Uncertainty Quantification

Estimating Aleatoric and Epistemic Uncertainty
in Medical Image Segmentation

Vanja Sophie Cangalovic

A thesis presented for the degree of
Master of Science

Faculty 3 - Mathematics and Computer Science
University of Bremen
August 14, 2022

1st supervisor: Dr. Hans Meine
2nd supervisor: Prof. Dr. Udo Frese
Advisor: Felix Thielke

Abstract

Medical imaging is a cornerstone for medical diagnosis, treatment planning, and clinical studies. In order to delineate anatomical structures and other regions of interest in such images, deep neural networks can be employed, performing image segmentation for the medical expert. Because of the high-risk setting, these models need to be not only exact and robust, but also indicate error likelihood via reliable and meaningful uncertainty estimates. This predictive uncertainty can be subdivided into *aleatoric* and *epistemic* uncertainty, and captured using deep ensembles, Bayesian neural networks, and additional loss attenuating output neurons. The main contribution of this work is a comprehensive comparison between the direct decomposition of uncertainty in Bayesian neural networks via the mutual information metric and the explicit modelling of epistemic and aleatoric uncertainty in Bayesian neural networks with an additional heteroscedastic loss-attenuating neuron. This comparison is performed in the context of medical image segmentation of the liver from CT scans, employing a 3D au-net as base architecture. The quality of the uncertainty decomposition in the resulting uncertainty maps is qualitatively evaluated and quantitative behaviour of aleatoric and epistemic uncertainty is systematically compared for different experiment settings with varying training set sizes, label noise, and distribution shifts. The results show the mutual information decomposition to robustly yield meaningful aleatoric and epistemic uncertainty estimates, with both largely conforming to their definitions and consistent with other works. Noisiness in the activation of the loss-attenuating neuron leads to the conclusion that the mutual information decomposition remains significantly more suited for uncertainty decomposition even for Bayesian neural networks combined with loss-attenuating neurons. This work further found that the addition of a heteroscedastic neuron does not improve the quality of the uncertainty estimates when decomposed via the mutual information metric. An ancillary contribution is the demonstration of a strong influence of the choice of loss function on the quality of uncertainty decomposition, with soft Dice loss heavily deteriorating the quality of the decomposed uncertainties.

Contents

List of Figures

List of Tables

Acronyms

BNN Bayesian neural network.

CNN convolutional neural network.

CT computed tomography.

ECE expected calibration error.

HLS heteroscedastic logit smoothing.

HLSN heteroscedastic logit smoothing neuron.

HUN heteroscedastic uncertainty neuron.

ID in distribution.

IoU Intersection over Union.

MC Monte Carlo.

MCE maximum calibration error.

MI mutual information.

MLE maximum likelihood estimate.

MRI magnetic resonance imaging.

NLL negative log likelihood.

NN neural network.

OOD out of distribution.

ReLU rectified linear unit.

Introduction

1.1 Motivation

Medical imaging is a cornerstone of medical diagnosis, treatment planning, and clinical studies. In order to delineate anatomical structures and other regions of interest in such images accurately and on a large scale, deep neural networks can be employed to perform image segmentation for the medical expert. Fully convolutional neural networks, for instance, achieve state-of-the-art results on various medical image segmentation tasks (Iantsen et al., 2021; Ma, 2021). However, deep networks are traditionally optimised via point estimates which have been shown to result in overconfident predictions and have the tendency to fail silently. This is especially problematic in the medical setting in which wrong model decisions can have severe consequences.

Because medical image segmentation is a high-risk scenario, the models need to be exact and robust, not only providing accurate predictions, but also reliable, i.e. well-calibrated, and meaningful *uncertainty estimates*. For instance, instead of blindly giving a prediction to practitioners, the model should indicate via a higher uncertainty estimate when its segmentation quality is likely to be lower because the input image does not resemble the training samples or because labels for such an image were noisy. In the setting of medical diagnosis support, reliable uncertainty estimates not only avoid potentially dire consequences of model errors, but might more generally help to increase the clinical staff's confidence in deep learning models.

Apart from being helpful to the end-application, uncertainty estimates might also support the training itself, as well as aid the scientific understanding of neural network dynamics in general, by providing additional insights into the network's decision process. Active learning is a particularly salient example, in which uncertainty estimates can be used to select samples to annotate and subsequently train on, with training efficiency largely depending on the quality of the given uncertainty estimates.

From a Bayesian perspective, the overall predictive uncertainty can be subdivided into two major uncertainty types:

- Aleatoric uncertainty captures the noise or stochasticity inherent to the underlying process which generates the training data.
- Epistemic uncertainty represents the uncertainty of the model itself.

The former is irreducible, while the latter can be explained away, e.g. by training a model on more data. This distinction increases the interpretability of the resulting uncertainty estimates, which is particularly desirable for human-in-the-loop scenarios. Moreover, state of the art active learning strategies employ epistemic uncertainty sampling, and can therefore profit from high-quality epistemic estimates (Nguyen et al., 2019). Aleatoric uncertainty, on the other hand, lends itself naturally to use in loss attenuation during training (Kendall and Gal, 2017), and might further be used to automatically detect annotation inconsistencies in the training data (Thulasidasan et al., 2019).

1.2 Research Question

To capture predictive uncertainty, deep ensembles and Bayesian Neural Networks have been proposed and numerous recent approaches employ additional output neurons that perform loss attenuation for difficult training samples. Two major approaches for obtaining decomposed uncertainty estimates are the direct decomposition of a Bayesian Neural Network's predictive distribution (Kwon et al., 2020; Mobiny et al., 2021), derived mathematically from the variability of the predictive distribution, and the combination of a Bayesian neural network with an additional loss-attenuating output neuron as proposed by Kendall and Gal (2017), which, by combining two uncertainty quantification methods, makes the sources of different uncertainty types more explicit and adds another measure for aleatoric uncertainty.

While the combination of BNNs with an additional output neuron for uncertainty decomposition has been investigated (Kendall and Gal, 2017), there exists, to the best of the author's knowledge, no comprehensive comparison of such an augmentation with a direct decomposition metric applied on the predictive distribution of a pure Bayesian neural network.[1] The goal of this work is to provide such a comparison, with a focus on the quality of the resulting decomposed uncertainty values. More succinctly:

> How does the quality of uncertainty decomposition compare for a Bayesian neural network with mutual information metric, and the combination of a Bayesian neural network with an additional heteroscedastic loss attenuating neuron?

1.3 Approach

The research question is explored in the context of semantic segmentation on computed tomography scans of the liver region. To this end, this work investigates two binary segmentation architectures applied on the Liver Tumor Segmentation data set (LiTS; Bilic et al., 2019): a classical Bayesian Neural Network and a Bayesian Neural Network extended with a simplified version of the heteroscedastic uncertainty neuron, henceforth called heteroscedastic logit smoothing neuron, first introduced for vanilla neural networks by (Neumann et al., 2018). This variant is assumed to perform comparably to Kendall and Gal's heteroscedastic neuron while being simpler, more intuitive, and computationally cheaper due to a lack of sampling.

Construing the activation of the heteroscedastic logit smoothing neuron as an aleatoric uncertainty measure, as done for Kendall and Gal's heteroscedastic uncertainty neuron by (Nair et al., 2020; DeVries and Taylor, 2018b; Kwon et al., 2020), is compared to the mutual information metric applied on the predictive distribution for decomposing the predictive uncertainty into aleatoric and epistemic parts.

In order to evaluate the quality of the decomposed uncertainties, experiments with varying training set sizes, artificial label noise, and inference on out of distribution (OOD) samples are performed. General segmentation performance as well as calibration measures are reported and the resulting uncertainty maps, derived for both uncertainty types, are qualitatively evaluated.

[1] Kwon et al. (2020) conduct a direct comparison between their proposed variance decomposition and the combination of a Bayesian neural network with heteroscedastic neurons. However, there appears to be a substantial flaw in their reproduction of Kendall and Gal (2017) which will be addressed in this work.

Varying the training set size is expected to affect epistemic uncertainty estimates, adding artificial label noise to the training masks should increase aleatoric uncertainty estimates, while inference on OOD samples is expected to result in higher epistemic uncertainty.

