November 2020

Unsupervised out-of-distribution detection in digital pathology

Thesis for the degree of Master of Science
in Computing Science –Data Science specialization

by Gabriel Raya

Radboud
Universiteit
Nijmegen

Master Thesis Computing Science
Data Science

Radboud University

Unsupervised out-of-distribution detection in digital pathology

Author: Gabriel Raya Rodríguez
Email: gaboraya@gmail.com
Submitted on: November 23, 2020

Supervisor & assessor: Dr. Twan van Laarhoven
Second Supervisor : MS. Jasper Linmans
Second assessor: Prof. dr. Tom Heskes

To my secondary school teacher Agustín Romo Venegas, who introduced me to the wonderful world of Mathematics. Rest in peace.

A mi maestro de secundaria Agustín Romo Venegas, quien me introdujo al maravilloso mundo de las Matemáticas. Descanse en paz.

Acknowledgments

I want to thank my supervisors Twan and Jasper, for their constant feedback and guidance throughout this project. Their supervision and collaboration allowed me to pursue my interests and broaden my research experience with endless ideas to explore. It has been a pleasure working with them.

I also want to thank my friends for all the times we have shared and for making me feel at home while away from home.

Finalmente, y sin lugar a dudas, quisiera a gradecer a toda mi familia por apoyarme de manera incondicional y creer en mis sueños en todo momento.

Nijmegen, The Netherlands. November 2020.

"In the beginning there was nothing, nothing but the silence of infinite darkness." — Noah

Abstract

Despite huge success, deep neural networks are still not reliable enough to work under critical conditions. As they are susceptible to data drawn from a distribution different to that of the training data, mistakenly assigning high confident predictions to such out-of-distribution (OOD) inputs. A natural approach to overcome this issue is to use a deep generative model (DGM) to reconstruct the probability density of the training data and use the likelihood estimates to find whether a new data point x^* comes from the training data distribution. However, in this thesis, we found that DGMs are susceptible to OOD inputs, as their likelihood estimates are poorly calibrated. This happened when we tested a variational autoencoder (VAE) over several image data sets. To mitigate this problem, we propose using a Bayesian Variational Autoencoder (BVAE) and an Ensemble of VAEs to robustify the OOD detection score by estimating the *epistemic* uncertainty of the likelihood model.
While experimental results show improvements over VAEs on simple tasks, these methods do not scale to more complex tasks such as digital pathology. Perhaps surprisingly, in our setting, we found that BVAEs and VAEs do not account directly for the typical set of the data distribution. Therefore, making direct use of the likelihood estimates is not enough to sufficiently model in-distribution inputs. For this reason, we further investigate the problem of typicality in VAEs and use the density of states estimator (DoSE) to measure the frequency of various model statistics. We empirically demonstrate how DoSE outperforms the former approaches and show how it can be used in digital pathology when training a model solely on healthy tissue to detect tumor tissue as OOD samples yielding an AUROC score of 0.84.

Contents

Nomenclature

Acronyms / Abbreviations

AI	Artificial Intelligence
OOD	Out-of-distribution
InD	In-distribution
NN	Neural Network
DNN	Deep Neural Network
BNN	Bayesian Neural Network
MCMC	Markov Chain Monte Carlo
SGMCMC	Stochastic Gradient Markov Chain Monte Carlo
KL	Kullback-Leibler
ELBO	Evidence lower bound
DGM	Deep Generative Models
VAE	Variational Auto Encoder
BVAE	Bayesian Variational Auto Encoder
RP	Random Prior
i.i.d	Independent and identically distributed

Symbol / Definition

$q_\phi(z\|x)$	Estimated posterior probability function, also known as **probabilistic encoder**, **inference/recognition network**
$p_\theta(x\|z)$	Likelihood of the generating true data given the latent vector z, also known as **probabilistic decoder**
$p(x)$	probability density function
$z \sim p(z)$	random variable z sample from a density function $p(z)$
$\mathrm{Var}(x)$	variance of a random variable x
σ_x	standard deviation of a random variable x
$\mathrm{H}(x)$	Entropy of a random variable x
$\mathrm{D}_{KL}(q\|\|p)$	KL-divergence of distribution q and p

Chapter 1

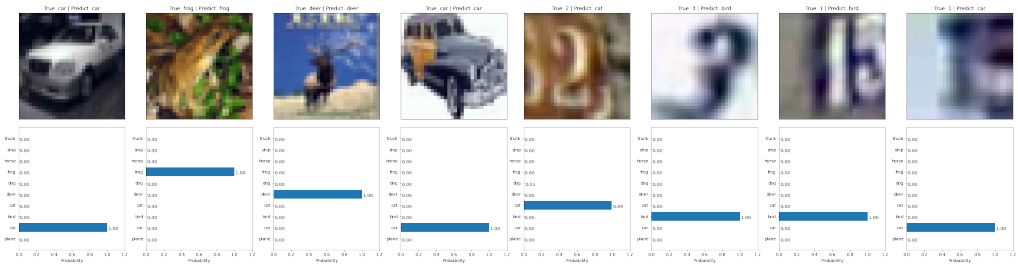
Introduction

> *I can live with doubt and uncertainty. I think it's much more interesting to live not knowing than to have answers which might be wrong.*
> *—Richard .P. Feynman*

Over the last few years, deep neural networks (DNNs) have made remarkable progress in many fields, ranging from applied to fundamental research, e.g., physics, biology, health, computer vision, autonomous driving cars, among many others. As a result, complex decision systems such as autonomous cars, medical diagnosis, and cybersecurity use this technology. Despite this considerable success, DNNs are still not reliable enough to work under dataset shift, a common phenomenon present in machine learning when the conditions in which the model is trained are entirely different to those in which the model is used, representing a critical problem to AI safety. As an example, let us consider the case of a pathologist who makes use of a computer-aided diagnosis (CAD) system to identify whether a patient has a tumor. The CAD system, powered by a DNN, tested and verified under controlled conditions, is then deployed 'in the wild', and therefore susceptible to different sources of uncertainty that were not present at training time. Consequently, the system could give erroneous medical diagnosis with high confidence, possibly resulting in patient neglect that could lead to fatal results. Similarly, an autonomous car could present undesired behaviors under inputs that the DNN is unfamiliar with.

To illustrate this failure, let's consider the example shown in Figure 1.1. A classifier is trained on samples $X = \{x_1, ..., x_n\}$ from the CIFAR10 [Krizhevsky, 2009] dataset. We assume that these samples are drawn independently from the same distribution $p(X)$. Since the model is only exposed to samples from $p(X)$ at training time, these are also known as in-distribution samples. By adjusting the configuration on the model parameters, the model learns to classify the ten different classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck) with a 94% accuracy on the test set. What if a user decides to test the model with a new input x^* from a distribution $q(X) \neq p(X)$, e.g., an image from the SVHN dataset?. These samples are also known as *out-of-distribution* (OOD) samples since they come from a distribution different than the training distribution. Figure 1.1 shows how the model not only assigns incorrect predictions to OOD samples, but it does so with high probability. For example, Figure 1.1b shows how an OOD input x^*, with the corresponding label '3', is incorrectly classified as 'bird' with maximum probability. This failure in DNNs is known as the out-of-distribution problem. OOD detection allows us to measure how a model generalizes to domain shift, detecting if the *model knows what it knows* [Lakshminarayanan et al., 2017], a fundamental assessment to safety-critical applications.

This vulnerability of DNNs has motivated the development of several approaches to detect such OOD samples by estimating predictive uncertainty estimates in supervised learning settings [Gal and Ghahramani, 2016, Lakshminarayanan et al., 2017]. However, these supervised learning OOD methods are task-specific, requiring labeled data, which is expensive to obtain most of the time. Some supervised methods may also require $q(X)$ to be specified at training time, which is not always possible, e.g., when



(a) Predictions for in-distribution data. (b) Predictions for OOD data.

Figure 1.1: DNNs fails to detect OOD inputs. A model trained on samples from the CIFAR10 dataset classifies in-distribution inputs from the test set with a 94% accuracy. However, when the model is evaluated on inputs from the SVHN dataset, the model incorrectly classifies these OOD inputs with high probability. The **bottom row** shows the probabilities predicted by the model for a given new input x^* (**top row**).

anomalous data is rare. On the other hand, unsupervised learning OOD distribution methods are *task agnostic* techniques that can leverage large amounts of unlabelled data to reconstruct the data density under the model. A possible advantage of unsupervised methods over discriminative models is that by modeling a density instead of just class boundaries, one could capture more relevant information required for effective OOD detection. Therefore, the unsupervised approach to detect such OOD samples is a promising avenue in digital pathology, where annotated training data is expensive and abnormal samples are challenging to obtain.
In this thesis, we study the problem of detecting out-of-distribution samples in a real-world digital pathology task using solely unlabeled data as in-distribution samples. Specifically, we want to train a model using only normal class data, data from the 'healthy' class, and detect tumor tissue or anomaly as out-of-distribution data. Therefore, we research the following question:

How can we effectively determine whether a new test input x^* was drawn from the training distribution $p(X)$ or from other distribution $q(X) \neq p(X)$ using only unlabeled data?.

To answer this question, we first started investigating how **random prior networks**, a method that has been successfully used to estimate high-quality uncertainty estimates for *regression tasks* [Osband et al., 2018], could be used to detect OOD samples on image data. Empirically, we show that while this approach works well for regression tasks, it does not scale to image data sets, such as CIFAR10 and PCam [Veeling et al., 2018], contrary to what [Ciosek et al., 2020] recently showed. We pointed out that [Ciosek et al., 2020] mistakenly evaluated their method to detect OOD samples by testing OOD scores against the train set rather than the test set. As a result, we show their approach does not generalize well to unseen samples, and consequently, their experiments do not support their findings. For this reason, we further explore the use of likelihood-based deep generative models (DGMs) (e.g., variational autoencoders, autoregressive models, or flow-based models). Specifically, we investigate whether variational autoencoders (VAE) [Kingma and Welling, 2014, Rezende et al., 2014] can be used for anomaly detection. Likelihood-based DGMs are commonly employed to reconstruct the data distribution $p(X)$ by maximizing the likelihood $p(X|\theta)$ under the model parameters θ. One can use a one-sided threshold on the model log-likelihoods as a decision rule to identify OOD samples from in-distribution samples. This approach resides in the idea that the likelihood under the model parameters $p(X|\theta)$ represents the 'probability

of the data under the model parameters,' explaining how well a specific configuration of the model parameters explains the data and, therefore, in-distribution data should have *higher* likelihoods than out-distribution data. Historically, this approach has remained to work on low dimensional data, as shown by [Bishop, 1994]. However, we found that this is not the case for higher-dimensional data, such as images. We found that VAEs often assign higher likelihoods to OOD samples than in-distribution samples, as recently noted by [Nalisnick et al., 2018], who empirically showed this failure on flow-based models, VAEs, and PixelCNNs. Not surprisingly, the lack of robustness of DGMs to OOD inputs could be explained by the *deterministic* behavior of deep neural networks, providing only a single point estimate for the model parameters without confident bounds, and therefore not acknowledging the possibility that other plausible models could have explained the data as well. To robustify this phenomenon, we study the uncertainty present in a variational autoencoder's model parameters, also known as *epistemic* uncertainty. We take both Bayesian and non-Bayesian approaches to model this source of uncertainty. The Bayesian approach leads to a Bayesian Variational Autoencoder (BVAE) [Daxberger and Hernández-Lobato, 2019], while the non-Bayesian approach leads to an ensemble of VAEs [Lakshminarayanan et al., 2017]. BVAEs account for model uncertainty by placing a posterior distribution over the decoder and encoder parameters. Computing the full posterior distribution over the model parameters requires Bayesian inference, which has been challenging to use in deep learning due to its high computational cost. Instead, we used a scaled adaptive version of Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) [Chen et al., 2014, Springenberg et al., 2016] that allows sampling from the posterior distribution in a Metropolis Hasting framework using a *mini-batch*-based optimization scheme. The resulting model provides samples from the posterior distribution that we used as a set of hypotheses to estimate agreements' variability to compute metrics to detect potential OOD samples. The resulting samples can be seen as an ensemble of VAEs. For this reason, we also implement an ensemble of VAEs [Lakshminarayanan et al., 2017] to compare the effectiveness of both Bayesian and non-Bayesian approaches to detect OOD samples.

Our empirical results show that the BVAE provides more robust scores than the traditional VAE when the VAE likelihoods are poorly calibrated. However, in some cases, we observe that the variability across the BVAE posterior samples for in-distribution is similar to that for OOD inputs, and so the information of a new OOD sample x^* is not enough to be detected as a potential OOD input. We report similar results to BVAEs with an ensemble of VAEs.

We close this work by discussing the problem of typicality in VAEs and illustrate that OOD methods that solely rely on the poorly calibrated likelihoods are not robust enough. A possible reason is that VAEs, trained under the Maximum Likelihood principle, do not maximize the typical set due to high dimensions' norm sensitivity. We implement the density of estates estimator (DoSE) [Morningstar et al., 2020] to measure the frequency of various model statistics using a one-class Support Vector Machine (SVM) or Kernel Density Estimator (KDE) to estimate the 'probability of the model probability.' We empirically demonstrate how DoSE outperforms the former approaches and show how it can be applied to digital pathology when training a model solely on healthy tissue to detect tumor tissue as OOD samples. All the methods are first validated on standard image dataset benchmarks, e.g., MNIST, Fashion-MNIST, CIFAR10, and SVHN.

1.1 Thesis contributions

The main contributions of this thesis are as follows:

- A review of current unsupervised methods for out-of-distribution detection with a particular focus on VAEs.
- We empirically demonstrate that *random prior networks* do not scale to large datasets for OOD detection and point out that [Ciosek et al., 2020] mistakenly evaluated their method to detect OOD inputs.
- We empirically show the now well-known problem that VAEs sometimes assign higher likelihoods to OOD samples and provide an in-depth analysis of this failure.
- We describe how to estimate *epistemic uncertainty* in DGMs using Bayesian inference and deep ensembles with automatic gradient differentiation tools.
- We provide an analysis of the problem of typicality in VAEs and show how DoSE can be applied to digital pathology yielding an AUROC score of 0.84 on a model trained solely on healthy tissue to detect tumor tissue as OOD samples.

1.2 Thesis outline

This thesis discusses the fundamental problem of out-of-distribution detection in an unsupervised setting. Even though the motivating application is in digital pathology, the methods discussed here can be used in a broader range of applications.

The thesis is structured as follows :

- **Chapter 2** provides the theory necessary to understand this work.
- **Chapter 3** presents a brief introduction to Digital Pathology and introduces the digital pathology data that used in the rest of this work.
- **Chapter 4** briefly describes *random prior networks* [Ciosek et al., 2020], presents the empirical results obtained when used for OOD detection, and demonstrates how [Ciosek et al., 2020] mistakenly evaluated their method to detect OOD inputs, and consequently, their experiments do not support their work.
- **Chapter 5** shows how VAEs *sometimes* assign higher likelihoods to OOD samples. It presents the results obtained by BVAEs and an ensemble of VAEs, and it experimentally shows the problem of typicality by using DoSE.
- **Chapter 6** presents the results obtained in this work and suggests possible future avenues of research.

1.3 Code

The code is available online at **https://github.com/gabrielraya/uncertainty-estimation**.

Chapter 2

Bayesian Modeling in Deep Learning

Bayesian modelling allows to express the degree of plausibility over the estimated functions $\hat{\theta}$ using the framework of probability theory [Depeweg, 2019].

This section presents the majority of the theory this work builds upon and is necessary to understand the later chapters. Section 2.1 and 2.2 discuss the frequentist and Bayesian approaches of deep neural networks. Section 2.3 presents a brief overview of modern approaches to approximate inference. Section 2.4 discusses the sources of uncertainty present in deep learning and introduces some standard metrics to measure uncertainty. Section 2.5 and 2.6 introduce Variational autoencoders and Bayesian Variational autoencoders, respectively. Section 2.7 presents the problem of typicality in deep generative models.

