[18] T. J. L. S. Michael Jordan, Zoubin Ghahramani, "An introduction to variational methods for graphical models," p. 183–233, 1999.

[19] T. Broderick, "Variational Bayes and beyond: Bayesian inference for big data," presented at 35th International Conference on Machine Learning, 2018.

[20] M. Blei, Kucukelbir, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, pp. 859–877, May 2018.

[21] E. Khan, "Kullback-Leibler proximal variational inference," *Advances in Neural Information Processing Systems (NIPS)*, vol. 2015, 2015.

[22] X. Yang, "Understanding the variational lower bound," reading note published online, 2017.

[23] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, vol. 14, no. 4, pp. 1303–1347, 2013. Available: http://jmlr.org/papers/v14/hoffman13a.html

[24] H. Wang, X. Shi, and D.-Y. Yeung, "Relational stacked denoising autoencoder for tag recommendation," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (AAAI'15). AAAI Press, 2015, p. 3052–3058.

[25] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," 2015.

[26] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2* (NIPS'15). Cambridge, MA, USA: MIT Press, 2015, p. 2575–2583.

[27] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Appendix," *arXiv e-prints*, p. arXiv:1506.02157, June 2015.

[28] Y. Wen, P. Vicol, J. Ba, D. Tran, and R. B. Grosse, "Flipout: Efficient pseudo-independent weight perturbations on mini-batches," *CoRR*, vol. abs/1803.04386, 2018. Available: http://arxiv.org/abs/1803.04386

[29] F. Laumann, "Bayesian convolutional neural networks with bayes by backprop," 2018. Available: https://medium.com/neuralspace/bayesian-convolutional-neural-networks-with-bayes-by-backprop-c84dcaaf086e

[30] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with Bernoulli approximate variational inference," 2016.

[31] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik, "Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation," *Computational Statistics Data Analysis*, vol. 142, p. 106816, 2020. Available: https://www.sciencedirect.com/science/article/pii/S016794731930163X

[32] H. J. Hortúa, R. Volpi, D. Marinelli, and L. Malagò, "Parameter estimation for the cosmic microwave background with bayesian neural networks," *Physical Review D*, vol. 102, no. 10, Nov 2020. Available: http://dx.doi.org/10.1103/PhysRevD.102.103509

[33] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," *CoRR*, vol. abs/1705.07115, 2017. Available: http://arxiv.org/abs/1705.07115

[34] R. Feng, N. Balling, D. Grana, J. Dramsch, and T. Hansen, "Bayesian convolutional neural networks for seismic facies classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, pp. 1–8, 01 2021.

[35] B. Venkatesh and J. J. Thiagarajan, "Heteroscedastic calibration of uncertainty estimators in deep learning," 2019. Available: https://arxiv.org/abs/1910.14179

[36] J. Zeng, A. Lesnikowski, and J. M. Alvarez, "The relevance of Bayesian layer positioning to model uncertainty in deep bayesian active learning," 2018. Available: https://arxiv.org/abs/1811.12535

[37] A. Nandar, "Bayesian network probability model for weather prediction," *2009 International Conference on the Current Trends in Information Technology (CTIT)*, pp. 1–5, 2009.

[38] O. Aydin and S. R. Shrestha, "Bayesian deep learning for extreme weather event forecasting in a changing climate," in *AGU Fall Meeting Abstracts*, Dec. 2018, vol. 2018, pp. IN52A–06.

[39] Y. Liu, J. Attema, and W. Hazeleger, "Exploring Bayesian deep learning for weather forecasting with the Lorenz 84 system," Sep. 2020, ECMWF-ESA Workshop on Machine Learning for Earth System Observation and Prediction (October 5-8th, 2020). Available: https://doi.org/10.5281/zenodo.4146850

[40] H. Ren, X. Yu, L. Bruzzone, Y. Zhang, L. Zou, and X. Wang, "A Bayesian approach to active self-paced deep learning for SAR automatic target recognition," *IEEE Geoscience and Remote Sensing Letters*, 2020.

[41] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new Bayesian approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 11, pp. 6440–6461, 2018.

[42] D. W. Draper, D. A. Newell, F. J. Wentz, S. Krimchansky, and G. M. Skofronick-Jackson, "The global precipitation measurement (GPM) microwave imager (GMI): Instrument overview and early on-orbit performance," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 7, pp. 3452–3462, 2015.

[43] Iguchi and coauthors, "GPM/DPR level-2algorithm theoretical basis document," *JAXA–NASA Tech.Rep.*, p. 81, 2017.