1.4 Contribution

The main contribution of this work is an extensive comparison of the quality of uncertainty decomposition between a vanilla Bayesian neural network and a Bayesian neural network with an added heteroscedastic logit smoothing neuron.

The comparison results in advice for practitioners on the decomposition afforded by the predictive distribution versus the separate activation of the heteroscedastic neuron.

This work is, to the best of the author's knowledge, also the first investigation into the addition of a heteroscedastic logit smoothing neuron to a Bayesian neural network, and the first work to explore the parallels between Kendall and Gal (2017) and Neumann et al. (2018).

Over the course of this work, some differences between Kendall and Gal (2017)'s work and its reproductions (Kwon et al., 2020; DeVries and Taylor, 2018b; Nair et al., 2020) are illuminated, most notably concerning the choice of aleatoric uncertainty measure.

While the tendency of models overfitting when trained with Dice loss has been observed in many prior works (Mehrtash et al., 2020; Guo et al., 2017; Bertels et al., 2021; Sander et al., 2019), the author is not aware of any work that focuses on the effect of Dice loss on the quality of uncertainty decomposition. This work evaluates the importance of the choice of loss function on the decomposed uncertainties derived from both model architectures.

1.5 Outline

The rest of this work is structured as follows. Section 2 provides the context of this work, it introduces the domain of medical imaging, outlines computational approaches for image segmentation, with a focus on the state of the art convolutional neural networks, as well as giving some definitions related to uncertainty and model calibration, which will be used throughout this work. Section 3 aims to give a comprehensive overview of current uncertainty quantification methods with a focus on learnt loss attenuation and uncertainty decomposition techniques. Section 4 describes in detail the underlying data, the neural network architectures and hyperparameters, as well as the evaluation setups and measures. Section 5 reports all results obtained from the conducted experiments. Section 6 uses these results in order to answer the research question and to gather additional insights into some observations that were made along the way. Finally, Section 7 summarises this work, describes its limitations, and provides an outlook for potential future work.

Background

This section provides an overview of the application domain, basic concepts, and the foundations of the employed methods. It first introduces the domain of medical imaging with a focus on computed tomography scans of the liver region, its characteristics, and challenges. Then, it describes computational approaches for image segmentation, with a particular focus on deep convolutional neural networks (CNNs) as the state of the art approach. Their architecture and training are briefly described, along with some popular regularisation techniques. Lastly, model uncertainty and its two types, aleatoric and epistemic uncertainty, are defined and model calibration along with some established metrics are presented.

2.1 Medical Imaging

Medical imaging provides 2D or 3D images of a patient's internal tissue structure and is therefore an important tool that enables clinicians to (often non-invasively) perform diagnosis, intervention planning, and the tracking of disease progression.

Medical images are commonly described along anatomical planes, which provide canonical viewing angles of the patient (Betts et al., 2013), as shown in Figure 2.1, or with region and organ specific coordinate systems, such as the regions and quadrants describing the abdomen in Figure 2.2.

The three prevalent technologies are sonography, computed tomography (CT), and magnetic resonance imaging (MRI). Figure 2.3 shows a rough comparison of the different imaging properties (O'Neill et al., 2015). The following explanations focus on computed tomography, as it is the method with which the data underlying this work was acquired.

Artefacts in medical imaging are undesirable visual phenomena appearing in the image that do not reflect reality, and are caused by physical limitations of the scanning process (Maier et al., 2018). Examples include noise as random variability in the intensity of voxels, partial volume effects where multiple structures smaller than the image resolutions produce a measurement of their average density, and motion artefacts caused by the patient or tissue moving during a scan.

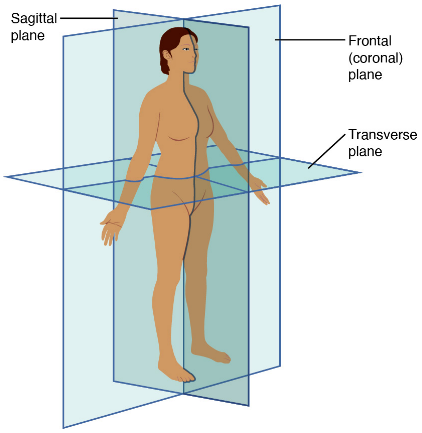


Figure 2.1: Visualisation of the most common anatomical planes: sagittal, coronal, and transverse. (Betts et al., 2013); image taken from OpenStax licensed under Creative Commons Attribution License v4.0.

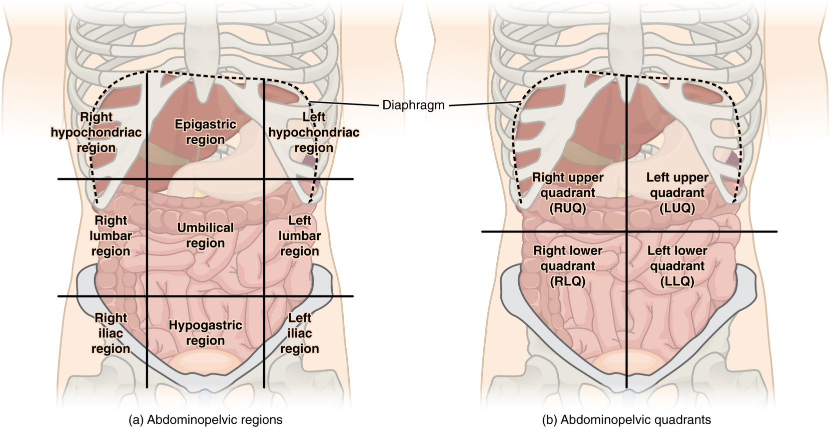

Right hypochondriac region
Epigastric region
Left hypochondriac region
Diaphragm
Right lumbar region
Umbilical region
Left lumbar region
Right iliac region
Hypogastric region
Left iliac region
(a) Abdominopelvic regions
Right upper quadrant (RUQ)
Left upper quadrant (LUQ)
Right lower quadrant (RLQ)
Left lower quadrant (LLQ)
(b) Abdominopelvic quadrants

Figure 2.2: The regions and quadrants used to describe the abdominal cavity (Betts et al., 2013); image taken from OpenStax licensed under Creative Commons Attribution License v4.0.

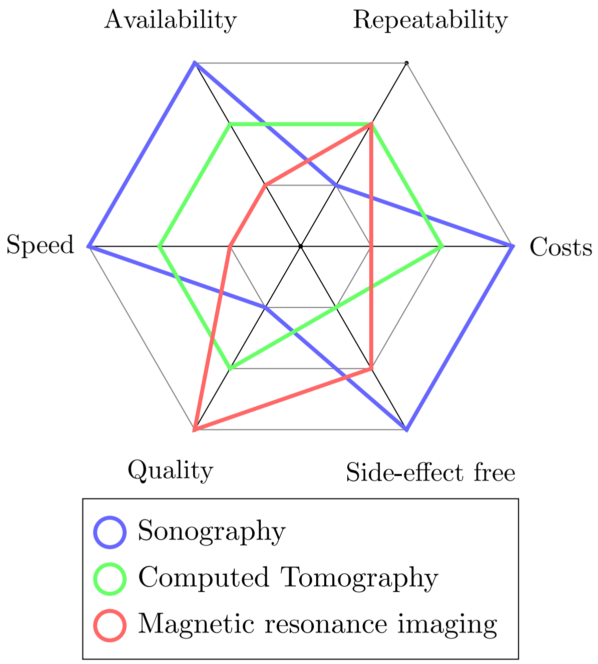


Figure 2.3: A simplified overview of the trade-offs between the most common imaging techniques for liver diagnostics. Ranking is only relative and based on qualitative descriptions found in literature (O'Neill et al., 2015; Hann et al., 2000).