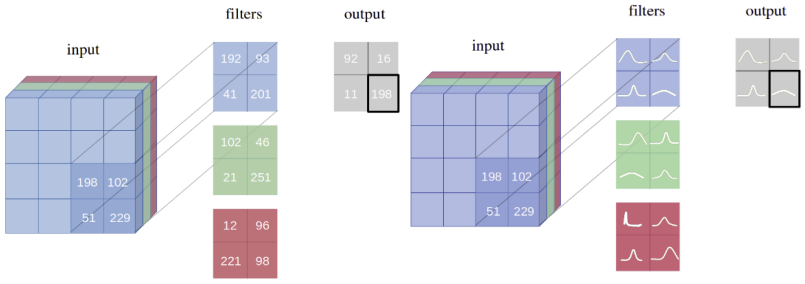
2.1 Single point estimates in neural networks

We start this section by defining the *classical statistical problem of parameter estimation*. Given i.i.d data $X = \{x_1, ..., x_n\}$ sampled from a parent distribution $p(X|\theta)$, the goal of parameter estimation is to find θ. For this, we rely on methods of parameter estimation. There are two classical approaches from the point of view of statistics: *frequentist* and *Bayesian*.
In the **frequentist approach**, the most common used method of parameter estimation is maximum likelihood estimation (MLE). MLE aims to maximize the likelihood function $\mathcal{L}(X;\theta)$. The likelihood is simply the joint probability density function (p.d.f) for our n measurements X given θ. Because the x_i are assumed to be independent, the joint p.d.f becomes the product of p.d.f's for the individual x_i, and therefore the likelihood can be re-expressed as the sum of the logs by the log rule as shown in equation 2.1. A parameter $\hat{\theta}_{MLE}$ is then estimated under MLE by equation 2.2.

$$\mathcal{L}(X;\theta) = \mathcal{L}(x_1, x_2, ..., x_n;\theta) = \prod_{i=1}^{n} f_i(x_i;\theta) = \sum_{i=1}^{n} \log f_i(x_i;\theta) \tag{2.1}$$

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}}\ p(x|\theta) = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{n} p(x_i|\theta) = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n} \log p(x_i|\theta) \tag{2.2}$$

We emphasize the fact that MLE provides a **single point estimate** $\hat{\theta}_{MLE}$ by using the subscript $_{MLE}$. Deep neural networks learn such single point estimate $\hat{\theta}_{MLE}$, typically by a gradient descent method (i.e., backpropagation). Therefore DNNs are *deterministic* functions that directly do not take into account *epistemic/model uncertainty* since this single set of weights $\hat{\theta}_{MLE}$ does not guarantee that our model will generalize well



(a) CNN (b) Bayesian CNN

Figure 2.1: Comparing single point estimates with full distribution estimates in deep learning. (**Left**) A Convolutional Network (CNN) with single point-estimates as parameters. (**Right**) A CNN with probability distributions over parameters. (Image taken from [Laumann, 2018]).

to unseen data because there will always be many plausible models that could have explained our data better. This is one of the reasons that MLE leads to generalization issues and undesirable behavior against OOD inputs.

2.2 Full distribution estimates

In contrast to the frequentist approach, the **Bayesian approach estimates a full posterior distribution** $p(\theta|X)$, as a consequence that now θ is a random variable. This randomness in the parameters allow us to incorporate *epistemic uncertainty*. To estimate the full posterior distribution $p(\theta|X)$ we must apply Bayesian theorem :

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)} = \frac{\prod_{i=1}^{n} p(x_i|\theta)p(\theta)}{\int \prod_{i=1}^{n} p(x_i|\theta)p(\theta)d\theta} \tag{2.3}$$

This posterior distribution describes how likely our set of random variables $\theta_1, ..., \theta_n$ is to take on each of its possible states, placing higher probabilities on settings which are more likely to have generated the data. The uncertainty in the model parameters can then be measured as the variability of disagreement among the posterior samples. Therefore, the Bayesian framework brings us a way to measure uncertainty [Murphy, 2012]. To estimate the posterior distribution we must compute the normalization constant (*evidence*) $p(X)$:

$$p(X) = \int p(X|\theta)p(\theta)d\theta \tag{2.4}$$

Inference for a new data point x^* yields to compute the so called *expected likelihood*:

$$p(x^*|X) = \mathbb{E}_{p(\theta|X)}[p(x^*|\theta)] = \int p(x^*|\theta)p(\theta|X)d\theta \tag{2.5}$$

Equation 2.5 translates uncertainty in the model parameters into uncertainty in predictions. Broadly, Bayesian modelling in deep learning can be summarized as follows:

1. **Learning** $\rightarrow p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)}$
2. **Inference** $\rightarrow p(x^*|X) = \int p(x^*|\theta)p(\theta|X)d\theta$

However, doing *inference* in Bayesian modelling requires marginalizing out over all the possible configuration of θ and computing the model evidence, which are intractable since modern neural networks often contain millions of parameters, and therefore, we need to rely on **approximate inference** methods.

2.3 Approximate Inference

The target of Bayesian inference is to estimate the posterior over the model's parameters $p(\theta|X)$ (eq. 2.3), which is generally intractable and therefore **approximate inference** is required. There are two common families of methods used to approximate this: 1) **variational inference** (VI) or 2) **sampling methods** (e.g., MCMC) [Bishop, 2006].

2.3.1 Variational Inference

Variational Inference (VI) [Hinton and van Camp, 1993], also known as Variational Bayes (VB), approximates the posterior distribution over the model parameters $p(\theta|X)$ with a parameterized *variational distribution* $q_\phi(\theta) \in \mathcal{Q}$ from a family of distributions $\mathcal{Q}$ with variational parameters ϕ. The main idea behind VI is to cast the task of finding the posterior distribution into an optimization problem [Blei et al., 2017, Jordan et al., 1999]

$$q^*_\phi(\theta) = \underset{q_\phi \in \mathcal{Q}}{\operatorname{argmin}} \, \mathrm{D}_{KL}[q_\phi(\theta)||p(\theta|X)] = \underset{q}{\operatorname{argmin}} \int q_\phi(\theta) \log \frac{q_\phi(\theta)}{p(\theta|X)} d\theta \tag{2.6}$$

By adjusting the variational parameters ϕ we aim to find a distribution $q^*_\phi(\theta)$ that minimizes the Kullback-Leibler (D_{KL}) divergence with the true posterior $p(\theta|X)$:

$$\begin{aligned} \mathrm{D}_{KL}[q_\phi(\theta)||p(\theta|X)] &= \mathbb{E}_{q_\phi(\theta)}[\log q_\phi(\theta) - \log p(\theta|X)] && (2.7)\\ &= \mathbb{E}_{q_\phi(\theta)}[\log q_\phi(\theta) - \log \frac{p(X|\theta)p(\theta)}{p(X)}] && (2.8)\\ &= \log p(X) + \mathbb{E}_{q_\phi(\theta)}[\log q_\phi(\theta) - \log p(X|\theta) - \log p(\theta)] && (2.9) \end{aligned}$$

To avoid the intractable model evidence $\log p(X)$ (Eq. 2.4), we can derive a bound to $\log p(X)$ on eq. 2.9, by making use of the non-negativity of the D_{KL}. This bound is known as the **Evidence Lower Bound** or **ELBO**:

$$\begin{aligned} \log p(x) \geq ELBO &= \mathbb{E}_{q_\phi(\theta)}[\log p(X|\theta) - \log q_\phi(\theta) + \log p(\theta)] && (2.10)\\ &= \mathbb{E}_{q_\phi(\theta)}[\log p(X|\theta)] - \mathrm{D}_{KL}(q_\phi(\theta)||p(\theta)) && (2.11) \end{aligned}$$

Maximizing the ELBO is equivalent to minimizing the VI objective of equation 2.6, which approximates to maximizing the $\log p(x)$. The ELBO decomposes in two terms: the first term is called as *data term*, it measures how well the model parameters explain the data, and the second term, the KL divergence, sometimes known as the *regularizer* term, penalizes the variational posterior for differing with the prior.
VI is efficient and easy to scale to large data. However, the freedom to choose the variational family of distributions induced bias.

2.3.2 Markov Chain Monte Carlo

Under some circumstances, the model evidence is not required to building a Markov with $p(\theta|X)$ as the stationary distribution. Eq. 2 can be then approximated as:

$$p(x^*|X) = \frac{1}{M} \sum_{m=1}^{M} p(x^*|\theta_m); \quad \theta_m \sim p(\theta|X) \tag{2.12}$$

However, in practice, when working with high dimensional datasets, sampling can be very slow, specially when the true distribution is highly correlated, making traditional MCMC methods unsuitable for deep learning.

Hamiltonian Monte Carlo (HMC)

Hamiltonian Monte Carlo [Duane et al., 1987, Neal, 2012], depicts an analogy to a fictitious dynamical system by defining a Hamiltonian function in terms of the target distribution from which we want to collect samples with an auxiliary variable in a Metropolis-Hasting framework. In this setting, a model's parameter space can be seen as an uneven surface with position variables θ. The probability of a specific configuration of the posterior density is $p(\theta|x) \propto \exp(-U(\theta))$. Therefore, the posterior density is related to the system's potential energy as follows:

$$\log p(\theta|X) \propto \log p(X,\theta) = -U(\theta) \tag{2.13}$$

with *potential energy* given by

$$U(\theta) = -\sum_{x \in X} \log p(x|\theta) - \log p(\theta) = -\log p(X,\theta) \tag{2.14}$$

This implies that at each configuration, the height is inversely related to the probability under the posterior distribution. To sample from $p(\theta|X)$, HMC introduces an auxiliary momentum variable r. This allows to define a proposal joint distribution of (θ, r) and, with it, the Hamiltonian function $H(\theta, r)$

$$log\pi(\theta, r) \propto -U(\theta) - \frac{1}{2} r^T M^{-1} r = -H(\theta, r) \tag{2.15}$$

M is the fictitious mass associated with each parameter. The Hamiltonian function $H(\theta, r)$ measures the total energy of the system, the *potential* and *kinetic* energy. To get samples, HMC simulates the Hamiltonian dynamics:

$$\begin{cases} d\theta = M^{-1} r dt \\ dr = -\nabla U(\theta) dt \end{cases} \tag{2.16}$$

The Hamiltonian dynamics leave π invariant, allowing us to get samples from the target distribution $p(\theta|X)$ by sampling first from $\pi(\theta, r)$ and then discarding the resulting r samples. This allows a more efficient exploration of the parameter space than random walk MCMC proposals. However, in practice, we cannot simulate exactly this fictitious continuous system, and instead, we consider a discrete system. MH steps must be implemented to compensate for the discretization error. A new sample is then saved every fixed number of steps.
However, a well-known limitation of HMC methods is that the required gradient computation $\nabla p(X|\theta)$ to simulate the Hamiltonian dynamics is infeasible to compute to large datasets [Chen et al., 2014]. A limited condition of using HMC in deep learning.

2.3.3 Stochastic Gradient Hamiltonian Monte Carlo (SGHMC)

An obvious approach to scale HMC to large dataset is simply to apply a stochastic gradient modification to HMC by replacing $\nabla U(\theta)$ of Eq. 2.16, which requires iterating over the entire dataset X, by a noisy estimate $\nabla_\theta \tilde{U}(\theta, \tilde{X})$ based on a *minibatch* $\tilde{X} \subset X$ as follows:

$$\nabla_\theta U(\theta, X) \approx \nabla_\theta \tilde{U}(\theta, \tilde{X}) = -\frac{|X|}{|\tilde{X}|} \sum_{x \in \tilde{X}} \nabla_\theta \log p(x|\theta) - \nabla_\theta \log p(\theta) = \nabla_\theta U(\theta, X) + v \tag{2.17}$$

Notably, [Chen et al., 2014] showed that this 'naive' approach no longer leads the desired target distribution as the stationary distribution since the stochastic gradient approximation introduces a noise $v \sim \mathcal{N}(0, 2B\epsilon)$ normally distributed. They proposed

Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) as a scalable variant of HMC with friction term C added to the momentum update from Eq. 2.16. The inclusion of the friction term counteracts the noise v introduced by the stochastic gradient estimates, maintaining the desired target distribution $p(\theta|X)$ as the stationary distribution that results in the following updates:

$$\begin{cases} d\theta = M^{-1}rdt \\ dr = -\nabla\tilde{U}(\theta)dt - CM^{-1}rdt + v'; \quad v' \sim \mathcal{N}(0, 2(C - \hat{B})dt) \end{cases} \tag{2.18}$$

$\hat{B}$ is an estimate of the gradient noise. In practice, the continuous system is transformed to ϵ-discretization, so $dt = \epsilon$. M is often set to the identity matrix $M = I$, $\hat{B} = 0$. C, the friction term is commonly identified as the momentum decay. The authors showed that there is a connection between SGHMC and SGD with momentum by defining $v = \epsilon M^{-1}r$ (the momentum in SGM parlance) and rewriting 2.18 with v in a ϵ-discretization manner as follows:

$$\begin{cases} \Delta\theta = v \\ \Delta v = -\epsilon^2 M^{-1}\nabla\tilde{U}(\theta) - \epsilon M^{-1}Cv + \mathcal{N}(0, 2\epsilon^3 M^{-1}(C - \hat{B})M^{-1}) \end{cases} \tag{2.19}$$

If we define $\eta = \epsilon^2 M^{-1}$, $\alpha = \epsilon M^{-1}C$, $\hat{\beta} = \epsilon M^{-1}\hat{B}$

$$\begin{cases} \Delta\theta = v \\ \Delta v = -\eta\nabla\tilde{U}(\theta) - \alpha v + \mathcal{N}(0, 2(\alpha - \hat{\beta})\eta) \end{cases} \tag{2.20}$$

with learning rate $\eta > 0$, and momentum term $(1 - \alpha)$. When the noise is removed (via $C = \hat{B} = 0$), SGHMC reduces to SGD with momentum, making it suitable to scale to large datasets, and therefore to deep learning. Following these updates 2.18 θ is guaranteed to be distributed according to $p(\theta|X)$. SGHMC marries the efficiencies in parameter space exploration of HMC methods with the computational efficiencies of stochastic SGD-based optimization techniques. However, SGHMC omits the metropolis step as it requires computation over the entire dataset, and therefore presents a bias that grows with the value of the step size.

To alleviate this problem [Springenberg et al., 2016] proposed a more robust variant of SGHMC [Chen et al., 2014] based on a scaled adaptation of SGHMC. This bias is avoided by the burn-in procedure used to adapt its own hyper-parameters during the initial stages of sampling.

In Figure 2.2 we compare SGHMC with Scale adapted SGHMC in a toy example for a bimodal target distribution $p(\theta|X) = \frac{1}{2}\mathcal{N}(x; \mu = \theta_0, \sigma_x = \sqrt{2}) + \frac{1}{2}\mathcal{N}(x; \mu = \theta_0 + \theta_1, \sigma_x = \sqrt{2})$. The data was created using this scheme : $x_i = \sqrt{(2)}\alpha_1 + \theta_0$ if $k < 0.5$ else $x = \sqrt{(2)}\alpha_2 + \theta_0 + \theta_1$, such that $k \sim \mathcal{U}(0, 1)$ and $\alpha_1, \alpha_2 \sim \mathcal{N}(0, 1)$ with $\theta = \{\theta_0 = 0, \theta_1 = 1\}$. We observed how adaptive scale SGHMC cancels out the bias present in SGHMC and at the same time provides a better exploration of the parameter space. We use automatic gradient differentiation tools[1] to implement the samplers. This example motivates the use of **Adaptive Scale SGHMC** to estimates *epistemic uncertainty* in deep learning.

[1] We used the PyTorch implementation from https://github.com/automl/pybnn, who uses a variant of SGHMC to scale the magnitude of the noise used during sampling

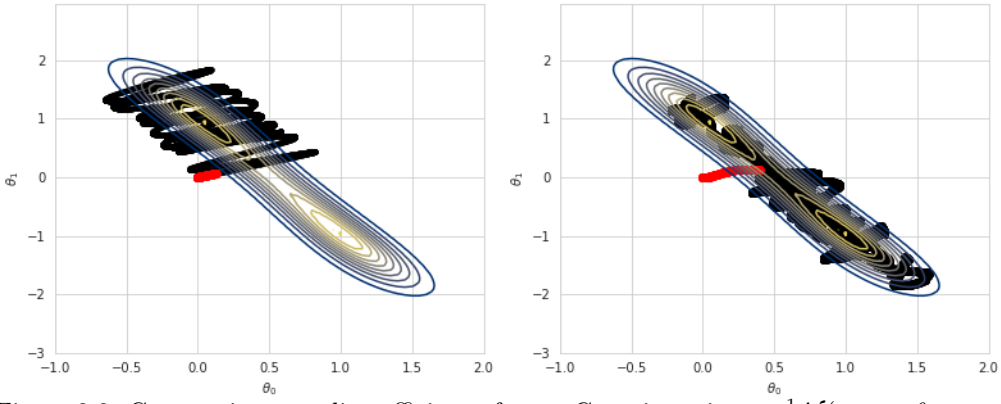


Figure 2.2: Contrasting sampling efficiency from a Gaussian mixture $\frac{1}{2}\mathcal{N}(x; \mu = \theta_0, \sigma_x = \sqrt{2}) + \frac{1}{2}\mathcal{N}(x; \mu = \theta_0 + \theta_1, \sigma_x = \sqrt{2})$ where $\theta_0 = 0$, and $\theta_1 = 1$, using [Left] SGHMC versus [Right] Adaptive scale SGHMC during 1000 steps, with burn-in = 100 (red) and steps after burn-in =9900 (black) with learning rate $\eta = 0.01$, and momentum decay of 0.01. Adaptive scale SGHMC provides a better exploration of the parameter space.