[44] C. Kummerow, D. Randel, M. Kulie, N.-Y. Wang, R. Ferraro, s. Munchak, and V. Petkovic, "The evolution of the Goddard profiling algorithm to a fully parametric scheme," *Journal of Atmospheric and Oceanic Technology*, vol. 32, p. 150813123809002, 08 2015.

[45] S. W. Powell, R. A. Houze Jr, and S. R. Brodzik, "Rainfall-type categorization of radar echoes using polar coordinate reflectivity data," *Journal of Atmospheric and Oceanic Technology*, vol. 33, no. 3, pp. 523–538, 2016.

[46] J. Awaka, M. Le, V. Chandrasekar, N. Yoshida, T. Higashiuwatoko, T. Kubota, and T. Iguchi, "Rain type classification algorithm module for GPM dual-frequency precipitation radar," *Journal of Atmospheric and Oceanic Technology*, vol. 33, no. 9, pp. 1887 – 1898, 2016.

[47] Y. Choi and S. Kim, "Rain-type classification from microwave satellite observations using deep neural network segmentation," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2020.

[48] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT press Cambridge, 2016, vol. 1, no. 2.

[49] J. Paisley, D. Blei, and M. Jordan, "Variational Bayesian inference with stochastic search," 2012. Available: https://arxiv.org/abs/1206.6430

[50] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2014. Available: https://arxiv.org/abs/1312.6114

[51] T. A. Developers, "tfp.layers.DenseReparameterization," 2021. Available: https://www.tensorflow.org/probability/api_docs/python/tfp/layers/DenseReparameterization

[52] T. A. Developers, "tfp.layers.DenseFlipout," 2021. Available: https://www.tensorflow.org/probability/api_docs/python/tfp/layers/DenseFlipout

[53] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. Available: http://jmlr.org/papers/v15/srivastava14a.html

[54] S. Park and N. Kwak, "Analysis on the dropout effect in convolutional neural networks," in *Computer Vision – ACCV 2016*, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds. Cham: Springer International Publishing, 2017, pp. 189–204.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[57] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, pp. 107–116, 04 1998.

[58] D. Tran, M. Dusenberry, M. van der Wilk, and D. Hafner, "Bayesian layers: A module for neural network uncertainty," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 660–14 672.

[59] "Aleatory or epistemic? does it matter?" *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009, risk Acceptance and Risk Communication.

[60] Y. Li, C. Wei, and T. Ma, "Towards explaining the regularization effect of initial large learning rate in training neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 674–11 685.

[61] A. M. Webb, "Cross entropy and log likelihood," May 2017. Available: http://www.awebb.info/probability/2017/05/18/cross-entropy-and-log-likelihood.html

[62] T. A. Developers, "tfp.distributions.kldivergence," 2021. Available: https://www.tensorflow.org/probability/api_docs/python/tfp/distributions/kl_divergence

[63] L. Zhu, "tfp resnet tutorial," 2019. Available: https://github.com/zhulingchen/tfp-tutorial/blob/master/tfp_resnet.py

[64] G. Huffman, R. Adler, P. Arkin, A. Chang, R. Ferraro, A. Gruber, J. Janowiak, A. McNab, B. Rudolf, and U. Schneider, "The global precipitation climatology project (gpcp) combined precipitation data set," *Bulletin of the American Meteorological Society*, vol. 78, 02 1997.

[65] C. Kidd, J. Tan, P.-E. Kirstetter, and W. Petersen, "Validation of the version 05 level 2 precipitation products from the gpm core observatory and constellation satellite sensors," *Quarterly Journal of the Royal Meteorological Society*, vol. 144, 10 2017.

[66] C. Kummerow, "On the accuracy of the Eddington approximation for radiative transfer in the microwave frequencies," *Journal of Geophysical Research*, vol. 98, pp. 2757–2765, 1993.

[67] C. Kummerow, W. Olson, and L. Giglio, "A simplified scheme for obtaining precipitation and vertical hydrometeor profiles from passive microwave sensors," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 5, pp. 1213–1232, 1996.

[68] scikit-learn developers, "sklearn.metrics.accuracy," 2020. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html#sklearn.metrics.accuracy_score

[69] scikit-learn developers, “sklearn.metrics.roc.score,” 2020. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html#sklearn.metrics.roc_auc_score

[70] K. Miyasato, “Classification report: Precision, recall, f1-score, accuracy,” 2020. Available: https://medium.com/@kennymiyasato/classification-report-precision-recall-f1-score-accuracy-16a245a437a5

[71] O. P. P. Ortiz, Marsh, “A systematic evaluation of Bayesian deep learning on satellite imagery for classification,” *IEEE*, p. 7, 2021.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California