2.2 Computed Tomography

Computed tomography creates medical 3D images by continuously moving an X-Ray imaging system around the patient. A computer program is then used to combine the 2D images taken from multiple angles into a 3D reconstruction (Maier et al., 2018). The resulting image shows the internal structure of the body, including abnormalities, such as diffuse changes or focal lesions, based on different tissue radiodensities. Injectable contrast media can be used to highlight the vascular system, show organ function, bleeding, and barrier disruptions. CT potentially offers higher spatial resolution than MRI or sonography, especially for hard tissues, at the cost of higher radiation exposure. As image noise is increased with reconstructions at higher resolution, and is reduced by higher X-ray intensities and longer scanning times, a balance needs to be struck between image quality and radiation dose. In comparison, MRI produces 3D images without radiation, allows for more flexible temporal-spatial resolution tradeoffs, and has higher soft tissue fidelity (Lin and Alessio, 2009), which allows for the differentiation between benign and malignant growths in the liver setting, and shows higher sensitivity for small growths despite lower spatial resolution (Alabousi et al., 2021). However, MRI scans are more time intensive, costly, and less widely available (O'Neill et al., 2015), and are contraindicated for patients that have electronic or metal implants, such as a pacemakers or orthopedic implants. CT scans are therefore the standard of care for many diagnostic plans, as they provide appropriate diagnostic insight, speed, price and availability (Eisenhauer et al., 2009).

2.3 The Liver

The liver is one of the largest organs and the largest gland of the body. It maintains and supports a variety of important body functions, such as metabolic processes, the storage of nutrients, the removal of waste products, toxins, and pathogens, and the synthesis and secretion of important proteins and hormones. Due to these diverse and important functions, diseases of the liver often have high morbidity and mortality rates (Sanyal et al., 2018).

The liver is asymmetrically shaped and located in the abdominal cavity. Most of its volume is in the right upper abdominal quadrant, reaching into the left upper abdominal quadrant due to its large size. It is comprised of multiple lobes that are connected to the gastrointestinal tract via the bile duct and gallbladder. These are also connected to the vascular system via the hepatic artery from the abdominal aorta, via the hepatic vein leading to the inferior vena cava, and via the portal vein leading to other organs, such as the gastrointestinal system, the pancreas, and the spleen (Qin and Crawford, 2018).

2.4 Deep Learning for Image Segmentation

Automatic segmentation of medical image data is becoming increasingly important in the context of medical image analysis (Ritter et al., 2011) and as an important preprocessing step in the field of radiomics (Timmeren et al., 2020).

This section briefly describes image segmentation in general, followed by an outline of the currently most commonly employed image segmentation technique, convolutional neural networks (CNNs). The loss and performance metrics applied for neural networks (NNs), as well as a brief overview of the concept of regularisation and three common regularising techniques, are presented afterwards.

2.4.1 Image Segmentation

The (automatic) process of dividing an image into partitions based on some criterion, such as ontological class, is called image segmentation. It is common to many applications, for instance, autonomous vehicles require semantically segmented street scenes (Chougula et al., 2020), and content-based image retrieval systems create abstractions of image contents via semantically segmented images (Manisha et al., 2020).

Another important field of application is the medical domain with a general need for cheap and effective preventive healthcare and reliable methods for tracking disease progression. Image segmentation can be used to locate or measure the volume of structures in CT or MRT scans, promising high quality and repeatability at a low cost and little time (Sutton et al., 2020). For example, tumor volume is superior to tumor diameter as a diagnostic criteria when tracking disease progression in liver cancer. It is too time consuming when measured manually, but can be computed with high precision and relative ease automatically. Incidentally, the liver provides a good testing ground for these approaches due to its well-defined radiological diagnostic criteria (Eisenhauer et al., 2009; Vander Kooi et al., 2018; O'Neill et al., 2015).

Image segmentation can be divided into two sub-types: semantic segmentation and instance segmentation. Semantic segmentation assigns each pixel to its respective class, while instance segmentation further differentiates between different instances of the same class that appear in an image.

Image segmentation has traditionally been solved via computer vision approaches, such as clustering or region-growing methods. These rely on assumptions about the properties of pixels of the same class, e.g. their spatial proximity or similarity in colour and brightness. In this context, information about the potential borders of class instances is provided by edge detectors. However, with the rise of deep learning, and in particular the invention of CNNs, AI-based techniques have largely superseded classical computer vision. With NNs image pre-processing steps and feature extraction are often omitted and the raw image is fed into the models, since they are able to learn salient image features from the training data themselves. In fact, CNNs are able to extract a hierarchy of features, from low-level edges over patterns to high-level abstract objects. CNNs have been successfully employed for various image segmentation tasks. The following section introduces the core concepts of CNNs and their aptitude for image processing.

2.4.2 Convolutional Neural Networks

From the perspective of machine learning, image segmentation is a classification problem in which every image pixel is assigned to one of several categories or classes. State-of-the-art image segmentation approaches employ convolutional neural networks (CNNs), a variant of deep neural networks (NNs) that use convolutional kernels instead of the traditional weight matrices for linear transformations.

After each such linear transformation, a non-linear activation function is used to transform the neuron's output. It is due to these activation functions that neural networks become highly-effective non-linear function approximators. A commonly used activation function employed in the hidden layers is the rectified linear unit (ReLU). ReLU maps all negative inputs to 0 while positive inputs are unaltered: $\text{ReLU}(x) = max(0, x)$. It has largely superseded the softplus activation, its differentiable variant, which is defined as:

$$\text{Softplus}(x) = \ln(1 + e^x)\,. \tag{2.1}$$

The differentiability of softplus and the fact that negative values are not "trapped" by a zero gradient make it attractive from a theoretical standpoint. However, ReLU is in practice not only computationally cheaper, but the sparsity induced by it's activations has been shown to aid the training process (Glorot et al., 2011).

The model's output is produced by the activations of the last layer in the network and common choices include sigmoid and softmax activations. The sigmoid, or logistic, function

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.2}$$

squashes the incoming value z to a score in $[0, 1]$. This is a common setup for binary classification, where the probability of one class is predicted and the probability of the other class assumed to be $1 - \sigma(x)$. The softmax function

$$\text{Softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.3}$$

is another very common choice, that can be used in both binary and multi-class classification. The incoming raw and unscaled values of the vector $\mathbf{z}$ that are associated with a respective class are also called logits. Softmax maps these logits to scores in the range of $[0, 1]$ which sum up to 1. The output thus conforms to the characteristics of a discrete probability distribution.

The convolution operation is able to learn local patterns, that is to say, patterns found in $m \times m$ (or $m \times m \times m$ in the case of 3D images) windows of the input. Convolution kernels are slid over the input with some specified padding, so each kernel processes local image regions, and the resulting values are stitched together into a (smaller) output image. This process allows the model to learn translation-invariant patterns, because a kernel that detects a specific pattern can do so regardless of the position of the pattern in the input image. Thus, CNNs possess an intrinsic generalisation power as to the location of learnt patterns. Moreover, because convolutions operate recursively on the output of earlier convolutions, they are able to learn ever-more abstract features that span ever-larger portions of the original image. In this way, a spatial hierarchy of patterns can be learnt, as is visible, for instance, in the activation atlas. (Carter et al., 2019) The assumption that translation-invariance, locality, and hierarchy of patterns are essential features in the visual domain has already been used in traditional image segmentation algorithms, as described in Section 2.4.1. They are also expressed in the inductive biases provided by the convolution operation and the architecture of CNNs, making them highly suitable for the processing of images.

CNNs are able to perform both image classification and segmentation. Image classification requires a flattening layer at the end, in which the spatial information is discarded. This operation is followed by one or more linear transformations with respective activation functions. An adequate activation function at the end produces either a class score or a class distribution for the given input image. Image segmentation, on the other hand, requires an image-to-image CNN, since each pixel is mapped onto a corresponding value. This computation is frequently achieved via fully-convolutional encoder-decoder models. An initial contracting encoder, or downsampling, path, composed of alternating convolutions, activation functions, and pooling operations, produces a small, i.e. low resolution, feature map. The following expanding decoder, or upsampling path, which reconstructs the output image, involves multiple layers comprising convolution and activation functions as well as upsampling operators. In addition, skip connections are commonly used to allow the decoder to access high-resolution, fine-grained features learnt in the encoder.

2.4.3 Losses and Performance Metrics

For training an NN, an objective function, or loss, $\mathcal{L}(\mathbf{w} \mid \mathcal{D})$ needs to be defined. This loss is used to compute the model error, and thus guides the optimisation process towards a plausible set of model weights.