2.4 Uncertainty in deep neural networks

Modelling in deep learning, as in any science, requires making assumptions that allow us to synthesis the complexity of the world. This simplify abstraction is a consequence of lack of information to fully describe the nature of the system or due to the physical limitation of the machine to embedded all the information [Pearl, 1988]. As a result, uncertainty is intrinsically present in our model's predictions. Some possible sources of uncertainty we can consider are as follows:

- There may be inherent noise in the data due to stochastic generative processes or unpredictable variations in the system's performance [Hora, 1996].
- There may be different plausible models that can explain the data.
- Limited knowledge to choose the right model to our task.

The first one is known as irreducible or **aleatoric uncertainty**. The other two are known as *model* or *epistemic uncertainty*. While aleatory uncertainty is irreducible, one can reduce the epistemic uncertainty by gathering more data. In this work, we only consider model uncertainty caused by the uncertainty in the parameters.

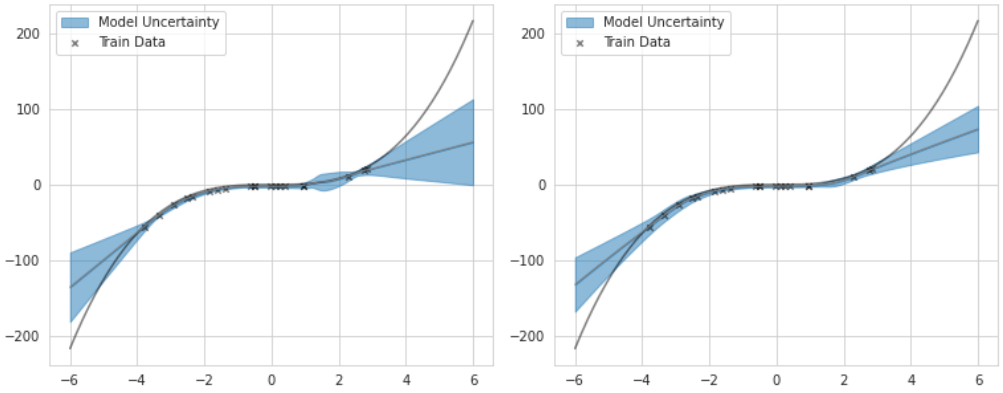
2.4.1 Measuring Epistemic Uncertainty

Bayesian approach

Bayesian modeling has the ability to capture model uncertainty. One way to obtain the uncertainty about the model parameters is as follows:

$$\sigma^2(E_{p(\theta_k|X)}[p(x^*|\theta_k)]); \quad \theta_m \sim p(\theta|X) \tag{2.21}$$

This model uncertainty reflects the variability induced by different models. The variance in the predicted likelihoods captures dissimilarity among explanations given by different parameter configurations.



(a) 50 samples $\sim p(\theta|X)$ (b) 50 Ensembles

Figure 2.3: Contrasting model uncertainty estimation for Bayesian approach (SGHMC) vs Non-Bayesian (Deep ensembles) on a regression toy example. a) Adaptive scale SGHMC taking 50 samples from the posterior. b) 50 independent random initialed trained models. Credible intervals correspond to ± 3 standard deviations. Both models were trained using 2000 epochs.

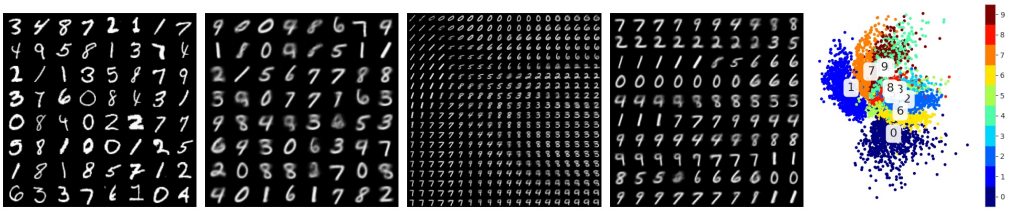
Non-Bayesian approach

Deep ensembles [Lakshminarayanan et al., 2017], is a non-Bayesian approach that independently trains an ensemble of models, providing a model combination that yields a more robust model. Similar to Eq. 2.21 the predictions are combined by averaging each model's predictions. The difference is that each $\theta_1, ..., \theta_m$ is not a sample from the posterior distributing $p(\theta|X)$ but an independent randomly initialized trained model. This frequentist idea of repeating our experiments several times presents another way to estimating model uncertainty. We use randomization-based ensembles to quantify model uncertainty on VAEs in Section 5.3.2.
Nevertheless, this work's focus is unsupervised learning; we use a regression toy example to illustrate the quality of predicted uncertainties obtained from these two approaches. We used the same toy regression problem from [Hernández-Lobato and Adams, 2015] with 20 training input points x generated by sampling uniformly at random in an interval of $[-4, 4]$. Each target y, is generated as $y = x^3 + \epsilon$ where $\epsilon \sim \mathbb{N}(0, 9)$. We trained a neural network with one hidden layer and 100 hidden units using Adaptive Scale SGHMC. However, HMC methods directly get uncertainties; choosing the right burn-in rate and sampling rate from the posterior is challenging. Appendix 7.1 shows experiments using 5,20,30,50 ensembles/posterior samples.

2.5 Variational Autoencoder

Variational autoencoders (VAEs) [Kingma and Welling, 2014, Rezende et al., 2014] provide a principle framework for learning deep latent variable models $p(x, z|\theta)$ and corresponding inference models $q(z|x, \phi)$ with intractable marginal likelihood $p(x|\theta) = \int p(x|z, \theta)p(z)dz$, where x are observable input variables, and z are continuous latent variables. We assume that $p(x, z|\theta) = p(x|z, \theta)p(z)$ factorizes into a likelihood $p(x|z, \theta)$ of x given z and θ and a prior distribution $p(z)$ over z. Analogous to Eq. 2.6 and Eq. 2.11, we can derived an ELBO to approximate $\log p(x)$ as follows:

$$logp(x) \geq ELBO = \mathbf{E}_{q(z|x,\phi)}[\log p(x|z, \theta)] - KL[q(z|x, \phi)||p(z))] \tag{2.22}$$



(a) Data (b) Samples (c) Data Manifold (d) Interpolations (e) Latent Space

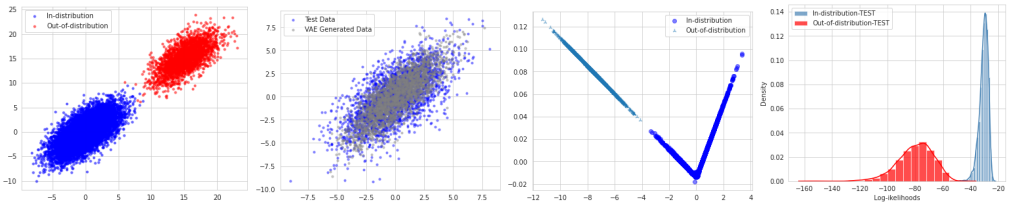
Figure 2.4: VAE on the MNIST dataset with input data $x \in \mathbb{R}^{28x28}$ and latent codes $z \in \mathbb{R}^2$. a) Samples from the train set. b) Data generated from the trained VAE. c) Data Manifold. d) Interpolations in the latent space: The n-dimensional vector μ. e) Behavior in the latent space; each test data point x_i is mapped to the latent space under the recognition model $q_\phi(z|x) : x_i \rightarrow z_i$. Model trained for 200 epochs, learning rate of 0.001, batch size of 256.

VAEs can be seen as two but independently parametrized models whose mean and variance are given by a deep neural network. The encoder or recognition model $q_\phi(z|x)$ approximates the posterior $p_\theta(z|x)$ with the variational parameters ϕ. The decoder or generative model $p_\theta(x|z)$ helps the recognition model to learn meaningful representations of the data with the generative parameters θ. The recognition model is the approximate inverse of the generative model $p_\theta(x|z)$ according to Bayes' rule. The main advantage of the VAE framework over traditional Variational Inference (VI) is that now the recognition model is a stochastic function of the input variables, in contrast to VI where each data-case has a separate variational distribution,which is inefficient for large data-sets [Kingma and Welling, 2019]. The recognition model uses a set of parameters, the variational parameters ϕ, to estimate the variational posterior that is obtained with a simple forward pass through the recognition model $q_\phi(z|x)$ and as such is called 'amortized variational inference' [Gershman and Goodman, 2014]. However, sampling from the stochastic approximate posterior induces sampling noise in the gradients required for learning. To alleviate this problem, an unbiased estimate of the ELBO can be obtained via the 'reparametrization trick' as follows:

$$ELBO \approx \frac{1}{K}\sum_{k=1}^{K} \underbrace{\log p(z_k) - \log q_\phi(z_k|x)}_{\text{KL term}} + \underbrace{\log p(x|z_k)}_{\text{Reconstruction term}} \quad ; \quad z_k \sim q_\phi(z|x) \tag{2.23}$$

In practice, the prior is commonly chosen to be an isotropic Gaussian $p(z) = \mathcal{N}(z; 0, I)$, the variational posterior takes the form of $q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma_\phi(x)^2 \cdot I)$, the generative model $p_\theta(x|z)$ can take the form of a Gaussian distribution for continuous x or Bernoulli for binary inputs x, and typically setting $k = 1$ is sufficient. Unless specified otherwise, we used these settings on the rest of this work.

Both the variational parameters ϕ and the generative parameters θ are jointly learned by maximizing the ELBO, equivalently to minimizing the negative ELBO. Once a model is trained, to generate data that resembles the true underlying generative process consist of : 1) sampling a latent code from the prior $z_i \sim p(z)$ and 2) pass it through the generative model $p_\theta(x|z_i)$. The probabilistic encoder $q\phi(z|x)$ creates codes that represent or disentangle semantically meaningful statistically independent and causal factors of variation in the data [Kingma and Welling, 2019]. We illustrate this in Fig. 2.4 for the MNIST data set using a Bernoulli decoder and a 2-dimensional latent space.



(a) Data (2-D slide) (b) Generated data (c) 2D Latent Space (d) Log-likelihoods

Figure 2.5: VAE toy example for OOD detection. VAEs can be use to reconstruct the density of the model and then use to detect OOD inputs. The model was trained on a) 16-dimensional in-distribution data for 50 epochs, a learning rate of 0.01, latent space = 2, and Adam optimizer. We plot a 2-D slice of this in-distribution (blue) and OOD (red) data. b) 2-D slice of the generated data by the VAE. c) 2D Latent space. d) Histogram of VAE log-likelihoods. VAE log-likelihoods yield to an AUROC =0.9999

2.5.1 VAEs for out-of-distribution detection

VAEs are likelihood based models that theoretically could be used to detect out-of-distribution data. Since VAEs (approximately) maximize the probability $p(X|\theta)$ of the training data X under the model parameters θ. For a new input x^*, one can estimate the density $p(x^*|\theta)$ under the generative model θ [Bishop, 1994]. If $p(x^*|\theta) > \lambda$, where λ is a threshold, then x^* under the generative model θ is in-distribution, OOD otherwise. To get an estimator $\hat{p}(x, |\phi, \theta)$ of the probability $p(x|\theta)$ of an input x under the generative model, one can use *importance sampling* w.r.t the variational posterior $q(z_k|x, \phi)$ as follows:

$$\hat{p}(x|\theta, \phi) = \mathbf{E}_{q(z|x,\phi)}\left[\frac{p(x|z,\theta)p(z)}{q(z|x,\phi)}\right] \simeq \frac{1}{K}\sum_{k=1}^{K}\frac{p(x|z_k,\theta)p(z_k)}{q(z_k|x,\phi)}; \quad z_k \sim q(z|x,\phi) \tag{2.24}$$

where $\hat{p}(x|\theta, \phi)$ is both dependent of the parameters θ and ϕ to emphasis the dependence on the variational parameters ϕ of the proposal distribution $q(z|x, \phi)$.

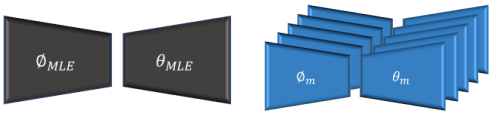
Figure 2.5 illustrates this natural approach to use likelihood-based models to detect out-of-distribution inputs. We fit a VAE on a 16-dimensional toy example. The in-distribution data $x \sim \mathcal{N}(0, \Sigma)$, while OOD data $\sim \mathcal{N}(16, \Sigma)$, where $\Sigma \in \mathcal{R}^{16x16}$ is a positive definite matrix created at random. We observe in Figure 2.5b that VAE generates data like the training data plotted as a 2D-slice. The estimated likelihoods (using Eq. 2.24) are well calibrated, assigning higher likelihoods to in-distribution data and lower to OOD. However, this does not hold for higher-dimensional data. We demonstrate this in the following sections, as recently demonstrated by [Nalisnick et al., 2018], likelihood based approaches in general sometimes assign higher densities to OOD inputs. This questions the use of DGMs (e.g., VAEs) for reliable density estimation detecting OOD inputs. To alleviate this problem, we proposed using Bayesian Variational Autoencoders to capture *epistemic uncertainty*. Consequently, θ_{mle} is not a point estimate anymore, but a random variable.

2.6 Bayesian Variational Autoencoder

VAEs are trained using approximate Maximum Likelihood Estimation (MLE). As a result of this, single point estimates (θ_{MLE}) of the model parameters are obtained. A Bayesian Variational Autoencoder instead, infers a distribution $p(\theta|X)$ over the model parameters. And thus, to obtain the *marginal likelihood* one must integrate out both the latent variable z and model parameters θ as follows

$$p(x|X) = \int\int p(x,z|\theta)p(\theta|X)d\theta = \int\int p(x|z,\theta)p(z)dzp(\theta|z,X)d\theta \tag{2.25}$$

To generate samples, one must now draw a $z \sim p(z)$ from the prior and a θ from the posterior $p(\theta|X)$, and then generate $x \sim p(x|z,\theta)$. Therefore, training a BVAE requires Bayesian Inference in both posteriors $p(z|x,X)$ over the latent variables and $p(\theta|x,X)$ over the model parameters. In this work, we learn both the posterior of the model parameters θ and the variational parameters ϕ. Together with latent variable z, the parameters ϕ, θ are learned jointly using adaptive scale SGHMC and the ELBO as approximation to $\log p(x)$. The resulting BVAE provides M samples $\{\theta,\phi\}_{m=1}^{M}$ from the posterior of the encoder and decoder, $\theta_m, \phi_m \sim p(\theta,\phi|X)$, which can be seen as an *ensemble* of M VAEs as shown in Figure 2.6.



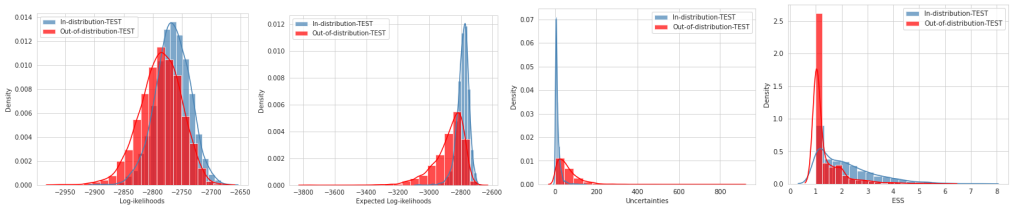
(a) A VAE (b) An Ensemble of M VAEs

Figure 2.6: Illustration of a VAE vs a BVAE.

Detecting OOD using BVAEs

[Nalisnick et al., 2018] showed that DGMs sometimes assign higher likelihoods to OOD inputs than in-distributions. We motivate the use of BVAE and esembles of VAEs by the following toy example. Let's consider a similar setting as in Figure 2.5 but for a higher dimensional space with $x \in \mathcal{R}^{28x282}$ where OOD inputs come from the same Gaussian but with $mu = 30$ rather than $\mu = 0$. Figure 2.7a shows the likelihoods obtained from a VAE, showing a strong overlap with likelihoods obtained for OOD inputs. An AUROC sore of 0.68 reflects that likelihoods estimates from VAEs are poorly calibrated for high dimensional data. The BVAE now generates samples from the posterior, we discard samples within a burn-in phase of B = 1 epoch and store a sample (of both encoder and decoder parameters) after every D = 1 epoch, and use the last recent ten samples. With these 10 samples we compute the average over the predicted likelihoods from each sample (b), the variability of the predictions among the samples (c), and the Effective Sample Size (ESS) (c), proposed by [Daxberger and Hernández-Lobato, 2019] to detect OOD in a BVAE. Table 2.1 reports the AUCROC scores for $M = \{5, 10, 15, 18\}$ samples, and due to space constrains, we only compare an ensemble of 5 VAEs.
[Daxberger and Hernández-Lobato, 2019] demonstrated using the ESS as a score to OOD detection has a relation with information theory since it measure the level of *information* for the model given a new x^* in a sequential Bayesian setting. We refer the reader to the paper for more details.
In this work, we consider both Bayesian and non-Bayesian approaches to quantify epistemic uncertainty. However, we do not study the influence of the prior since this is out of the scope. We emphasize that the quality of Bayesian Modelling Average depends on the choice of the prior [Minka, 2000] and thus could improve the results presented in this work.