From a probabilistic perspective on deep learning, an NN is understood as a model $f(\mathbf{x}, \mathbf{w})$ with input $\mathbf{X}$ and weight parameters $\mathbf{w}$, where for classification the model output $p(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{w})$ is a categorical distribution over a set of classes. In the case of binary classification, the output more specifically follows a Bernoulli distribution. The likelihood function given a training data set $\mathcal{D} = \{\mathbf{x}_t, \mathbf{y}_t\}_t^N$ is $p(\mathcal{D} \mid \mathbf{w}) = \prod_i p(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{w})$. Maximising this likelihood function gives the maximum likelihood estimate (MLE) of the network parameters, i. e. a point estimate of the weights $\mathbf{w}$:

$$\begin{aligned} \mathbf{w} &= \arg\max_{\mathbf{w}} \log P(\mathcal{D} \mid \mathbf{w}) \\ &= \arg\max_{\mathbf{w}} \sum_{i=0}^{N} \log P(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{w})\,. \end{aligned} \tag{2.4}$$

Note that the logarithm is added in order to increase numerical stability. Since the logarithm is a strictly monotone function, i. e. it is order-preserving, the maximum to be computed via MLE stays the same. Unfortunately, no closed-form solution exists for Equation 2.4, instead, MLE is achieved iteratively via stochastic gradient descent optimisation. Assuming that both the network as well as the loss are differentiable, the gradient $\frac{d\mathcal{L}}{d\mathbf{w}}$ can be calculated via backpropagation and used in gradient descent.

Maximising the above likelihood is equivalent to minimising the negative log likelihood (NLL), or the so-called cross-entropy loss. The cross-entropy loss thus arises naturally from the MLE of a classification task and is a

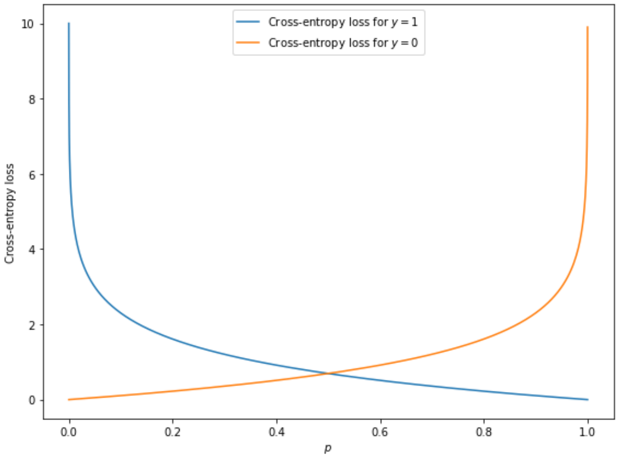


Figure 2.4: Binary cross-entropy loss behaviour

popular choice as a loss function for classifiers. In general, the multi-class cross-entropy loss

$$\mathrm{H}(p,q) = -\mathbb{E}_p[\log q] = -\sum_{x\in\mathcal{X}} p(x)\log q(x)\,, \tag{2.5}$$

with $\mathbb{E}_p$ the expected value with respect to model output p, is an information theoretic measure which is used to compute the difference between two probability distributions p and q.

$$\mathrm{CE} = \frac{1}{N}\sum_{t=1}^{N} -y_t\log(p_t) + (1-y_t)\log(1-p_t) \tag{2.6}$$

Equation 2.6 constitutes the binary version of the cross-entropy loss for a one-hot encoded ground truth vector y. It can be used in conjunction with a single output neuron whose activation function is set to sigmoid, as defined in Equation 2.2. Cross-entropy loss requires the model predictions to lie between 0 and 1, and it increases exponentially with increasing divergence between predicted value and actual class. Figure 2.4 shows the range of the binary cross-entropy loss function given true and false observations. The graph demonstrates that predictions far away from the ground truth class, i. e. confidently wrong predictions, are heavily penalised.

Another loss function that is often employed in classifiers is the Brier score. It computes the mean squared error between the model's predicted probability distribution and the ground truth labels:

$$\text{Brier} = \frac{1}{N} \sum_{t=1}^{N} (p_t - y_t)^2 \,. \tag{2.7}$$

Since both the cross-entropy loss and the Brier score are applied on single voxels and then averaged, training a model on a data set with unbalanced classes might be dominated by the most frequent class. This can be counteracted by weighting the loss differently for each class. An alternative loss formulation which is especially suited for imbalanced data sets is the soft Dice loss, which works on whole image patches instead of single voxels. It is derived from the Dice coefficient (Sørensen, 1948; Dice, 1945):

$$\text{Dice}(X, Y) = \frac{2 \mid X \cap Y \mid}{\mid X \mid \cup \mid Y \mid} \,, \tag{2.8}$$

which measures the similarity between two samples X and Y. Its range is $[0, 1]$ with 1 indicating a perfect overlap. The Dice coefficient is a widely used metric for computing the similarity, or more concretely the spatial overlap, between two images. It is highly similar to the Jaccard similarity coefficient, also called Intersection over Union (IoU) (Jaccard, 1912):

$$\text{IoU}(X, Y) = \frac{\mid X \cap Y \mid}{\mid X \cup Y \mid} \,. \tag{2.9}$$

In fact, there exists a bijection between both coefficients, i. e. $Dice = \frac{2\ \text{IoU}}{1+\text{IoU}}$, with the Dice coefficient always being larger than or equal to the IoU. The Dice loss lends itself naturally to semantic segmentation tasks in which the model is optimised for the Dice coefficient as its evaluation metric.

2.4.4 Regularisation

A model that overfits its training data does not generalise well, which means it heavily depends on training data-specific (noise) patterns and consequently performs poorer on held-out test data (Szegedy et al., 2014). In general, using the MLE during training tends to result in models that overfit the training data $\mathcal{D}$. MLE can also be seen as a special case of maximum a posteriori estimation (MAP) using a uniform prior over the weights. Since overfitting usually correlates with diverging, higher-valued weights, choosing the weight prior to instead follow a Gaussian or Laplace distribution, has a regularising effect on the model. Multiplying the likelihood with a prior distribution $p(w)$ on the weights is, by Bayes theorem, proportional to the posterior distribution, i. e. $p(\mathbf{w} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \mathbf{w})p(\mathbf{w})$. Maximizing this posterior corresponds to the MAP estimate of $\mathbf{w}$:

$$\begin{aligned} \mathbf{w} &= \arg\max_{\mathbf{w}} \log p(\mathbf{w} \mid \mathcal{D}) \\ &= \arg\max_{\mathbf{w}} \log p(\mathcal{D} \mid \mathbf{w}) + \log p(\mathbf{w}) \,. \end{aligned} \tag{2.10}$$

This has a regularising effect and can prevent overfitting. Thus, the optimisation objective stays the same, there is just an additional regularisation term from the prior. With a Gaussian prior, this yields weight decay. Given a Laplace prior, L-2-regularisation is achieved.

Another common regularisation technique is dropout, in which during training a random subset of neurons per forward pass is dropped out, removing the neuron with all of its incoming and outgoing connections. The regularising effect is commonly interpreted to stem from dropout's capacity to impede training data-specific co-adaptations of units. Also, it introduces further stochasticity into the training process, which is thought to make the network more robust, in general.

The problem of overfitting, however, is not as pronounced in the setting of binary semantic segmentation, where each pixel or voxel can be interpreted as a training sample for which a corresponding ground truth is available. Training models on only two CT volumes, for example, already involves a massive amount of training data and leads to nearly decent segmentation performance, as can be seen in Section 5.3.

2.5 Calibration and Uncertainty Types

Probabilistic classifiers applied in the real world are expected to deliver robust predictions that are indicative of the true data distribution, or ground truth correctness likelihood. This need gave rise to the notion of model *reliability* or model *calibration*. This section introduces these notions and presents some calibration metrics for binary classifiers. It then describes the decomposition of general ”predictive model uncertainty” into two types: aleatoric and epistemic uncertainty.

2.5.1 Model Calibration

A binary probabilistic classifier outputs a single prediction $p \in [0, 1]$ representing the probability assigned to its respective class. For perfectly calibrated predictions, the predicted class probability p needs to match its true proportion. In other words, whenever the model predicts a class with probability 0.9 a total of 100 times, the prediction should be correct for 90 of those samples. In general, the notion of calibration follows a frequentist perspective on uncertainty, whereby class probabilities are expected to match their respective long-time observed frequencies, which are typically approximated on a held-out test set. Note that perfect calibration does not imply accurate predictions, e. g. always predicting the class' empirical accuracy results in perfect calibration but uninformative predictions.

Model calibration is the process of turning the output scores of a classifier into reliable probabilistic class probabilities. Some common approaches relevant to this work are outlined in Section 3.

Calibration Metrics Given a probabilistic classifier, its calibration can be visualised and measured with a number of diagnostic tools.

A popular method for illustrating a classifier's calibration are reliability diagrams (DeGroot and Fienberg, 1983). Reliability diagrams plot the expected accuracy of samples as a function of the model's confidence, where the diagonal represents perfect calibration such that the model's confidence aligns with the expected accuracy. However, the posterior distribution representing the true accuracy is unknown. Hence, one usually approximates the observed accuracy on a test set, i. e. the fraction of correctly classified test samples is computed and plotted against the model's respective confidence.

Note that this formulation of reliability diagrams is constrained for binary classifiers, but adaptations for multi-class classifiers have been proposed (Guo et al., 2017; Kock et al., 2022). In particular, this work uses Kock et al.'s extension which adds boxplots that reveal the otherwise hidden bin-internal variances. An example reliability diagram with this extension can be seen in Figure 2.5.

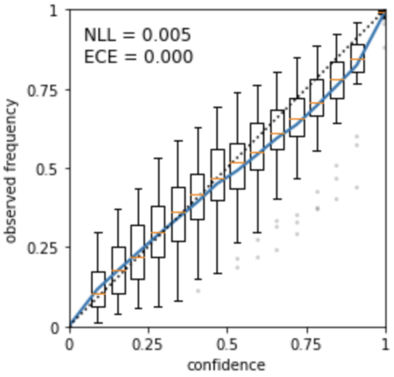


Figure 2.5: Example reliability diagram with boxplots

The reliability diagram's binning induces an estimator for the binary expected calibration error (ECE) (Naeini et al., 2015), which can be computed as the average of the bins' accuracy-confidence differences, weighted by the number of samples per bin. The maximum calibration error (MCE) analogously represents the highest bin-wise divergence between the ground truth accuracy and predicted confidence. Other calibration metrics include the negative log likelihood, as shown in Equation 2.5, and the Brier score, depicted in Equation 2.7, since both constitute strictly proper scoring rules. The notion of proper scoring rules will be described in the following.

Proper Scoring Rules Scoring rules measure the quality of probabilistic predictions, assigning numerical scores to predictive distributions. They are functions $\mathrm{S}(p(x),(y,x))$, where $p(x)$ is the model prediction for input x and $(y,x) \sim q(y \mid x)$ with q the true distribution of (y,x)-tuples. Classifiers are evaluated by applying a scoring rule on multiple samples and comparing the average scores. Proper scoring rules are defined by $\mathrm{S}(q,q) \leq \mathrm{S}(p,q)$ for all p and q. Scoring rules are strictly proper if this holds with equality iff $p(x) = q(y \mid x)$.

Strictly proper scoring rules facilitate calibrated probabilistic predictions (Mehrtash et al., 2020), since at their minimum and given infinite training data, they guarantee perfectly calibrated model predictions. In practice, however, this minimum is not reached and overfitting still frequently occurs, thus resulting in less calibrated scores. Both cross-entropy loss and Brier score, introduced in Section 2.4.3, constitute proper scoring rules.

2.5.2 Aleatoric and Epistemic Uncertainty

Senge et al. (2014) and later Kendall and Gal (2017) propose to decompose the overall uncertainty of a machine learning model into two subcategories: aleatoric uncertainty and epistemic uncertainty. Aleatoric, or statistical, uncertainty represents data-inherent statistical variability or randomness, hence its derivation from the Latin term *alea* - dice. In the setting of image segmentation, aleatoric uncertainty thus stems from label noise in the source distribution, either due to inherent randomness of the data-generating process, or constrained observability thereof. Label noise can originate from visual difficulty to the point of visual ambiguity in the underlying data, be it sensor noise or occluded objects, and from divergent annotation regimes, e.g. whether or not to include the vena cava inside the liver mask. Both conditions lead to inter-annotator disagreement, where semantically similar pixels are labelled differently, resulting in ground truth samples with intrinsic stochasticity - or uncertainty. Consequently, the data-inherent aleatoric uncertainty a model reports cannot be reduced with longer training or more training data. In the case of machine leaning, it can only be reduced to zero if a given sample provides sufficient information to uniquely determine the correct label with a probability of 1 (Kull and Flach, 2015). Epistemic or systematic uncertainty, on the other hand, represents a model's lack of knowledge about the underlying input data-generating process, i. e. the model's epistemic state. It is therefore also called model uncertainty, since it describes the lack of knowledge about which model best explains the given data. It can in principle be explained away by e.g. looking at more data.

Aleatoric uncertainty can be further decomposed into homo- and heteroscedastic uncertainty. The former is constant over all samples and depicts the overall noise level of the training data, whilst the latter is input-dependent. This thesis is concerned with heteroscedastic aleatoric uncertainty, since it provides human-interpretable indications for label-noisy regions and it lends itself to use as a loss attenuator, as will be described in Section 3.5.

Uncertainty estimates are usually visualised in so-called uncertainty maps, which plot for each pixel the model's correspondingly predicted uncertainty value. Qualitative investigations of such maps for the overall prediction uncertainty, as well as for both subtypes of uncertainty are commonly reported. Based on the definition of aleatoric and epistemic uncertainty, several expectations hold: Aleatoric uncertainty should light up in areas where the

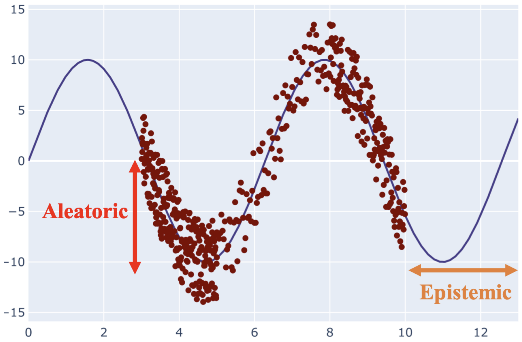


Figure 2.6: Illustration of aleatoric and epistemic uncertainty, image taken from Abdar et al. (2021).

segmentation is particularly difficult even for humans, thus in regions where annotations are likely to involve label noise. This happens in particular at class boundaries and in image parts with visually ambiguous objects, e.g. due to motion artefacts or blurry regions. Also the use of different annotation regimes whilst creating ground truth images is expected to result in aleatoric uncertainty. Epistemic uncertainty, on the other hand, is expected to be present in samples whose patterns were either rare in the training set or that are completely dissimilar from training samples, i. e. ones that are likely to be drawn from a different distribution than the training distribution. Such samples are also called out of distribution (OOD), in contrast to in distribution (ID) data. Figure 2.6 illustrates this distinction. The training data covers some range on the x-axis, data out of this range is considered OOD and would be assigned high epistemic uncertainty while data inside the training range is closely scattered around the true underlying function, still exhibiting some noise which constitutes aleatoric uncertainty.

Related Work

Quantifying uncertainty in the context of medical image analysis is a challenging and important task, as already described in Section 1.1. Numerous uncertainty quantification approaches have been proposed in recent years, that calibrate the predictions of deep learning models or devise new ways to derive an uncertainty estimate from a neural network. This work aims to evaluate the uncertainty decomposition quality of two approaches for the task of binary liver segmentation in CT scans. Thus, in the following, the aim is to provide a systematic overview of some of the most prominent uncertainty quantification approaches, with a focus on Bayesian neural networks and loss attenuation.

Firstly, a classifier's inherent uncertainty is introduced as a baseline, followed by various calibration methods that aim to improve on the quality of this baseline predictive uncertainty. While such post-hoc calibration methods require no change in model architecture or training procedure, they are restricted insofar as only being able to work with the output of the original model. Approaches that improve a model's uncertainty estimate at training-time, on the other hand, potentially benefit from "deeper" knowledge of the problem and are able to directly interact with the optimisation process itself. Therefore, various other methods for estimating a neural network's uncertainty are presented afterwards. Deep ensembles and Bayesian neural networks constitute approaches that compute a model's uncertainty over the predictive distribution's variability. Afterwards, the concept of loss attenuation is introduced and several approaches that employ a learnt uncertainty value are outlined. Lastly, some prominent methods for decomposing the uncertainty of the predictive distribution into aleatoric and epistemic uncertainty are presented.