[2]Same dimension as MNIST samples



(a) VAE Log-likelihoods (b) Expected $\log p(x|\theta)$ (c) BVAE Uncertainties (d) BVAE ESS

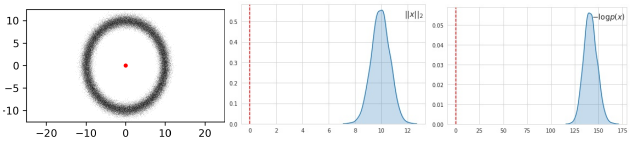
Figure 2.7: VAE toy example for OOD detection. BVAE measures of the expected likelihood, the uncertainties as result of measuring the variability across the ensembles and the ESS score.

Metrics	$M = 5$	$M = 10$	$M = 15$	$M = 18$	5 Ensembles
Expected Log-likelihoods	0.7971	0.8267	0.8841	0.8852	0.7207
Standard deviation	0.8524	0.8709	0.8844	0.8837	0.8529
ESS	0.7707	0.77940	0.7754	0.7752	0.7384

Table 2.1: AUROC scores on an out-of-distribution 784-dimensional toy example using the expected likelihoods, standard deviation and ESS score over 5,10,15,18 samples from a BVAE and on an ensemble of 5 VAEs.

2.7 Typicality

A very intriguing property of high dimensional distributions is that samples concentrate in an annulus ratio of $\sqrt{(dim)}$. The Gaussian Annulus Theorem [Blum et al., 2020] formalizes this idea. Let's consider a 100 dimensional Isotropic Gaussian distribution with **zero mean** and unit variance. We use a similar plot to [Morningstar et al., 2020], and show in Figure 2.8 (a) a 2-D slice of this distribution. Although we would expect to find most of the samples around zero (the point with the highest density), we observe while the mean (red) has the highest likelihood, it is clearly *atypical* since samples (black dots) concentrates in *lower likelihoods*, as shown in (c), around a spherical shell of $\sqrt{(100)}$, as pictured in (b).



(a) Euclidean norm (b) Euclidean norm (c) - log-likelihood

Figure 2.8: (a) A two dimensional projection of a 100 dimensional multivariate normal distribution (Image from [Morningstar et al., 2020]). The origin is shown in red. We show measurements of 100,000 random draws from this distribution: (b) The observed norm of the draws, (c) the negative log-likelihood

This property of high dimensional distributions suggests that a possible reason why VAEs sometimes assign higher likelihoods to OOD data is that the data's typical set may not intersect with the region of high likelihood. Specifically, even when a VAE perfectly maximizes the likelihood of the data under the model parameters, this does not guarantee that the in-distribution data reside in the regions of the highest probability distribution given by the trained VAE. First, let's formalize the definition of the **typical set**. The typical set of a probability is the set whose elements

have an information sufficiently close to that of the expected information (entropy) [MacKay, 2003, Cover and Thomas, 2006]

Definition 2.7.1 (ϵ-Typical set)**.** *For a distribution $p(x)$ with support $x \in \mathcal{X}$ the ϵ-typical set $A_{\epsilon}^{(n)}[p(x)]$ is comprised of all N sequences that satisfy*

$$A_{\epsilon}^{(n)}[p(x)] = \left\{ x \in A_x^N : \left| \frac{1}{N} \log_2 \frac{1}{p(x)} - H[p(x)] \right| \right\} \tag{2.26}$$

where $H[p(x)]$ is the entropy of the unknown data distribution and $\epsilon \mathcal{R}^+$ is a small constant

The phenomenon of typicality in DGMs has been recently studied by various works [Choi et al., 2018, Nalisnick et al., 2019, Morningstar et al., 2020]. In this work, we follow ideas from [Morningstar et al., 2020] who proposed the Density State of Estimators (DoSE). DoSE is inspired from statistical physics, where the probability of observing a particle in a given state is governed by the state's probability and the system's geometry. The DoSe codifies these ideas and proposes several statistics to model the number of different configurations that describe the system. To account for the typical set in practice, the DoSE method constructs an estimation on several summary statistics of the in-distribution data. Particularly, we build the $DoSE_{SVM}$ [Morningstar et al., 2020] by creating a N x M dimensional matrix for N training samples with M statistics per sample. We first run Principal Component Analysis (PCA) to learn a whitening transformation. Then we apply a one-class Support Vector Machine (SVM) to model the frequency of these statistics in the in-distribution test set. Therefore, for a new test point x^* if it does not resemble similar frequency on theses statics, then x^* is a potential OOD sample. For the DoSe method, we computed the following statistics:

1. The approximate posterior prior divergence $KL(q(z|x)||p(z))$. This measure resembles the goodness of fit between the approximate posterior $q(z|x)$ and the prior $p(z)$
2. posterior entropy $H(q(z|x))$. This can be understand as the uncertainty generated by the approximate distribution by the encoder network given a new test point x^*.
3. The approximate posterior/prior cross-entropy H(q(z—x),p(z)), or mutual information.
4. The reconstruction loss $p(x|z)$
5. The negative log-likelihoods

In all cases, the intractable expectation of the posterior distribution is estimated using Monte Carlo approximation with 16 samples.

Chapter 3

Digital pathology

Digital pathology has emerged with the digitization of patient tissue samples on glass slides and the use of whole slide images (WSIs)[Pantanowitz et al., 2018]. WSIs result from scanning a glass slide in a high-resolution digital file using WSI's scanners. A single WSI typically contains trillions of pixels from which thousands of examples of cancerous cells in the form of patches can be extracted. Consequently, the introduction of WSIs has enabled the collection of massive amounts of data needed to train complex deep learning architectures. As a result, deep learning has been successfully applied to digital pathology to automatically analyze WSIs in applications such as prostate cancer and identification of metastases in sentinel lymph nodes [Litjens et al., 2016] using image analysis techniques such as object detection, segmentation, and classification. The success of deep learning in these applications in digital pathology depends, in part, on the reliability of the predictions and the amount of available annotated data. Specifically, a well-calibrated model should be accurate for classes seen during training while assigning high uncertainty estimates to unseen classes or irregularities. Recent efforts have been made to provide uncertainty estimates in the predictions of the deep learning models [Lakshminarayanan et al., 2017, Gal, 2016], and some such as [Linmans et al., 2020] in the context of digital pathology. However, most of these approaches train a discriminative model in a supervised fashion, requiring annotated data. Collecting such precise annotations is expensive and prone to human error since it requires trained pathologists to do this task manually. On the other hand, unsupervised learning techniques require no annotations and can leverage large amounts of image data sampled from WSIs. For this reason, a promising approach in digital pathology is to train a model solely on normal class (e.g., healthy tissue) and use this model to detect train data as in-distribution and any other data type as out-of-distribution.
To this end, we first train all our models exclusively on *normal lymph node tissue* (in-distribution) without having access to annotations. We evaluate the performance to detect out-of-distribution samples using *metastasized breast cancer*. In the rest of this work, we will refer to normal lymph node tissue as 'healthy tissue' and metastasized breast cancer as 'tumor'. The data we used comes from the Camelyon16 challenge [Ehteshami Bejnordi et al., 2017]. Figure 3.1 shows two WSIs at different resolutions. On the left, we show a WSI from the 'healthy' class. There is some background present in the WSI. Features from the healthy class can be visualized better as we zoom deeper in the WSI, e.g., the lower right image. On the right, we show a WSI from the 'tumor' class. We see in blue the annotations where the cancer has metastasized.
We test our model on two variants that we referred to as **PCam32** and **Camelyon**.

PCam32

We used the Kaggle competition Histopathologic Cancer Detection. A modified version of PCam dataset [Veeling et al., 2018] containing 96x96 color images labeled as either "healthy tissue" or "tumor" sample from WSIs of the Camelyon16 dataset. The data was center cropped in 32x32 pixels. A single 32x32 image is consider to be 'tumor' if at least 1 pixels is labelled as tumor. We cleaned the data by thresholding the Frobenius

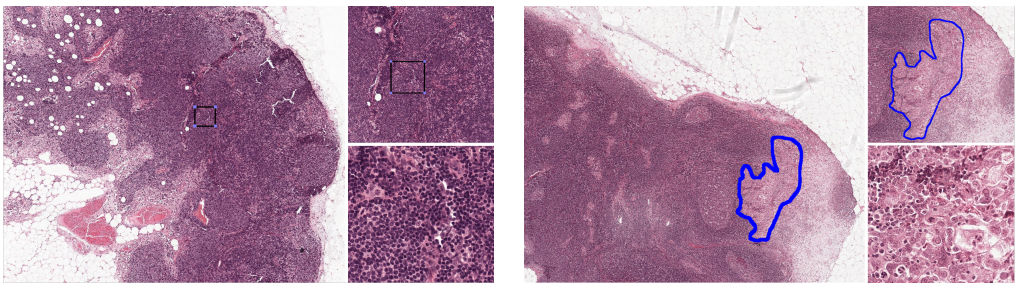


Figure 3.1: Images from in-distribution and OOD at different resolutions. **Left:** A WSI sampled from in-distribution showing 'healthy tissue'. **Right:** A WSI sampled from OOD diagnosed with 'tumor'.

norm for each sample in the dataset to remove possible white images mostly covered by background information. The final dataset has a data split of 88851, 26182, 17824 for train, test and OOD test respectively. The first two rows in Figure 3.2 show samples from the healthy and tumor class respectively.

Camelyon

We manually selected regions from the WSIs of the Camelyon16 dataset, with either healthy or tumor tissue and sampled patches of 32x32 pixels. The resulting dataset is a set of images of 32x32 pixels. All the pixels in the 32x32 image are normal tissue or tumor; e.g., in one image from the tumor class, each pixel contains information taken from the metastasized breast cancer regions only. Data split of 200000, 20000, 20000 for train, test and OOD test respectively. The last two rows in Figure 3.2 show samples from the healthy and tumor class respectively.

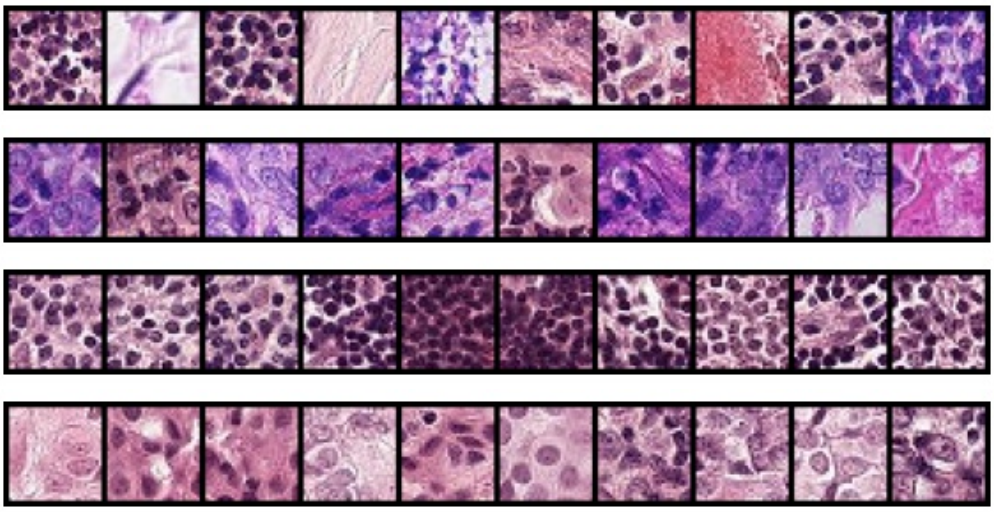
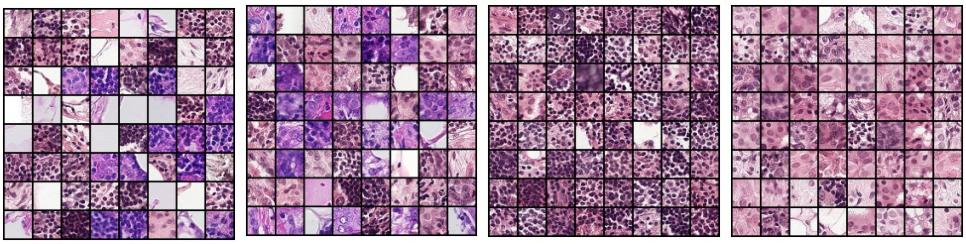


Figure 3.2: Two variant of images from in-distribution and OOD from Camelyon16. First two rows corresponds to **PCam32** from 'healthy' and 'tumor tissue', followed by **Camelyon** correspondingly. Last row show same variant but as a grid of images.

We observed that PCam32 contains more data variability (e.g., color, shapes) than Camelyon, and less information about the out-of-distribution signal in the tumor sam-



(a) Healthy tissue Pcam (b) Tumor tissue Pcam (c) Healthy tissue Camelyon (d) Tumor tissue Camelyon

Figure 3.3: Samples from **PCam32** and **Camelyon** visualized as a grid of images.

ples. In the rest of this thesis, we present these datasets as image grids as shown in Figure 3.3.

Chapter 4

Random prior networks

This chapter introduces *random prior networks* [Ciosek et al., 2020], the first approach we investigated in this thesis. We motivate using random prior networks in digital pathology by the promising state-of-the-art results on out-of-distribution detection shown in [Ciosek et al., 2020]. Particularly in hard tasks such as CIFAR10 vs. SVHN with an AUC score of 0.95 using unlabeled data. Unfortunately, we found a severe flaw in how the authors validated their experiments, and consequently, this flaw may invalidate their work. Specifically, they calculated AUC scores between in-distribution and out-of-distribution samples using the train set rather than the test set. This error can be found directly in the author's code (line 145) and in Table 1 of their paper, where they compare Train vs. cat/deer, Train vs. vehicles, Train vs. excluded, Train vs. SVHN. To illustrate whether this method scales well to image datasets, we reproduced their experiments using the authors' code[1] and report the results obtained using the proper validation test set vs. in-distribution and out-distribution samples. This implementation yields an AUC score of 0.59 in the CIFAR10 vs. SVHN task rather than the authors' presumed score of 0.95. Even when this method work for *regression tasks*, as already shown by [Osband et al., 2018], we show in this section that it does not scale to higher dimensional datasets such as CIFAR10 or PCam32. Therefore we caution the use of random prior networks to high dimensional datasets.

4.1 Method

The paper shows that for a new point x^*, under some reasonable assumptions, a conservative estimate of the uncertainty in expectation over an ensemble of $B = \{1, ..., i\}$ networks pairs $(\{h_i(x), f_i(x)\})$ is obtained using Equation 4.2. This requires using the Mean Square Error (MSE) of Equation 4.1 of a predictor network $\{h_{Xf_i}(x)\}$ trained on unlabeled training data $x = \{x_1, ..., x_n\}$ to match a fixed random network $f_i(x)$, also referred to as the prior network, and estimating the variance $\hat{v}(x_*)$ across the ensembles. Both the prior and the predictor are training using the same MSE objective function of Equation 4.1.

$$\hat{\sigma}^2_\mu(x) = \sum_{i=1}^{B} \frac{1}{MB} ||f(x) - h_{Xf_i}(x)||^2 \tag{4.1}$$

Each prior and predictor network has M number of outputs. The subscript Xf_i in the predictor network $\{h_{Xf_i}(x)\}$ is used to state dependence on the data.

$$\hat{\sigma}(x_*) = max(0, \hat{\sigma}^2_\mu(x_*) + \beta\hat{v}(x_*) - \sigma^2_A) \tag{4.2}$$

Note that if the number of ensembles $B = 1$, estimating uncertainties reduces to 4.1, as $\hat{v}(x_*)$, the variance across the ensembles goes to zero. The aleatoric noise σ^2_A in the implementation is not considered.