3.1 Baseline Uncertainty Quantification

Probabilistic classifier NNs output categorical predictive distributions which inherently contain some form of predictive uncertainty. More specifically, their probabilistic predictions can be interpreted as capturing learnt aleatoric uncertainty, while measuring epistemic uncertainty would require an additional confidence estimate for the output probabilities (Hüllermeier and Waegeman, 2021). Despite this theoretical consideration, an NN's predictive distribution has been proposed by Hendrycks and Gimpel (2016) as a baseline for predicting misclassified samples and for detecting OOD samples. The authors found that a classifier's maximum softmax score correlates with both model performance and the probability of the respective sample being OOD. Moreover, computing the KL divergence between the softmax distribution and the uniform distribution delivers similar results. Other approaches calculate the entropy of the softmax distribution (Williams and Renals, 1997) or compute the difference between the highest and second-highest softmax score (Monteith and Martinez, 2010).

A neural network's softmax distribution indeed constitutes a strong baseline as uncertainty estimate whenever the model is trained with a proper scoring rule, such as the cross-entropy loss. The soft Dice loss, on the other hand, increases segmentation performance at the cost of overconfident predictions (Mehrtash et al., 2020; Guo et al., 2017; Bertels et al., 2021; Sander et al., 2019). Not only the choice of loss function impacts a model's calibration. Guo et al. (2017) have also shown that deeper models, i. e. models with more layers, tend to produce *overconfident* results, as well. Meanwhile, Minderer et al. (2021) revisited these observations and found that deeper models are indeed slightly more overconfident on ID data, but when evaluated on OOD data, this trend is reversed with deeper models providing more calibrated uncertainty estimates. Moreover, Guo et al. (2017) show that regularisation has a positive impact on calibration in general.

3.2 Post-hoc Calibration

In order to mitigate overconfident and, more generally, unreliable predictions, a number of approaches have been devised to calibrate deep learning models, either at training time or post-hoc. Calibrating a model means rescaling the model output scores to respectively calibrated probability scores, as introduced in Section 2.5.1.

Two popular calibration approaches are isotonic calibration (Zadrozny and Elkan, 2002), in which a piecewise-constant non-decreasing function is fitted as calibration map, and Platt scaling (Platt, 1999), or logistic calibration, which runs a logistic regression on the classifier scores.

Menawhile, temperature scaling, the simplest special form of logistic calibration, has been shown to achieve remarkably good results for neural networks applied in various task settings (Guo et al., 2017). It works by optimising a scalar temperature T by which all logits are divided, as shown in Equation 3.1.

$$\hat{\sigma}_i(j; z_i) = \frac{exp(z_{i,j}/T)}{\sum_{k=1}^{K} exp(z_{i,k}/T)} \qquad T > 0 \tag{3.1}$$

Temperature scaling is a simple and cheap operation, since minimising the cross-entropy loss on a test set over T is a one-dimensional convex optimisation problem. It has been shown to successfully calibrate medical image segmentation models (Kock et al., 2022; Ding et al., 2021).

ODIN The Out-of-distribution Detector for Neural Networks (ODIN; Liang et al., 2017) employs temperature scaling and input perturbation in order to separate the maximum softmax scores for ID and OOD inputs. During detection, the input image is firstly preprocessed, and then a calibrated softmax score is computed whose maximum value is compared to some threshold δ.

Input preprocessing means adding small perturbations of magnitude ϵ to the input image x that increase the maximum softmax score:

$$\tilde{x} = x - \epsilon \, \mathrm{sign}(-\nabla x \log S_{\hat{y}}(x; T)) \,, \tag{3.2}$$

with sign denoting the element-wise indication of the sign of the gradient. This step can be done unsupervisedly, as the perturbation direction is computed by backpropagating the gradient of the cross-entropy loss w.r.t. the input. This method can be seen as a variation on the fast gradient sign-method (Goodfellow et al., 2015) in reversed direction. If one chooses the perturbation parameter ϵ to be sufficiently small, the model output regarding to predicted class does not change. This perturbation has a stronger effect on ID inputs because of a larger gradient of log-softmax scores for those inputs, a phenomenon that has also been demonstrated in Huang et al. (2021). Thus, after preprocessing, images $\tilde{x}$ are more separable into ID and ODD.

ODIN does not require retraining the original model, since temperature scaling is a post-hoc calibration method and input preprocessing is performed on the images to be inferred. The authors found a correlation between difficult-to-classify samples and OOD samples, thus both aleatoric as well as epistemic uncertainty appears to be captured by this method.

3.3 Deep Ensembles

Instead of employing a model's single softmax prediction as an overall uncertainty estimate, deep ensembles provide a frequentist approach to uncertainty quantification by observing the output variance of multiple trained models with equal architecture (Lakshminarayanan et al., 2017). Their prediction variance stems from the randomness of both weight initialisation and training process. The authors show that the uncertainty estimates derived from deep ensembles are of high quality, i. e. they reliably identify OOD samples and indicate inaccurate predictions. This finding is supported by others who found deep ensembles to lead to even better reliability than post-hoc calibration methods and different variants of Bayesian neural networks (Ovadia et al., 2019). However, the computational effort for both training and inference, when no parallelisation is used, increases linearly with the number of models.

While deep ensembles are often viewed as frequentist approaches to uncertainty quantification (Lakshminarayanan et al., 2017; Ovadia et al., 2019), recent work interprets deep ensembles as essentially performing Bayesian model averaging, thereby approximating the posterior predictive distribution of a Bayesian neural network (Wilson and Izmailov, 2020). According to the authors, ensembles provide functionally diverse results by each ensemble member representing a different "basin of attraction" in the loss landscape. This multimodal Bayesian average lets deep ensembles perform even better than single-basin marginalisation approximations. The following section describes Bayesian neural networks and popular approximation approaches in more detail.

3.4 Bayesian Neural Networks

Following a Bayesian perspective on uncertainty, Bayesian neural networks (BNNs) model distributions over the model weights, which in turn result in a predictive distribution on the output. They thus constitute a non-deterministic variant of vanilla NNs. Instead of optimising the network weights directly, BNNs average over all possible weights, a process called marginalisation. Several metrics can then be employed for estimating the model's uncertainty given its predictive distribution.

Both MLE and MAP estimation yield point estimates of a model's parameters, thus the training of a vanilla NN works by optimising a single setting of weight values w. As introduced in Section 2.5.2, epistemic uncertainty represents the uncertainty about which model constitutes the true data-generating function. Therefore, in order to model epistemic uncertainty, the weights of an NN might be modelled as distributions $p(w \mid \mathcal{D})$ instead, explicitly representing the uncertainty about the model's own parameters. Bayesian inference, in contrast to MLE and MAP, provides a way to compute this full posterior probability distribution over the weights given a data set $\mathcal{D} = (x_i, y_i)_i$:

$$p(w \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid w) \cdot p(w)}{p(\mathcal{D})}\,. \tag{3.3}$$

During training, the aim is to estimate $p(w \mid \mathcal{D})$, which captures the set of plausible model parameters given the training data. In order to do so, the prior distribution $p(w)$, which stands for the belief about the weights before observing any training data, is updated over the network weights by multiplying the likelihood $p(\mathcal{D} \mid \mathbf{w})$ with the prior. The likelihood is exactly the same as in the frequentist approach for vanilla NNs, it quantifies how well the observed training data can be explained by a specific setting of w. In order to arrive at a distribution, one needs to marginalise over all possible parameter settings, since

$$p(\mathcal{D}) = \int p(\mathcal{D} \mid w) p(w) dw. \tag{3.4}$$

The final posterior distribution over the weights then allows to make predictions by, again, marginalising over the weights:

$$\mathbb{E}_{p(w|\mathcal{D})}[p(y \mid x, w)] = \int p(y \mid x, w) p(w \mid \mathcal{D}) dw\,. \tag{3.5}$$

The above equation shows the Bayesian model averaging employed during inference, where given an input sample x each possible configuration of weights, weighted according to the posterior distribution, makes a prediction about the unknown label. Note that the posterior assigns high density to weights that better explain the data $\mathcal{D}$. It is in this way that the resulting predictive distribution takes the BNN's weight uncertainty into account. Therefore, the model's uncertainty can in principle be derived from its predictive distribution. As an aside, this inference process is equivalent to averaging predictions from an infinite ensemble of neural networks. Since marginalising over all possible weights during inference is intractable, however, the predictive distribution is usually approximately computed via Monte Carlo (MC) integration:

$$p(y \mid x, \mathcal{D}) \approx \sum_{l=1}^{L} p(y \mid x, w^{(l)}) \,, \quad \text{where } w^{(l)} \sim p(w \mid D) \,. \tag{3.6}$$

This same practical problem appears in the training phase, when marginalising over all possible weight combinations. Since the probability distributions for weights in neural networks are highly complex and extremely high-dimensional, no closed-form computations exist. Therefore, numerous methods have been proposed in the literature for approximating the posterior $p(w \mid \mathcal{D})$, of which just some are briefly mentioned in the following.