[1] https://github.com/microsoft/conservative-uncertainty-estimation-random-priors

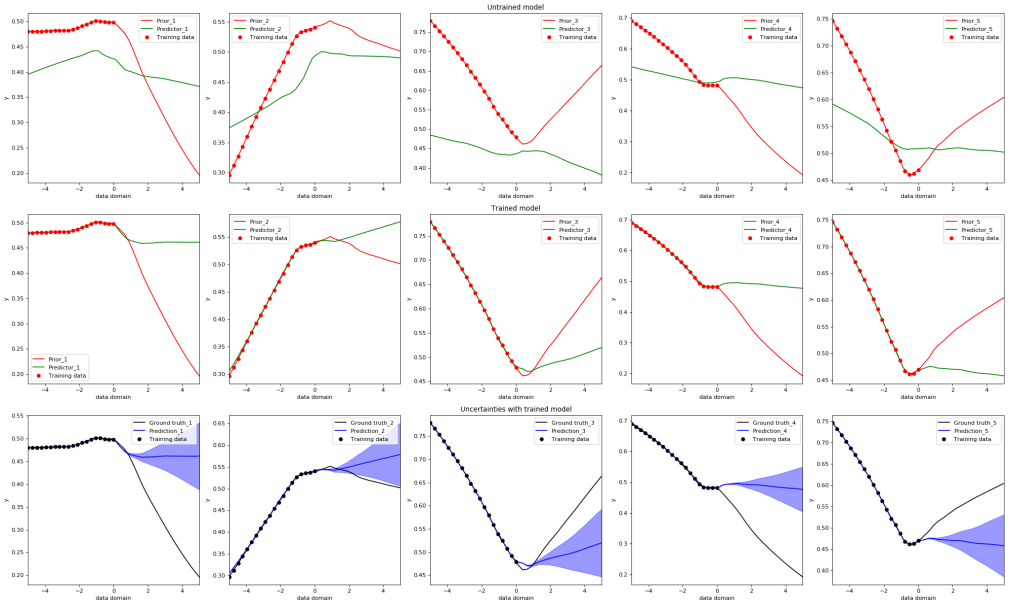


Figure 4.1: First row shows 5 prior-predictor pairs before training. Second row shows the trained prior-predictor pairs after 200 epochs for 20 equidistant training points $x_i \in [-5, 0]$. Third row shows the uncertainty estimate as the shaded blue area for 40 equidistant points $x_i \in [-5, 5]$ obtained after having a trained predictor network.

4.2 Experiments

4.2.1 Regression Task

We start the experiment section by reproducing the the 1D regression toy example as shown by [Ciosek et al., 2020]. We used feed-forward neural networks with 2 layers of 128 units each and a 1-dimensional output layer. A predictor is trained on 20 equidistant training points $x_i \in [-5, 0]$ to match its corresponding prior networks f_i. The first row show the state of the 5 prior(red)-predictor(green) pairs before training. During training each prior network is evaluated on the unlabeled training data and the predictor aims to match its output by minimizing the MSE between its prediction and the output of the prior. In the second row we see how the predictor matches the prior's random pattern after 200 epochs. Finally, the last row shows how at test time, for a new point x^*, the uncertainty estimates (shaded blue area) increase as we go far from the training data, e.g. for points $x'_i \in [0, 5]$, and small uncertainty estimates in regions close to the training points $x_i \in [-5, 0]$.
Continuing with this 1d regression example, we explore the effect of the depth on each hidden layer in the estimated uncertainties. This is motivated by the fact that prior networks approach a Gaussian Process as the layer's width goes to infinity. Table 4.1 reports AUC scores for the baseline (**second column**) as computed by [Ciosek et al., 2020], a 2 layers NN, each layer with 128 units for both prior and predictor networks. **Third column** reports results obtained when increasing the depth of the layers of the prior to 1000 units instead of 128 in each layer. The last column report the results when both the prior and predictor layer's depth are increased to 1000 units. The results show that increasing the layer's depth produces higher uncertainty estimates in regions far from the training domain and so, the separation between in-distribution and out-of-distribution samples is clearer. This yield to a higher AUC score of 0.99 when at the prior layer's depth is increased to 1000 compared to an AUC score of 0.96 when the depth is only

OOD Score (AUCROC)			
Method	[Ciosek et al., 2020]	$W_{f_i} = 1000$	$W_{f_i,h_i} = 1000$
Train vs Excluded AUC	0.96375	0.99625	0.99625
Test vs Excluded AUC	0.96249	0.99250	0.9925

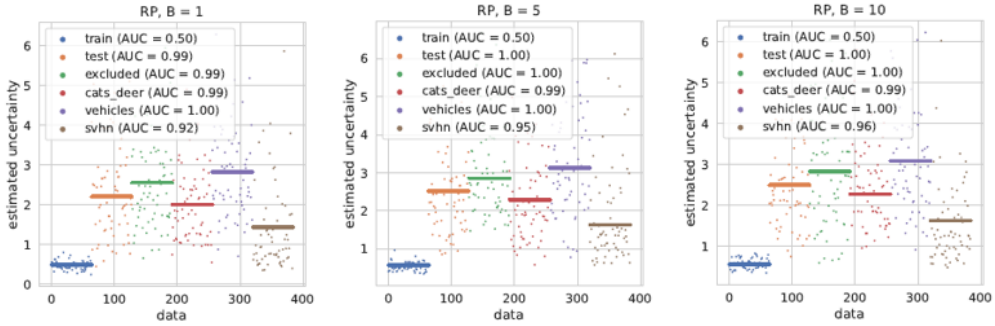
Table 4.1: Evaluating OOD on a 1-d toy regression task. An ensemble of 5 prior-predictor networks is trained on in-in-distribution data $x_i \in [-5, 0]$, and tested in out-of-distribution data $x_i \in [0, 5]$. (**Second column**) Results obtained when the prior and predictor networks are a 2-layer NN with 128 hidden units in each layer. (**Third column**) The prior is now increased to 1000 units in each layer. (**Third column**) Both prior and predictors are now increased to 1000 units in each layer.

128 units in each layer. We observed not change when the predictor network is also increased to 1000 units in each layer.

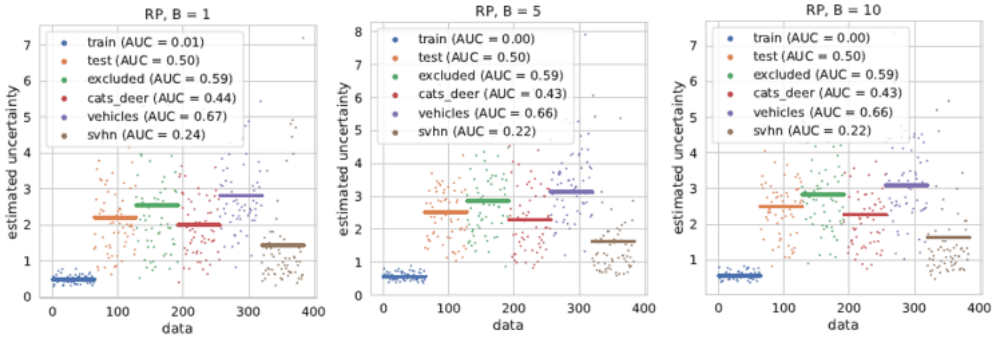
4.2.2 Random prior on image datasets

We now continue our study of estimating uncertainties fitting prior networks on image datasets. First, we reproduced the CIFAR10 experiments as described in [Ciosek et al., 2020]. The results reported in the paper consist of training a model on the classes {bird, dog, frog, horse}, we refer to this set as the 'Train' set. The classes {cat, deer, airplane, automobile, ship, truck} are excluded from the training set and cosider as OOD samples. The model is then tested on this *excluded* set, on a subset of only cats and deers, on samples coming from the vehicles class, and on the SVHN dataset. The AUC score reported are always computed using the 'Train' set. Figure 4.2a shows the results for an ensemble of $B = \{1, 5, 10\}$ prior-predictor networks. This results as computed here, were reported on their paper. We address this issue by using the proper validation test set vs. in-distribution and out-distribution samples. We report these results in Figure 4.2b.
A more sever flaw is found on the reported histogram in Figure 3 [Ciosek et al., 2020]. The authors reported a histogram of seen vs. unseen data. The unseen data histogram is the result of plotting a combined set of the uncertainty estimates obtained from the test and excluded set. The seen data histogram is the plot of the uncertainties obtained from the train set, the same data the model was exposed to during training. This can be seen directly from their code (Line188). Testing the quality of uncertainties in this way is mistaken. In Figure 4.3 we show the results using the proper validation test set. We only plot the uncertainties obtained from the test set and compare them with the uncertainties obtained from the excluded set. We see that both histograms overlap significantly which shows that using the uncertainties estimated from random prior networks on this task as proposed by [Ciosek et al., 2020] is not a good measure to detect OOD samples. This experiment yields an AUC score of 0.59.

Finally, we test this approach in our pathology task. For this, we use the PCam32 datset to train a model using only on the 'healthy' class. To detect out-of-distribution samples, we test the model on the 'tumor' class expecting the model to give high uncertainties to the sample coming from this class, and to obtain low uncertainties to samples coming from the 'healthy' class. However, as shown in Figure 4.4, we observe that the uncertainty estimates coming from the test set overlap significantly with the uncertainties estimated from the tumor class. This results in an AUC score of 0.4073. The model was train using the same training configurations used in the paper for the CIFAR10 experiments (200 epochs, learning rate of 1e-4, Adam optimizer, and an ensemble size of 1). Similar results were obtained using more ensembles.



(a) AUC scores between in-distribution and out-of-distribution samples using the train set rather than the test set as reported by [Ciosek et al., 2020]. Results obtained by directly using the authors' code.



(b) Results obtained using the proper validation test set vs. in-distribution and out-distribution samples. This shows that the proposed method by [Ciosek et al., 2020] do not scale well for high dimensional data.

Figure 4.2: Comparing results obtained by [Ciosek et al., 2020] with results obtained by properly evaluating OOD using the test set instead of the train set.

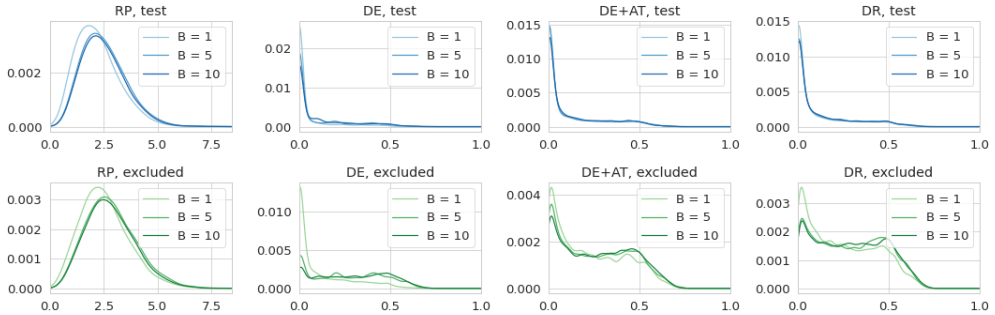


Figure 4.3: Similar to Figure 3 in [Ciosek et al., 2020] but using **test vs excluded** data sets instead. Random prior networks perform poorly, both histograms for the test and excluded overlap considerably.

4.3 Discussion

In this section we empirically demonstrate that the theoretical justification provided by [Ciosek et al., 2020] for the use of random priors to obtained conservative estimates in the context of deep learning do not scale to large datasets such as images. Con-

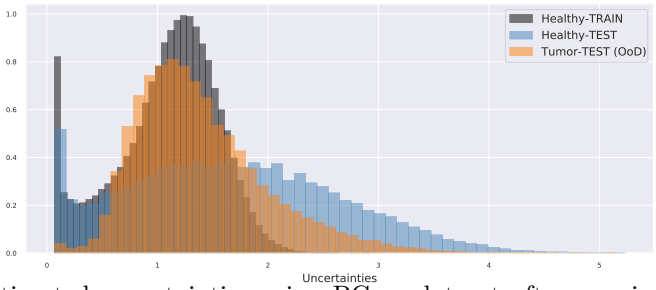


Figure 4.4: Estimated uncertainties using PCam dataset after running the model for 200 epochs, learning rate: 1e-4, using Adam optimizer, ensemble size= 1 (same training setting in the paper). **AUC**: 0.4073. Similar results were obtained using more ensembles.

trary to what the authors claimed, we pointed out that they mistakenly evaluated their experiments by calculating AUC scores between in-distribution and out-of-distribution samples using the train set rather than the test set. Besides the CIFAR10 experiment presented in the paper, we provided an extra experiment to show how this approach does not scale for image data and therefore for digital pathology.

Chapter 5

Experiments

This chapter provides implementation details for our experiments and demonstrates how using the frequency of various model statistics helps to detect OOD samples.

Section 5.1 introduces the datasets we used in our experiments and provide hyper-parameters settings for each case. Section 5.2 explores VAEs as a natural approach to detecting out-of-distribution. Section 5.3 measures epistemic uncertainty and uses it to detect OOD samples using a Bayesian VAE or an ensemble of VAEs. Section 5.4 empirically shows how different model statistics can be used to outperform previous approaches. Finally, section 5.6 provides a failure study and remarks.

5.1 General Settings

We consider six datasets for all the experiments: FashionMNIST, MNIST, SVHN, CIFAR10, Pcam32, and Camelyon. Figure 5.1 shows samples of these datasets. One remark about OOD samples from the PCam32 dataset is that the image is consider to be *tumor* even when at least there is 1 pixel with tumor information in the 32x32 image. Contrary to this, in the Camelyon dataset, each OOD sample has 32x32 pixels of tumor information. For more details about PCam32 and Camelyon dataset we refer the reader to Chapter 3.

(a) FashionMNIST (b) MNIST (c) SVHN (d) CIFAR10 (e) PCam32 (f) Camelyon

Figure 5.1: Datasets

We use these datasets to train VAEs, BVAEs, and ensembles of VAEs. All models have the same encoder architecture. We use convolutional neural networks with padding set to 1 in each layer of the encoder network and place a non-linearity ReLU function after each convolution operation. Table 5.1 shows the architecture used for the encoder. The decoder is the transposed architecture of the generator. We use a standard normal

Operation	Kernel	Strides	Output Channels
Convolution	3 x 3	1 x 1	32
Convolution	3 x 3	2 x 2	64
Convolution	3 x 3	2 x 2	128
Convolution	3 x 3	s x s	256

Table 5.1: Encoder architecture used for the VAE model. $s = 3$ for images of size of 28 x 28, otherwise $s = 5$

prior $p(z) = N(0, I)$ for the latent variables z, an approximate posterior $q_\phi(z|x) = N(z; \mu_\phi(x), \Sigma_\phi(x))$, where $\mu_\phi(x)$ is the mean vector, and $\Sigma_\phi(x)$ is a diagonal covariance matrix. For the continuous cases, we use a Gaussian decoder $p(x|z) = N(x; \mu_\theta(z), I)$, where $\mu_\theta(z)$ is the mean vector (we did not learn the covariance matrix of the decoder). In the results, we make a remark for cases where we use a Bernoulli decoder.
For VAEs, we used Adam optimizer with learning rate of 0.001. For BVAEs we used scale adaptive SGHMC with step size of 0.001 and momentum decay of 0.05. All our experiments are implemented in Python using the Pytorch [Paszke et al., 2017] deep learning framework. We measure the performance on OOD detection using the Area Under the ROC curve (AUCROC) on the in-distribution test set and several OOD datasets.

5.2 VAEs for OOD detection

In this section, we investigate whether VAEs can be used for anomaly detection. We expect to see a well-calibrated VAE assigns higher likelihoods to in-distribution data compared to out-of-distribution data. However, we showed that VAEs *sometimes assign higher likelihoods* to OOD than to in-distribution samples.

To ensure the latent space provides enough flexibility to learn meaningful representations of the data, in Table 5.2 we present an ablation study across different dimensions of the latent space and measure the likelihoods in average bits per dimension (BPD, lower means better). We choose the values that lead to the best likelihoods on average BPD and use this same configuration for all the experiments.

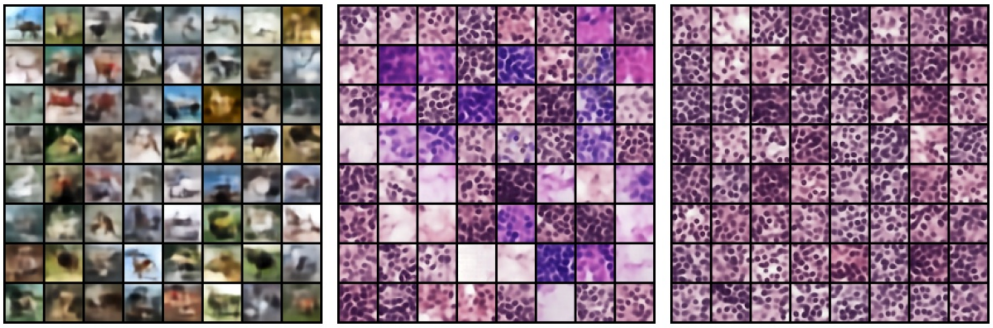
Dataset	**Epochs**	$z = 2$	$z = 8$	$z = 16$	$z = 32$
		VAE			
Mnist†	200	0.27	0.18	**0.17**	0.17
FashionMnist†	200	0.47	**0.43**	0.43	0.43
FashionMnist	200	0.16	0.10	**0.10**	0.10
SVHN	200	0.09	0.06	**0.04**	0.04
		$z = 32$	$z = 64$	$z = 128$	$z = 256$
CIFAR10	1000	0.10	**0.08**	0.08	0.08
PCam32	1000	0.17	0.15	**0.13**	0.13
Camelyon	1000	0.25	0.20	0.18	**0.17**

Table 5.2: Average bits per dimensions (BPD) for different dimensions of z, latent space, for a VAE trained for 200 and 1000 epochs. .

†$p(x|z) \sim$ Bernoulli

When training a VAE on MNIST with Bernoulli decoder, FashionMNIST, and SVHN with a Gaussian decoder, the optimal latent space size is $z = 16$ using 200 epochs. For FashionMNIST with Bernoulli decoder the optimal latent space is $z = 8$. For CIFAR10, PCam32 and Camelyon it is $z = 64, 128, 256$ correspondingly using 1000 epochs. Figure 5.2 shows samples generated by the VAE with such configuration. The images generated under the model parameters $x_{\text{generated}} \sim p(x|z, \theta)$ are visually similar to those from the in-distribution data as a consequence that the VAE approximates the (unknown) data distribution $p^*(x)$ with the model distribution $p(x|\theta)$ under the model parameters θ. This may suggests that VAEs likelihoods can be used to detect OOD samples. Particularly, we may expect to get low likelihoods for OOD samples and high likelihoods to in-distribution samples. In Table 5.3, we report likelihoods in

(a) FashionMNIST (c) MNIST (e) SVHN



(b) CIFAR10 (d) Pcam32 (f) Camelyon

Figure 5.2: Samples generated by a VAE trained on (a) FashinMNIST, (b) CIFAR10 (c) MNIST, (d) Pcam32, (e) SVHN and (f) Camelyon datasets. These samples are visually similar those from the in-distribution data 5.1

average BPD (lower is better) obtained from these trained models but tested with in-distribution and OOD samples. In the upper right side, We observed that a model trained on CIFAR10 assigns *lower* BPD to samples from the SVHN dataset than to samples from the CIFAR10 dataset. This phenomenon is intriguing since the model was not trained on SVHN data and neither generates samples like SVHN. We follow the same notation used by [Nalisnick et al., 2018], showing the average BPD for the training data (CIFAR10-Train), the in-distribution test data (CIFAR10-Test) and the out-of-distribution test data (SVHN-Test). This notation allows us to observe if the model generalizes well to unseen in-distribution data. For example, we see that this model assigns slightly *lower* BPD to the training data (CIFAR10-Train) than to the in-distribution test data (CIFAR10-Test). We report the normalized histograms in Figure 5.3c of the log-likelihoods (higher is better) of the three splits. We see that SVHN test set is shifted to the right hand side of the plot (highest likelihood).
Continuing with this study, a VAE trained on FashionMNIST assigns similar likelihoods to samples from the MNIST dataset. This is again intriguing since the model did not was exposed to any sample from the MNIST dataset, and however the model still assigns similar likelihoods to these samples. Figure 5.3a shows the histogram of the log-likelihoods of the three splits. This phenomenon is not symmetric. Figure 5.4 shows that likelihoods are properly calibrated for a VAE trained on MNIST and tested on FashionMNIST. Similarly, CIFAR10 does not have higher likelihoods under a VAE trained on SVHN.
We find similar results for our digital pathology datasets. Figure 5.3b shows a complete overlap between log-likelihoods obtained from a VAE trained solely on healthy tissue and tested on Tumor tissue. This results is not surprising, since the information content of samples from the PCam32 tumor class overlapped with samples from PCam32 healthy.

Dataset	Avg BPD
VAE trained on FashionMNIST†	
FashionMNIST-Train	0.4281
FashionMNIST-Test	0.4360
MNIST-Test	0.4378
VAE trained on MNIST†	
MNIST-Train	0.1705
MNIST-Test	0.1746
FashionMNIST-Test	3.8490

(a) FashionMNIST case

Dataset	Avg BPD
VAE trained on CIFAR10†	
CIFAR10-Train	0.0350
CIFAR10-Test	0.0362
SVHN-Test	0.0223
VAE trained on SVHN†	
SVHN-Train	0.0156
SVHN-Test	0.0171
CIFAR10-Test	0.0477

(b) CIFAR10 case

Table 5.3: Average bits per dimensions calculated on VAE for FashionMnist for different dimensions of the latent space. $^\dagger p(x|z) \sim$ Bernoulli

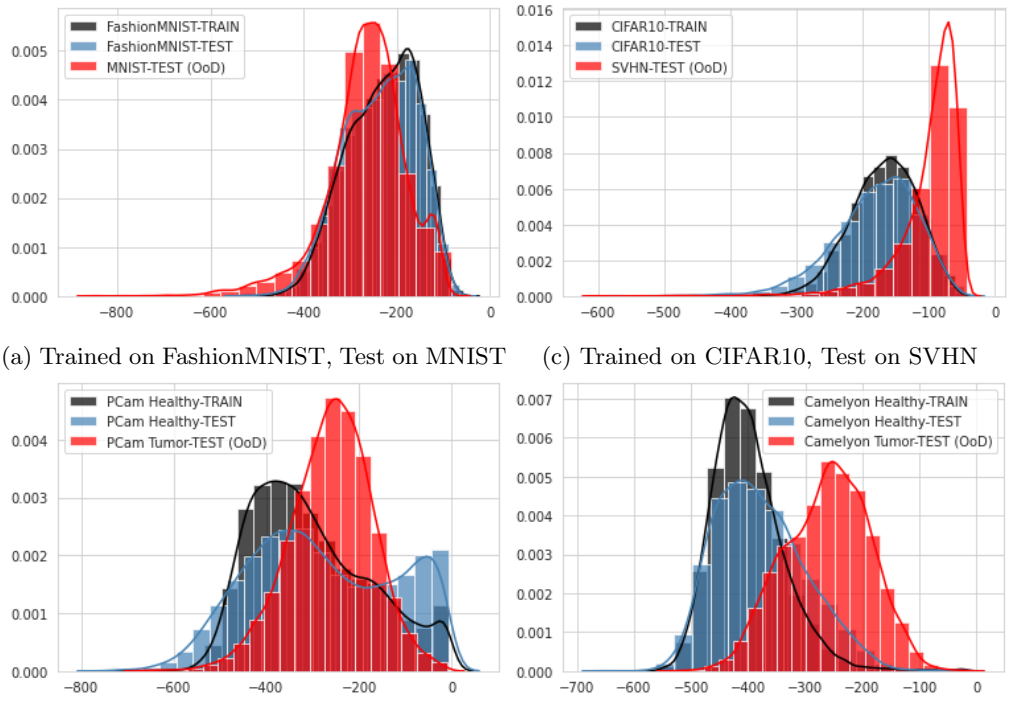
This is because an image sample from the PCam32 dataset is considered to be *tumor* if at least 1 pixel out of the 32x32 pixels is tumor, while the rest of the pixels are healthy tissue. However, more striking results were obtained when we tested the model on Camelyon. Figure 5.3d shows how a model trained on healthy class assigns higher log-likelihoods to the tumor class. Contrary to [Nalisnick et al., 2018] who showed that FashionMNIST vs MNIST assigns higher likelihoods to data from MNIST, we found that this problem can be solved by using a Gaussian decoder instead of a Bernoulli decoder to model the data. Nonetheless, we conclude this section by warning that using only the log-likelihoods estimated by a VAE is not a informative statistic for anomaly detection. In the next section we estimate *epistemic uncertainty* in VAEs and use it to discriminate between in-distribution and OOD samples.

5.3 Epistemic Uncertainty

We now estimate the epistemic uncertainty in VAEs using a Bayesian VAE and an ensemble of VAEs. We use this information to discriminate between in-distribution and OOD samples using several metrics. Particularly, we use the *Expected likelihoods* (ELL), the *Effective Sample Size* (ESS), the standard deviation (σ) across the posterior samples predictions and the entropy. Following [Daxberger and Hernández-Lobato, 2019], for BVAEs we discard samples within a burn-in phase of B = 1 epoch and store a sample (of both encoder and decoder parameters) after every D = 1 epoch, and use the last recent ten samples to compute the metrics. We use an ensemble of 10 VAEs and compute the same metrics.

5.3.1 Bayesian Variational Autoencoders

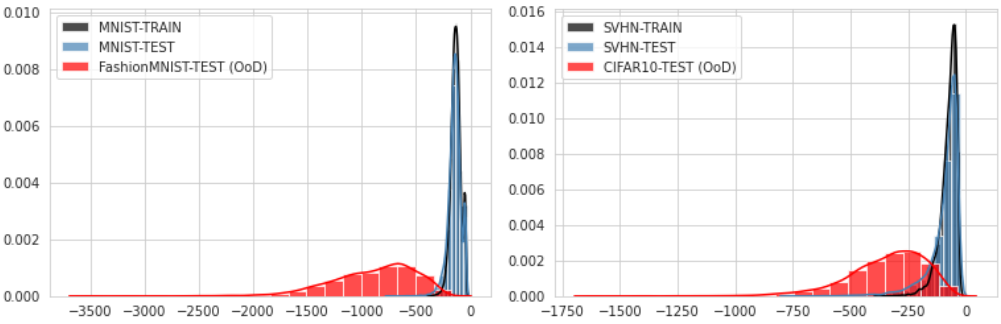
To check that BVAEs provide similar generative properties to VAEs, we take the last sample of the posterior distribution of the model parameters, obtained by training the BVAE using SGHMC, and use it to generate new data. Figure 5.5 shows data generated by the BVAEs using the 'optimal' latent space values as obtained from Table 5.2. We observe that data generated from the BVAEs are similar to those from the VAEs. Both models were trained using the same number of iterations.
We present the results obtained from a BVAE in Table 5.4. ESS and σ outperform the likelihood scores from VAEs when the log-likelihoods are poorly calibrated. For example, a BVAE trained on FashionMNIST with a Bernoulli decoder and tested on the MNIST test set, the AUCROC scores are improved from 0.5992 to 0.94666 and



(a) Trained on FashionMNIST, Test on MNIST (c) Trained on CIFAR10, Test on SVHN

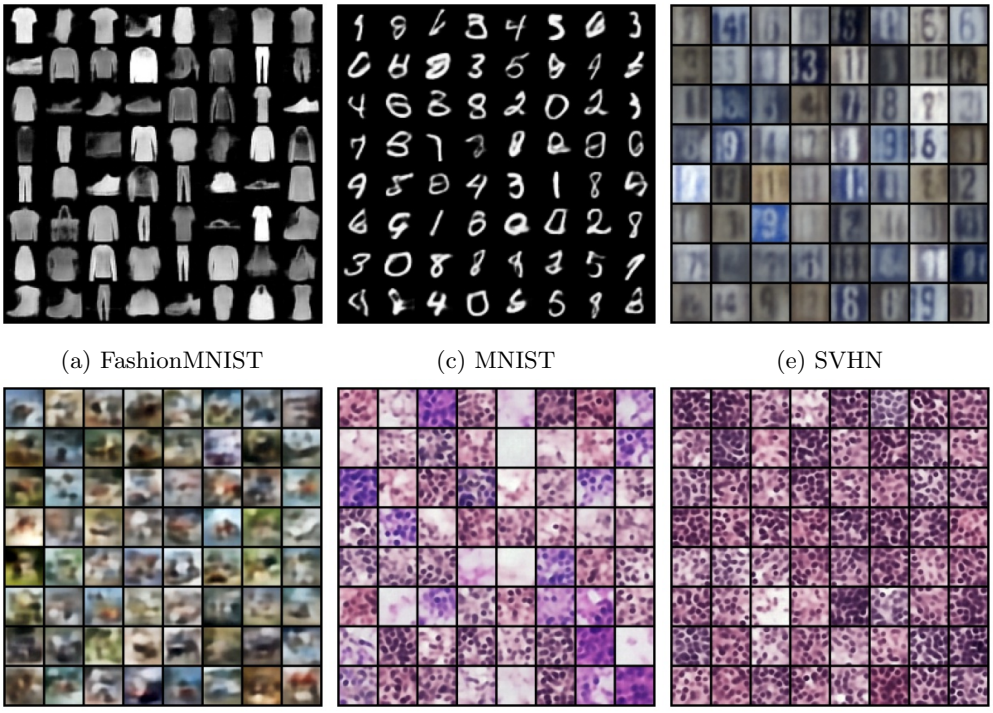
(b) Trained on PCam32 Healthy, Test on Tumor (d) Trained on Camelyon Healthy, Test on Tumor

Figure 5.3: Histogram of VAE log-likelihoods for FashionMNIST vs MNIST, CIFAR10 vs SVHN, Pcam32 Healthy vs Tumor, and Camelyon Healthy vs Tumor.



(a) Trained on MNIST, Test on FashionMnist (b) Trained on SVHN, Test on CIFAR10

Figure 5.4: Histogram of VAE log-likelihoods for MNIST vs FashionMNIST, and SVHN vs CIFAR10. Note that both models assign lower log-likelihoods to the OOD test sets yielding an AUCROC score of 0.99 and 0.94 respectively. This illustrates the asymmetric property of VAEs compare to Figures 5.3a and 5.3c

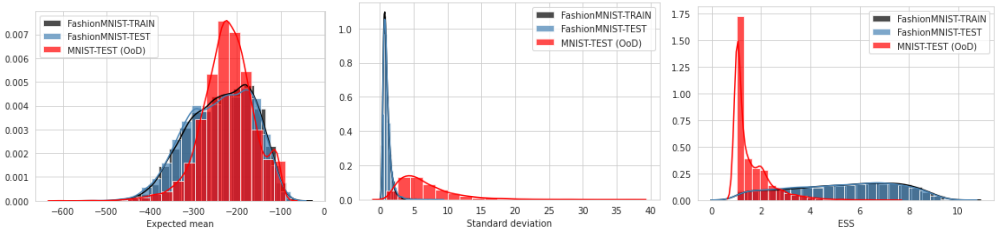


(a) FashionMNIST (c) MNIST (e) SVHN

(b) CIFAR10 (d) Pcam (f) Camelyon

Figure 5.5: Samples generated by a BVAE trained on (a) FashinMNIST, (b) CIFAR10 (c) MNIST, (d) Pcam32, (e) SVHN and (f) Camelyon datasets. These samples are visually similar those from the in-distribution data 5.1 but less sharp to those obtained by the VAE 5.2.

0.9914 respectively. The expected likelihood yields similar results to those obtained using the log-likelihoods of the VAEs. In Figure 5.6 we shows the histograms obtained from the ELL, σ, and ESS scores. We observed that the variability across the ensembles predictions reflects the uncertainty of the model to OOD data. Unfortunately, this does not scaled to other experiments, e.g., CIFAR10 vs. SVHN results are lower than random chance (0.5), except for the entropy that yields an AUCROC of 0.72.



(a) Expected $\log p(x|\theta)$ (b) BVAE Uncertainties (c) BVAE ESS

Figure 5.6: FashionMNIST vs MNIST. BVAE measures of the expected likelihood, the uncertainties as result of measuring the variability across the ensembles and the ESS score.

Experiment	LL	ELL	ESS	σ	H
FashionMnist † vs Mnist	0.5992	0.4367	0.9466	0.9914	0.1007
FashionMnist vs Mnist	0.9921	0.9848	0.8641	0.9723	0.1221
Mnist vs FashionMnist †	0.9999	0.9998	0.9470	0.9967	0.0089
CIFAR10 vs SVHN	0.1193	0.1073	0.3434	0.2856	0.7213
SVHN vs CIFAR10	0.9467	0.9454	0.7837	0.8456	0.4360
PCam32 Healthy vs Tumor	0.4522	0.4524	0.4224	0.3596	0.3910
Camelyon Healthy vs Tumor	0.1470	0.1518	0.4575	0.3991	0.1687

Table 5.4: AUCROC OOD scores using the likelihoods (LL) obtained from a VAE, the expected likelihoods (ELL), the Effective Sample Size (ESS), the variance shown as the standard deviation over the likelihood estimates from a BVAE (σ), and the entropy H of the categorical distribution obtained from the BVAE.

For our digital pathology experiments we see no relevant improvement, all the scores are below than 0.5. Contrary to [Daxberger and Hernández-Lobato, 2019] and similarly to [Nalisnick et al., 2018], we found that the log-likelihoods of a VAE can easily distinguish SVHN vs. CIFAR10.