Markov chain Monte Carlo Estimating the posterior weight distribution by sampling $w \sim p(w \mid D)$ via Markov chain Monte Carlo methods allows to approximately compute the model prediction (Gamerman, 1997; Papamarkou et al., 2022).

Variational inference Alternatively, the BNN's posterior can be approximated via variational inference. In general, variational inference entails approximating an intractable probability distribution p by another simpler, tractable distribution q via optimisation by finding $\text{argmin}_\theta \mathcal{KL}(q \mid\mid p)$. The surrogate distribution q is then used instead of p. This method has been introduced for approximating the posterior weight distribution in BNNs in Blundell et al. (2015) and termed "Bayes by Backprop". Gaussian distributions are commonly used variational distributions in this setting, which introduce an additional learnable parameter per weight, replacing w with (μ, σ).

MC dropout variational inference Dropout has originally been introduced as a regularisation technique during training, but Gal and Ghahramani (2016) have demonstrated that using dropout at test-time, as well, allows for BNN uncertainty quantification where the model outputs are MC samples and interpreted as samples of the predictive distribution as in Equation 3.6. Reinterpreting dropout in this way, the approximating variational distribution over the weights is a Bernoulli distribution. This approach does not introduce additional learnable parameters. However, the prediction requires several forward passes, thereby increasing the computation time linearly in the number of samples drawn for inference. In practice, the test time can be reduced with concurrent forward passes or decoder-only dropout, for which half of the model's forward pass can be saved and thus needs not be recomputed. The latter is a common and well-performing alternative (Kendall et al., 2017), but lacks the clean mathematical justification given for models employing dropout after each layer. In general, MC dropout retains the computational efficiency of vanilla neural networks, i. e. training via stochastic gradient descent and backpropagation. Due to this simplicity and relatively low computational burden, MC dropout is frequently used for approximating BNNs (Mobiny et al., 2021). In the domain of medical imaging, MC dropout has been used successfully as uncertainty quantification method (Leibig et al., 2017; Yang et al., 2016), with Ng et al. (2020) showing that Bayes by Backprop and MC Dropout lead to similar segmentation and uncertainty performance for the semantic segmentation of cardiac MRI scans.

Another advantage of BNNs is their inherent regularisation, whereas MLE point estimates computed for vanilla NNs can lead to severe overfitting. In line with this observation, BNNs have been shown to be trainable on small data sets without overfitting (Depeweg et al., 2018).

Given the predictive distribution of a BNN, various measures for computing the model's uncertainty have been proposed. Some of these will be described in the following.

Uncertainty Metrics for Predictive Distributions

The variability of the predictive distribution of a BNN constitutes the model's uncertainty, and several methods for computing the predictive uncertainty have been proposed. This section presents three common metrics for deriving a model's uncertainty from a BNN's predictive distribution: predictive entropy, predictive variance, and mutual information.

Given a BNN's predictive distribution $P(y \mid x, \mathcal{D})$, approximated by T MC samples, as shown in Equation 3.6, the predictive entropy

$$\mathrm{H}[y_i \mid x_i, \mathcal{D}] \approx \sum_{c=1}^{C} (\frac{1}{T} \sum_{t=1}^{T} p(y_i = c \mid x_i, W_t)) \log(\frac{1}{T} \sum_{t=1}^{T} p(y_i = c \mid x_i, W_t)) \tag{3.7}$$

and the predictive variance

$$\mathrm{Var}(p(y_i \mid x_i, W_1), \ldots, p(y_i \mid x_i, W_T)) \tag{3.8}$$

are commonly used metrics that straightforwardly derive predictive uncertainty estimates.

An alternative metric is the mutual information (MI) between the model parameters and the model prediction:

$$\mathrm{MI}[y_i, W \mid x_i, \mathcal{D}] \approx \mathrm{H}[y_i \mid x_i, \mathcal{D}] - \mathbb{E}[\mathrm{H}[y_i \mid x_i, W]]\,. \tag{3.9}$$

The MI metric can be approximated by subtracting the expected value of the entropy of the model predictions across samples from the entropy of the expected predictive distribution, or the predictive entropy, as shown in Equation 3.7. Intuitively, the mutual information (MI) metric computes the information gain about the model parameters when the ground truth label is known. It can, thus, be interpreted as explicitly measuring the BNN's epistemic uncertainty.

Kendall and Gal (2017) report computing the predictive entropy for measuring the model uncertainty of their BNN, while Bayesian SegNet, a MC dropout-approximated BNN for semantic segmentation, computes uncertainty estimates using the predictive variance (Kendall et al., 2017). Nair et al. (2020) compare the performance of all three presented metrics, as well as the heteroscedastic uncertainty neuron, described in Section 3.5.4, for the task of multiple sclerosis lesion segmentation on MRI sequences. The authors found no differences between the three BNN-based metrics, apart from the fact that the predictive variance results in values with smaller magnitude.

3.5 Learnt Loss Attenuation

Yet another approach to uncertainty quantification is loss attenuation. Instead of modelling epistemic uncertainty via model or weight uncertainty, as in deep ensembles and BNNs, the model's aleatoric uncertainty is encouraged to be implicitly learnt during training. Without knowing the target distribution's underlying aleatoric uncertainty, the uncertainty estimate can instead be learnt via the loss function. If the model is able to learn the input-specific amount of aleatoric uncertainty, it benefits from attenuating the loss for samples with particularly high aleatoric uncertainty. This allows the model to focus on learnable, i. e. easier, samples with consistent labels for approximating the underlying input data-generating function without being distracted by being forced to learn label noise.

3.5.1 DCA regularisation

Mihail et al. (2019) basically employ loss attenuation without explicitly mentioning its use. The authors introduce an additional loss term which constitutes the difference between a batch's average predicted confidence and the true accuracy, i. e. its calibration error. This loss attenuating term helps to calibrate the model, but cannot be returned at inference time, so that only the softmax distribution is observable for inferred samples. The authors show that their approach counteracts the overconfidence which is frequently learnt via the normal cross-entropy loss.

3.5.2 Learnt Confidence Estimate

DeVries and Taylor (2018a) and DeVries and Taylor (2018b) propose to learn a confidence estimate that can be applied for OOD detection and image-level quality estimation as well as semantic segmentation quality estimation. The authors employ a two-headed neural network that outputs a prediction $\mathbf{z}$ as well as a confidence score $c \in [0, 1]$ which has been passed through a sigmoid activation function. During training, the model receives "hints", i. e. the target probability distribution is added to the model prediction, proportional to its predicted confidence, as shown in Equation 3.10. A log penalty on the confidence score is included as an additional loss term in order to prevent the network from solely relying on the ground truth hints.

$$p = c \times \text{Softmax}(\mathbf{z}) + (1 - c) \times \text{Onehot}(\mathbf{y}) \tag{3.10}$$

The authors found the confidence score to often converge towards uniform scores for all samples. In order to prevent this, a budget hyperparameter is introduced which controls the weight of the confidence score penalty, adjusting it during training. If the confidence score penalty exceeds the budget threshold, the penalty is increased, incentivising the model to ask for hints, and vice versa. In this way, the confidence penalty tends towards the budget parameter and confidence scores converge towards their intended meaning.