5.3.2 Ensemble of VAEs

Experiment	LL	ELL	ESS	σ	H
FashionMnist † vs Mnist	0.5992	0.5210	0.9647	0.9985	0.0001
FashionMnist vs Mnist	0.9921	0.9912	0.9408	0.9942	0.0002
Mnist vs FashionMnist †	0.9999	0.9995	0.9105	0.9910	0.0000
CIFAR10 vs SVHN	0.1193	0.1185	0.2963	0.2480	0.3751
SVHN vs CIFAR10	0.9467	0.9667	0.7963	0.8849	0.0621
PCam32 Healthy vs Tumor	0.4522	0.4518	0.4313	0.4313	0.1518
Camelyon Healthy vs Tumor	0.1470	0.2930	0.1463	0.1398	0.1244
PCam32 Healthy vs CIFAR10	0.5760	0.55199	0.8446	0.8517	0.0242

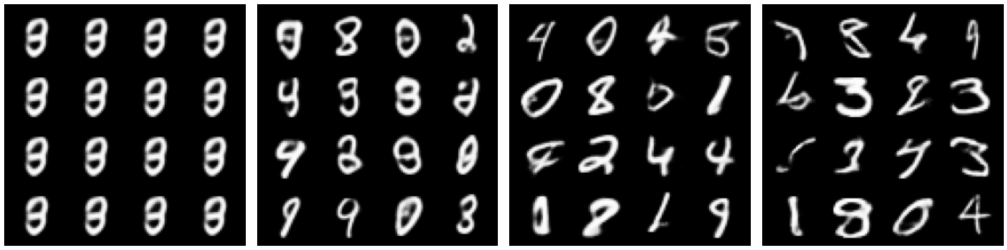
Table 5.5: Similar to Table 5.4 but using an Ensemble of 10 VAEs

Continuing with our study of epistemic uncertainty, in Table 5.5 we show results obtained when training an ensemble of 10 VAEs. For comparison, we used the same metrics used in a BVAE. We emphasize that using the ESS in an ensemble of VAEs has not theoretical foundations. Interestingly, VAE's ensembles yield slightly similar results to those obtained from a BVAE. However, this method is approximately 10 times more

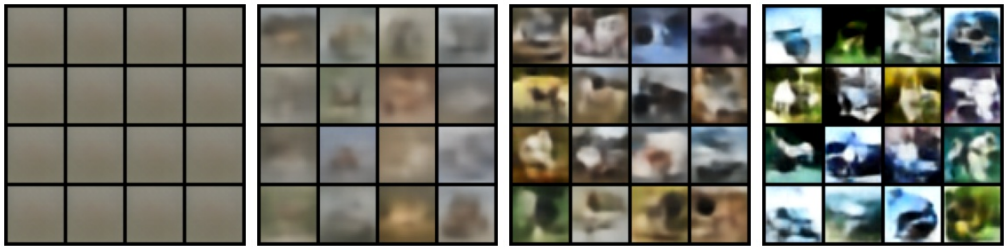
computationally complex than training a BVAE. The main difference is that the entropy score for the CIFAR10 vs. SVHN in a BVAEs outperformed all the other methods with an AUC score of 0.72 while the same metric in an ensemble of VAEs yields 0.37. It is noteworthy to notice that the expected likelihood for both BVAEs and ensembles of VAEs yields similar results to those obtained using the log-likelihoods of a VAEs. In this settings, both BVAEs and ensembles of VAEs do not provide a robust approach to detect OOD data in digital pathology.

5.4 Typicality

In this section, we now empirically evaluate DoSE [Morningstar et al., 2020] using a one class SVM to estimate the support of our summary statistics. We used the same trained VAEs from section 5.2 and compute statistics on the training set. In particular, we compute 5 statistics: 1) The KL divergence between the posterior and the prior, 2) the cross-entropy between the posterior and the prior, $T_n^{\text{ent}}(X) = H[q(Z|X), p(z)]$, 3) the posterior entropy $T_n^{\text{ent}}(X) = H[q(Z|X)]$, 4) the distortion (the expected log-likelihood from the decoder) $T_n^{\text{distortion}}(X) = \mathbb{E}_{q(X,Z)}[\log p(X|Z)]$ and 5) the log-evidence, computed using a 16-sample IWAE estimate. Figure 5.8 shows examples of these statistics obtained for a model trained on Camelyon (first row) or Pcam32(second row). After computing these statistics on the training set, we then fit the SVM on the set of statistics for each data point.
The problem of typicality states that for high dimensional distribution, samples x sampled from the input space X concentrate in an annulus ration of $\sqrt{dim(x)}$. We hypothesize that this phenomenon is mapped into the latent space through the encoder in VAEs and therefore the samples that the VAE generates ares samples from the typical set. To illustrate this, we sample from the normal prior $p(z) = \mathbb{N}(0, \sigma * I)$ over several regions to find where the samples concentrate as a result of the generative process. Figure 5.7 shows samples generated by VAEs trained on MNIST or CIFAR10 for $\sigma = \{0, 0.5, 1.0, 1.5\}$. We find that the best results are found when we sample from $\sigma = 1$, which turns out to be where the z variables concentrate ($\sqrt{dim(z)}$), appealing to the annulus theorem. For example, for a VAE trained on MNIST with a latent space dimension of 16, the z variables that generated samples like the train data concentrate around 4. For MNIST when we observe that we get some semantic information when we sample from the point with highest density in the latent space contrary to CIFAR10. We report same experiment for PCam32 and Camelyon in Figure 7.2 of the appendix.
Table 5.6 reports the results for all baselines. We did not integrate results obtained from the VAE's ensemble since they performed very similar to BVAEs. In general, we find DoSE to outperform almost all baselines or achieve similar results. Notably, DoSE yield an **AUCROC score of 0.84 in Camelyon Healthy vs. Tumor**. However, for PCam32 DoSE does not provide a clear separation between healthy and tumor samples. A possible reason is the amount of information present in the OOD samples overlaps with the information in in-distribution samples. This is because an image sample from the PCam32 dataset is considered to be *tumor* if at least 1 pixel out of the 32x32 pixels is tumor, while the rest of the pixels are healthy tissue. Consequently, the statistics obtained from from in-distribution test set (red) and OOD test set (red) are very similar as shown in Figure 5.8 (**second row**). Since DoSE builds a classifier on top of the statistics of the model, we also explore DoSE using only the log-likelihoods as a feature. We report this results on last column as DoSe_{T_1}. We observe that DoSe_{T_1} using only the log-likelihoods as the unique feature provides similar results to those using 5 statistics. We note that for FashionMNIST vs MNIST using as Bernoulli decoder DoSe_{T_1} performe poorly. A possible reason is that FashionMNIST samples are better



(a) Mean $|z| = 0.0$ (c) Mean $|z| = 1.97$ (e) Mean $|z| = 3.91$ (g) Mean $|z| = 5.03$



(b) Mean $|z| = 0.0$ $\sigma = 0$ (d) Mean $|z| = 3.95$ $\sigma = 0.5$ (f) Mean $|z| = 7.96$ $\sigma = 1$ (h) Mean $|z| = 11.81$ $\sigma = 1.5$

Figure 5.7: (**First row**) VAE trained on MNIST with a latent space dimension = 16, annulus ratio = 3.87. (**Second row**) VAE trained on CIFAR10 with a latent space dimension = 64 and an annulus ratio = 7.94. The best results are found when we sample from $\sigma = 1$, which turns out to be where the z variables concentrate ($\sqrt{dim(z)}$), appealing to the annulus theorem.

model with a Gaussian than with a Bernoulli distribution and so, the statistics provided by this model are already defected by the choice of the way we model the data.

Experiment	**LL**	**ELL**	**ESS**	σ	**DoSE**	**DoSE**$_{T_1}$
FashionMnist † **vs Mnist**	0.59	0.4367	0.9466	0.9914	0.9472	0.5269
FashionMnist vs Mnist	0.99	0.9848	0.8641	0.9723	0.9925	0.9920
Mnist vs FashionMnist †	0.99	0.9998	0.9470	0.9967	0.9999	0.9999
CIFAR10 vs SVHN	0.11	0.1073	0.3434	0.2856	0.7513	0.7278
SVHN vs CIFAR10	0.94	0.9454	0.7837	0.8456	0.9430	0.9400
PCam32 Healthy vs Tumor	0.45	0.4524	0.4224	0.3596	0.3939	0.3764
Camelyon Healthy vs Tumor	0.14	0.1518	0.4575	0.3991	**0.8427**	**0.8061**

Table 5.6: AUCROC OOD scores using the likelihoods (LL) obtained from a VAE, the expected likelihoods (ELL), the Effective Sample Size (ESS), the variance shwon as the standard deviation over the likelihood estimates from a BVAE (σ), the DoSE scores and DoSe$_{T_1}$ using only the log-likelihoods as the unique statistic.

Figure 5.9 shows the histograms of VAE and DoSE. This illustrates how using several model's statistics helps to detect OOD samples in digital pathology when sufficient information about the OOD signals is provided.
We qualitatively evaluated DoSE by computing a confusing matrix for a given experiment. This allows us to visually analyze the performance of the algorithm. We take the top 16 images with the highest and lowest OOD scores for a given trained model. We show the results for DoSE, and the ESS, STD and ELL obtained from a trained BVAE. Figure 5.10 shows in the top row the SVHN vs. CIFAR10 which results to be an easy task, since using only the log-likelihoods yields an AUC score of 0.94. In the bottom row, shows the difficult task for CIFAR10 vs SVHN. Even when it is difficult

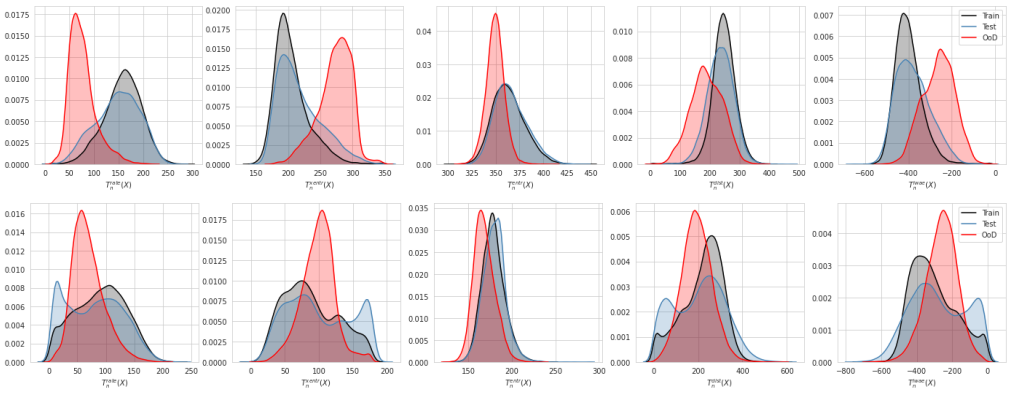
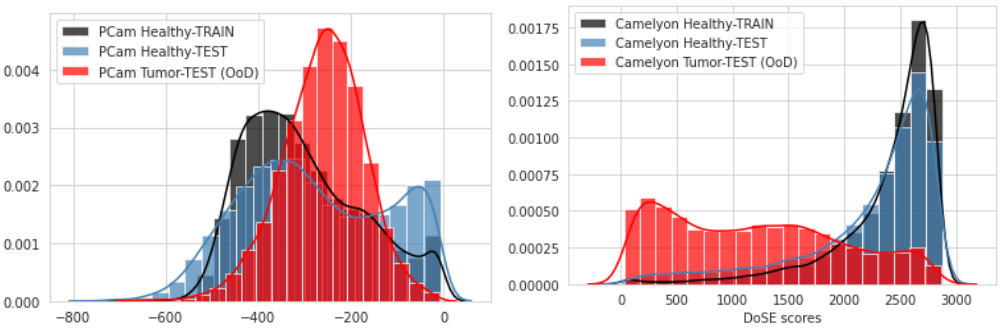


Figure 5.8: Statistics used as proposed in the DoSE paper for this cleaner dataset (Camelyon16): leftmost column shows the KL divergence between the posterior and the prior. The second column shows the cross-entropy between the posterior and the prior. The third column shows the entropy of the encoder. The fourth shows the distortion (the expected log-likelihood from the decoder). The last column shows the log-evidence, computed using a 16-sample IWAE estimate.(**First row**) Statistics from Camelyon data set. (**Second row**) Statistics from PCam32.



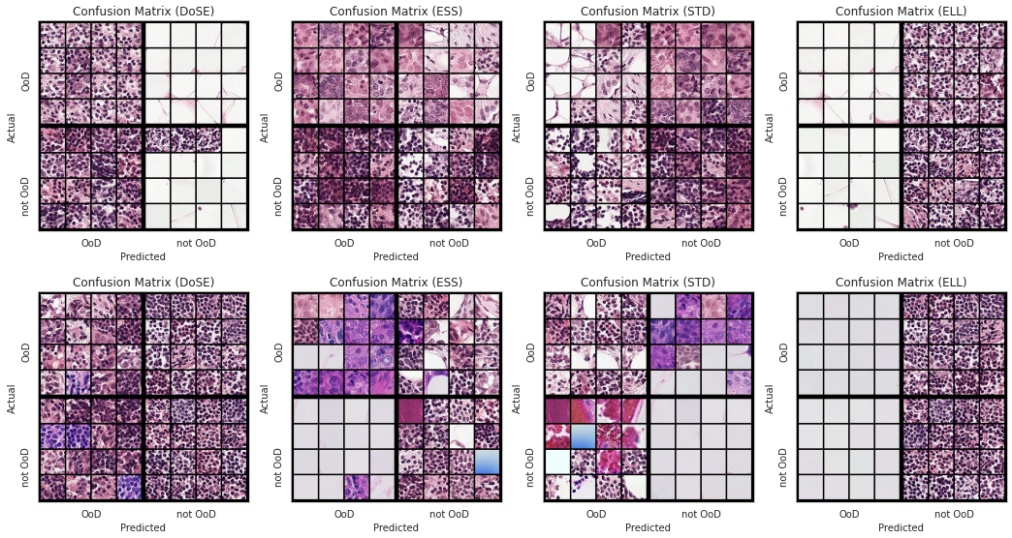
(a) VAE log-likelihoods AUROC=0.14 (b) DoSE **AUROC=0.84**

Figure 5.9: Histograms for Camelyon Healthy vs Tumor, using a) VAE log-likelihoods and b) DoSE scores to detect OOD samples.

to provided a solid argument on these results, we speculate that DoSE identifies as potential in-in-distribution candidates samples with high color contrast. Figure 5.11 shows the confusion matrix for (**first row**) Camelyon Healthy vs Tumor and (**second row**) PCam32 Healthy vs Tumor. Interestingly, DoSe helped us identify that in the Camelyon dataset there was noise in the data in the form of white background. Because this dataset was carefully selected to guarantee that all the 32x32 pixels in an image from the tumor class were indeed *tumor*. This noise is also presented in the *healthy* class, therefore the model predicts this data as OOD. In the second row we see the confusion matrices for the PCam Healthy vs Tumor experiment, we see again that the limited amount of information of OOD signal in samples from the tumor class affects the prediction of the model.

(a) DoSE (b) ESS (c) STD (d) ELL

Figure 5.10: Histogram of VAE log-likelihoods for FashionMNIST vs MNIST, CIFAR10 vs SVHN, Pcam32 Healthy vs Tumor, and Camelyon Healthy vs Tumor.



(a) DoSE (b) ESS (c) STD (d) ELL

Figure 5.11: Confusion matrices for the methods used to evaluate OOD detection for (**first row**) Camelyon Healthy vs Tumor, and (**second row**) PCam32 Healthy vs Tumor, and. The images in each quadrant are order by the confidence of each model.

Chapter 6

Discussion and Future Work

This last chapter discusses the results obtained in this thesis and promising avenues for future work.

6.1 Discussion

In this work, we explore unsupervised methods to detect out-of-distribution samples in digital pathology. We aim to train a model solely on healthy tissue to detect tumor tissue as OOD data. Nonetheless, recent efforts have been developed to tackle this problem using supervised learning methods. These methods require annotated data, which is very expensive to collect in digital pathology. Our experiments show promising avenues to detect OOD samples in digital pathology without having access to annotations.
We started exploring random prior networks [Ciosek et al., 2020] and found that the authors did not correctly validated their experiments. As a consequence of this, their work may be invalidated. We then explore VAEs, as a natural approach to model the data density distribution and use the likelihoods, we expect to see a well-calibrated VAE assigns higher likelihoods to in-distribution data than to out-of-distribution data. However, we find that VAEs sometimes assign higher likelihoods to OOD samples. Experimentally, we observe that VAE likelihoods do not reflect the semantic properties of the data, and they assign higher likelihoods to samples that are visually distinct to a human, such as images of healthy tissue to images of frogs, horses, and dogs. Because the VAE log-likelihoods are not sufficient to discriminate OOD samples, we then aim to estimate the epistemic uncertainty in VAEs using a Bayesian VAE and an ensemble of VAEs. We use this information to discriminate between in-distribution and OOD samples using several metrics. However, we find this to not be useful to digital pathology since all the AUCROC score for our digital pathology tasks were below 0.5.
A possible reason that VAEs fail to detect OOD data is caused by a mismatch between the areas of high probability density and the model's typical set due to the sensitivity to high dimensional distributions. We hypothesize that this phenomenon may translate through the encoder to the latent space. We illustrate this by showing that samples generated using latent variables from the annulus ration the latent space look like the data distribution, while samples outside of this annulus don't. We further explore DoSE and model the frequency of different statistics using a one class SVM. We find DoSE to outperform best, discriminating *healthy* (in-distribution) from *tumor* (out-of-distribution) samples with an **AUCROC score of 0.84**.
Finally, we observed that the amount of information of the OOD signal present in an image is fundamental to detect OOD samples effectively. The experiment with PCam32 dataset show this to be the case since an image sample from this dataset is considered to be tumor if at least 1 pixel out of the 32x32 pixels is tumor, while the rest of the pixels are healthy tissue. Consequently, the statistics obtained from the in-distribution test set and OOD test set are very similar, so DoSE performs poorly with an AUCROC score of 0.39.

6.2 Future work

The results obtained in this thesis suggest promising avenues for future research in unsupervised out-of-distribution methods in digital pathology. Although DoSe yields an AUCROC score of 0.85 in our digital pathology task, we observed that in PCam32, DoSE performs poorly with an AUCROC score of 0.39. This indicates that the amount of information of the OOD signal present in an image is fundamental to detect OOD samples effectively. We did not explore the influence of the amount of information of the OOD signal in the OOD sample. We remark that an OOD sample can be composed of two signals: the background signal and the information signal. A similar approach was recently explored by [Ren et al., 2019], who removes background information and focuses on the semantics. However, we would like to consider that the image is composed of three signals: the background signal, the tissue signal, and the tumor signal. We would then like to explore whether removing the background and the tissue signal combining this approach with DoSE will help to detect OOD samples.
The choice of the DoSe statistics is rather empirical than a principle approach. Therefore, there are multiple avenues of research in this direction. We could try to model different statistics using expertise from pathologist about what relevant features characterized the *healthy* tissue. We could try to learn this features using DNNs.
Despite we provide an ablation study over the latent space of the VAEs, we did not explore different configuration in the VAE itself, e.g., choosing a different prior than the standard normal prior, and increasing the complexity of the network. Another promising research line is to explore how SurVAE Flows, a method that combines normalizing flows and VAEs, can be used to OOD detection [Nielsen et al., 2020].
Finally, for our experiments using BVAE we did not optimize the hyper-parameters such as the optimal number of posterior samples, the prior of the parameters. A promising line of research line is to identify the optimal configuration on SGHMC that leads to sampling from the typical set. [Betancourt, 2018] suggested the use of the Effective Samples Size to find such configuration. [Zhang et al., 2019] proposed Cyclical Stochastic Gradient MCMC to automatically explore high dimensional and multimodal distributions present in deep learning. This method could be potentially useful to ameliorate the problem of typicality in DGMs.
Despite the encouraging results in our digital pathology task, many questions remain open. It is unclear how this can be applied in real settings; since we only tested our models in patches of 32x32 pixels, we did not explore how this could be scale to WSIs. In digital pathology, there is huge variability in the data, and so robust *unsupervised OOD* methods should handle such randomness. Therefore we need to provide ways to incorporate prior knowledge and constraints into these models so they can be safely deployed in real clinical settings.

Bibliography

[Betancourt, 2018] Betancourt, M. (2018). A conceptual introduction to hamiltonian monte carlo.

[Bishop, 1994] Bishop, C. M. (1994). Novelty detection and neural network validation. *IEE Proceedings-Vision, Image and Signal processing*, 141(4):217–222.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

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

[Blum et al., 2020] Blum, A., Hopcroft, J., and Kannan, R. (2020). *Foundations of Data Science*. Cambridge University Press.

[Chen et al., 2014] Chen, T., Fox, E. B., and Guestrin, C. (2014). Stochastic gradient hamiltonian monte carlo.

[Choi et al., 2018] Choi, H., Jang, E., and Alemi, A. A. (2018). Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*.

[Ciosek et al., 2020] Ciosek, K., Fortuin, V., Tomioka, R., Hofmann, K., and Turner, R. (2020). Conservative uncertainty estimation by fitting prior networks. In *Eighth International Conference on Learning Representations (ICLR)*.

[Cover and Thomas, 2006] Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA.

[Daxberger and Hernández-Lobato, 2019] Daxberger, E. and Hernández-Lobato, J. M. (2019). Bayesian variational autoencoders for unsupervised out-of-distribution detection.

[Depeweg, 2019] Depeweg, S. (2019). *Modeling Epistemic and Aleatoric Uncertainty with Bayesian Neural Networks and Latent Variables*. Dissertation, Technische Universität München, München.

[Duane et al., 1987] Duane, S., Kennedy, A., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics Letters B*, 195(2):216 – 222.

[Ehteshami Bejnordi et al., 2017] Ehteshami Bejnordi, B., Veta, M., Johannes van Diest, P., van Ginneken, B., Karssemeijer, N., Litjens, G., van der Laak, J. A. W. M., , and the CAMELYON16 Consortium (2017). Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. *JAMA*, 318(22):2199–2210.

[Gal, 2016] Gal, Y. (2016). Uncertainty in deep learning. *University of Cambridge*, 1:3.

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

[Gershman and Goodman, 2014] Gershman, S. and Goodman, N. (2014). Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36.

[Hernández-Lobato and Adams, 2015] Hernández-Lobato, J. M. and Adams, R. P. (2015). Probabilistic backpropagation for scalable learning of bayesian neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1861–1869. JMLR.org.

[Hinton and van Camp, 1993] Hinton, G. E. and van Camp, D. (1993). 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, page 5–13, New York, NY, USA. Association for Computing Machinery.

[Hora, 1996] Hora, S. C. (1996). Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety*, 54(2-3):217–223.

[Jordan et al., 1999] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y., editors, *ICLR*.

[Kingma and Welling, 2019] Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *CoRR*, abs/1906.02691.

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

[Lakshminarayanan et al., 2017] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *ArXiv*, abs/1612.01474.

[Laumann, 2018] Laumann, F. (2018). Bayesian convolutional neural networks with bayes by backprop.

[Linmans et al., 2020] Linmans, J., van der Laak, J., and Litjens, G. (2020). Efficient out-of-distribution detection in digital pathology using multi-head convolutional neural networks. In *Medical Imaging with Deep Learning*, pages 465–478. PMLR.

[Litjens et al., 2016] Litjens, G., Sánchez, C. I., Timofeeva, N., Hermsen, M., Nagtegaal, I., Kovacs, I., van de Kaa, C. H., Bult, P., van Ginneken, B., and van der Laak, J. (2016). Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. *Scientific Reports*, 6.

[MacKay, 2003] MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Copyright Cambridge University Press.

[Minka, 2000] Minka, T. (2000). Bayesian model averaging is not model combination.

[Morningstar et al., 2020] Morningstar, W. R., Ham, C., Gallagher, A. G., Lakshminarayanan, B., Alemi, A. A., and Dillon, J. V. (2020). Density of states estimation for out-of-distribution detection.

[Murphy, 2012] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

[Nalisnick et al., 2018] Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. (2018). Do deep generative models know what they don't know?

[Nalisnick et al., 2019] Nalisnick, E., Matsukawa, A., Teh, Y. W., and Lakshminarayanan, B. (2019). Detecting out-of-distribution inputs to deep generative models using typicality. *arXiv preprint arXiv:1906.02994*.

[Neal, 2012] Neal, R. M. (2012). Mcmc using hamiltonian dynamics.

[Nielsen et al., 2020] Nielsen, D., Jaini, P., Hoogeboom, E., Winther, O., and Welling, M. (2020). Survae flows: Surjections to bridge the gap between vaes and flows.

[Osband et al., 2018] Osband, I., Aslanides, J., and Cassirer, A. (2018). Randomized prior functions for deep reinforcement learning.

[Pantanowitz et al., 2018] Pantanowitz, L., Sharma, A., Carter, A. B., Kurç, T., Sussman, A., and Saltz, J. (2018). Twenty years of digital pathology: An overview of the road travelled, what is on the horizon, and the emergence of vendor-neutral archives. *Journal of Pathology Informatics*, 9.

[Paszke et al., 2017] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Ren et al., 2019] Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., Dillon, J., and Lakshminarayanan, B. (2019). Likelihood ratios for out-of-distribution detection. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32, pages 14707–14718. Curran Associates, Inc.

[Rezende et al., 2014] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models.

[Springenberg et al., 2016] Springenberg, J. T., Klein, A., Falkner, S., and Hutter, F. (2016). Bayesian optimization with robust bayesian neural networks. In *Advances in neural information processing systems*, pages 4134–4142.

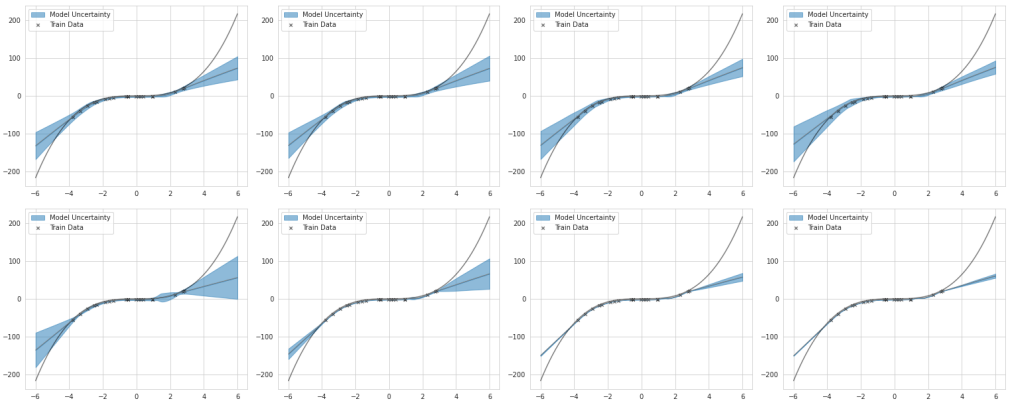
[Veeling et al., 2018] Veeling, B. S., Linmans, J., Winkens, J., Cohen, T., and Welling, M. (2018). Rotation equivariant CNNs for digital pathology.

[Zhang et al., 2019] Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. (2019). Cyclical stochastic gradient MCMC for bayesian deep learning. *CoRR*, abs/1902.03932.

Chapter 7

Appendix

7.1 Deep Ensembles



(a) 50 samples (b) 30 samples (c) 20 samples (d) 5 samples

Figure 7.1: Deep ensembles (**Upper row**) vs SGHMC (**Botton row**) using 50,30,20,5 ensembles/posterior samples.

7.2 VAE ELBO derivation

$$\mathcal{KL}(q(z)||p(z|X) = \mathbb{E}_{z\sim q}\left[\log q(z) - \log p(z|X)\right] \tag{7.1}$$

By applying Baye's rule $p(z|X) = p(X|z)p(z)/p(X)$:

$$\begin{aligned}
\mathcal{KL}(q(z)||p(z|X) &= \mathbb{E}_{z\sim q}\left[\log q(z) - \log \frac{p(X|z)p(z)}{p(X)}\right] && (7.2)\\
&= \mathbb{E}_{z\sim q}\left[\log q(z) - (\log(p(X|z)p(z) - \log p(X))\right] && (7.3)\\
&= \mathbb{E}_{z\sim q}\left[\log q(z) - \log(p(X|z) - \log p(z) + \log p(X)\right] && (7.4)\\
&= \mathbb{E}_{z\sim q}\left[\log q(z) - \log(p(X|z) - \log p(z)\right] + \log p(X) && (7.5)
\end{aligned}$$

Last step comes by the fact of log(X) does not depend on z. Later we can rearrange the terms by negating both sides.

$$\begin{aligned}
\log p(X) - \mathcal{KL}(q(z)||p(z|X) &= \mathbb{E}_{z\sim q}\left[-\log q(z) + \log(p(X|z) + \log p(z)\right] && (7.6)\\
&= \mathbb{E}_{z\sim q}\left[\log p(X|z)\right]] + \mathbb{E}_{z\sim q}\left[\log p(z) - \log q(z)\right] && (7.7)\\
&= \mathbb{E}_{z\sim q}\left[\log p(X|z)\right]] + \mathcal{KL}(q(z)||p(z)) && (7.8)
\end{aligned}$$

7.3 Numerical implementation of the log-likelihood

For a numerically stable implementation of the log-likelihood we always work with log probabilities. The log-likelihood $p(x)$ is approximate using Monte Carlo. The intractable expectation under the approximate posterior is replace by the following equation using k samples z_k of the approximate posterior distribution $q_\phi(z|x)$

$$\log p(x|\theta) \simeq \log\Big(\frac{1}{K}\sum_{k=1}^{K}\frac{p(x|z_k,\theta)p(z_k)}{q(z_k|x,\phi)}\Big); \quad z_i \sim q(z_k|x,\phi) \tag{7.9}$$

$$= \log\Big(\frac{1}{K}\sum_{k=1}^{K}\exp\Big(\log\frac{p(x|z_k,\theta)p(z_k)}{q(z_k|x,\phi)}\Big)\Big) \tag{7.10}$$

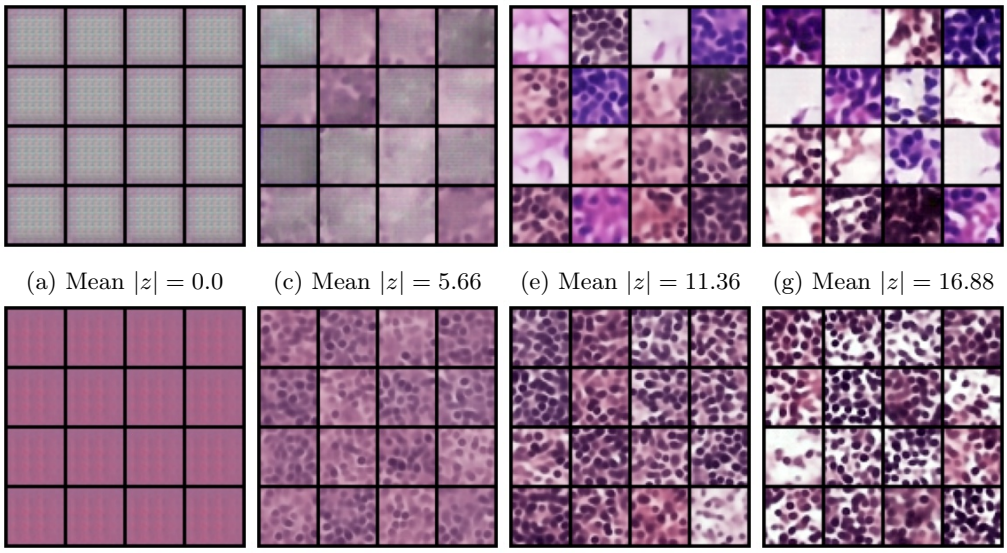
$$= \text{logmeanexp}_{z_k}\Big(\log\frac{p(x|z_k,\theta)p(z_k)}{q(z_k|x,\phi)}\Big)\Big) \tag{7.11}$$

$$= \text{logmeanexp}_{z_k}\Big(\log\frac{\exp\left(\log p(x|z_k,\theta)p(z_k)\right)}{\exp\left(\log q(z_k|x,\phi)\right)}\Big)\Big) \tag{7.12}$$

$$= \text{logmeanexp}_{z_k}\Big(\log\frac{\exp\left(\log p(x|z_k,\theta)+\log p(z_k)\right)}{\exp\left(\log q(z_k|x,\phi)\right)}\Big)\Big) \tag{7.13}$$

$$= \text{logmeanexp}_{z_k}\Big(\log p(x|z_k,\theta)+\log p(z_k)-\log q(z_k|x,\phi)\Big) \tag{7.14}$$

7.4 Typicality



(a) Mean $|z| = 0.0$ (c) Mean $|z| = 5.66$ (e) Mean $|z| = 11.36$ (g) Mean $|z| = 16.88$

(b) Mean z norm = 0.0 (d) 7.99 (f) z norm mean: 16.04 (h) 23.95

$\sigma = 0$ $\sigma = 0.5$ $\sigma = 1$ $\sigma = 1.5$

Figure 7.2: (**First row**) VAE trained on PCam32 with a latent space dimension = 128, annulus ratio = 11.27. (**Second row**) VAE trained on Camelyon with a latent space dimension = 256 and an annulus ratio = 15.97