In DeVries and Taylor (2018b), the authors compare their learnt confidence estimate approach with the maximum softmax probability as baseline, the output of Kendall and Gal's heteroscedastic uncertainty neuron, which is introduced in Section 3.5.4, and MC dropout for BNN, which is described in Section 3.4, on a skin lesion segmentation task. The resulting uncertainty maps indicate no difference between the methods for correctly segmented regions, with all approaches highlighting class boundaries. However, whilst most uncertainty estimates also highlight incorrectly segmented parts of the image, the heteroscedastic uncertainty neuron continues to only output a thin layer of uncertainty around the (in)correctly segmented object. DeVries and Taylor (2018a) compare the learnt confidence approach with thresholded maximum softmax probability as baseline and ODIN, see Section 3.2. For OOD detection, the authors found the confidence estimate to perform consistently better than the baseline and more reliable than ODIN.

3.5.3 Abstention Networks

Thulasidasan et al. (2019) also incorporate a learnt confidence score alongside the standard classification prediction and interpret it as an additional class. In their loss formulation, shown in Equation 3.11, the prediction for the so-called abstention class is normalised out of the real classes' predicted probabilities. This induces the desired loss attenuation behaviour, since predicting the abstention class decreases the first loss term. The second term includes an abstention penalty α which regulates the amount of abstention, thus preventing the model from abstaining on all samples.

$$p = (1 - p_{k+1})(-\sum_{i=1}^{k} t_i \log \frac{p_i}{1 - p_{k+1}}) + \alpha \log \frac{1}{1 - p_{k+1}} \tag{3.11}$$

The authors also demonstrate how α can be auto-tuned during the training process in order for the abstention class to robustly learn the model's confidence. In their training regime, an abstention-free initial period of training is followed by excessive abstention, which is then iteratively reduced to only apply on the most difficult training samples. The abstention scores were evaluated on an image classification task with artificial structured and unstructured label noise. The former was constructed by randomising the labels of images of a certain class or those which include some other consistent pattern, whereas unstructured label noise was introduced by changing the label of random samples. Abstention scores reliably indicate both types of label noise and consistently surpass a baseline model without abstention.

3.5.4 Heteroscedastic Uncertainty Neuron

Analogously to regression uncertainty, which can be represented as the variance of a Gaussian distribution over the model output, Kendall and Gal (2017) capture a classifier's predictive uncertainty via additional heteroscedastic uncertainty neurons, one for each class. The output of these neurons defines the variance of Gaussian noise which is placed over the logits as follows:

$$\mathbf{p} = \mathbf{z} + \sigma * \epsilon \,, \tag{3.12}$$

with $\mathbf{p}$ the final model prediction, $\mathbf{z}$ the logit vector, σ a diagonal covariance matrix defined by the output of the heteroscedastic uncertainty neuron, and $\epsilon \sim \mathbb{N}(0, \mathbb{I})$ a small noise parameter that is introduced via the reparameterisation trick (Kingma and Welling, 2014), which allows to backpropagate over sampled values.

Sampling in this way through the softmax over the logits smoothes the output scores, i. e. the scores are moved towards a uniform distribution $\mathcal{U}(1, \text{classes})$. Thus, the resulting score distribution has a lower loss than the original prediction if the model predicted a wrong class. This naturally leads to the effect of loss attenuation without the need for additional penalty terms. The model is not incentivised to attenuate on all samples, since pushing its output scores towards the uniform distribution increases the loss for correctly predicted samples.

Kendall and Gal compare their loss-attenuated model for multi-class semantic segmentation tasks against a baseline model without explicit uncertainty quantification capability as well as against a BNN's predictive distribution. The final segmentation performance of the loss-attenuated model is slightly higher than that of the baseline and of the BNN (IoU of 67.4, 67.1, and 67.2, respectively). Whether a model's uncertainty can reliably predict segmentation quality, i. e. the correlation between uncertainty and accuracy, can be evaluated by dropping all those samples whose predicted uncertainty lies below a threshold. Resulting precision-recall curves demonstrate that when removing pixels at various uncertainty thresholds, for both the loss-attenuated model's predictive uncertainty as well as for the predictive entropy of a BNN, uncertainty estimates correlate with accuracy, since precision increases as the number of uncertain samples decreases. Moreover, plotting the calibration reveals a slight increase in reliability for the loss-attenuated model as compared against the baseline, with a mean squared error of 0.003 and 0.005, respectively.

3.5.5 Heteroscedastic Logit Smoothing

Similar to Kendall and Gal's work, Neumann et al. (2018) implement a heteroscedastic neuron whose output is interpreted as a confidence score that directly smoothes the logits before going into the softmax. This approach has been termed "relaxed softmax" and can be interpreted as learnt heteroscedastic logit smoothing or learnt temperature scaling, because of its derivation from the post-hoc calibration method called temperature scaling, which is described in Section 3.2. Equation 3.13 depicts the smoothing and softmax calculation for some logit $z_{i,j}$ and a learnt smoothing factor α_i for sample i, where α_i can be interpreted as a more numerically-stable equivalent to the temperature in temperature scaling.

$$\hat{\sigma}_i(j; z_i, \alpha_i) = \frac{exp(\alpha_i z_{i,j})}{\sum_{k=1}^{K} exp(\alpha_i z_{i,k})} \qquad \alpha_i := \frac{1}{T_i} \tag{3.13}$$

The authors compare their approach to a baseline network with and without temperature scaling on the task of pedestrian detection with two data sets, the results are included in Figure 3.1. They found the proposed heteroscedastic logit smoothing model to output significantly more calibrated scores than the baseline models. More specifically, the overconfidence exhibited by the baseline model can be counteracted by post-hoc temperature scaling, but the heteroscedastic logit smoothing leads to more effective output calibration. However, the proposed approach results in a slightly underconfident network on one data set, in which case post-hoc linear scaling further improves model calibration.

3.5.6 Learnt Label Smoothing

Instead of smoothing the predicted logits according to the model's confidence as in heteroscedastic logit smoothing, see Section 3.5.5, the model's confidence score might also be used to smooth the ground truth labels, instead. Training with soft targets, also called label smoothing, has originally been introduced as a regulariser for neural networks (Szegedy et al., 2016), but McKinley et al. (2019) have shown label smoothing to improve classifier calibration and to also be directly learnable by the model. Their loss function for a binary classifier is defined as:

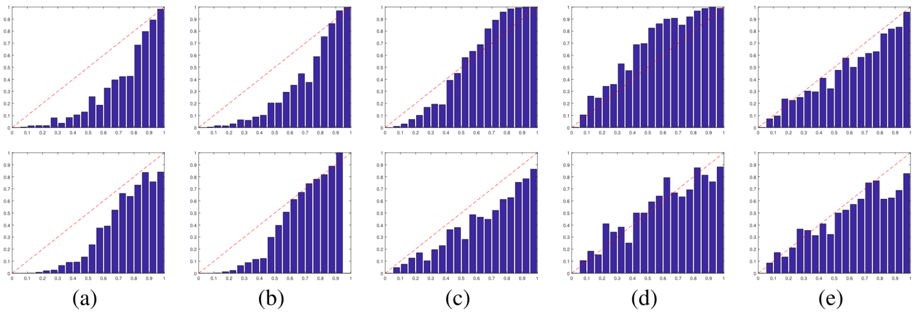


Figure 3.1: Image taken from Neumann et al. (2018): Reliability diagrams on two different data sets (rows) and with different output calibration methods: (a) softmax, (b) softmax with linear scaling, (c) softmax with temperature scaling, (d) softmax with heteroscedastic logit smoothing, (e) softmax with heteroscedastic logit smoothing and linear scaling

$$L(p, (1-y)u + y(1-u)) + L(u, z) \qquad \text{with } z = \begin{cases} 1 & \text{if } p > 0.5 \text{ and } x \neq 1; \\ 0 & \text{otherwise.} \end{cases} \tag{3.14}$$

It includes a learnt uncertainty score $u \in (0, 0.5)$ that smoothes the ground truth label, thereby acting as loss attenuator. An additional loss term applies the same loss, e.g. cross-entropy, on the predicted uncertainty and the indicator for disagreement between model prediction and label, which penalises high uncertainty for correctly predicted samples. Standard binary cross-entropy loss is recovered when zero uncertainty is predicted and the predicted label agrees with the ground truth.

The authors demonstrate that the confidence score captures label noise, i. e. aleatoric uncertainty, and leads to effective loss attenuation in the medical imaging domain.

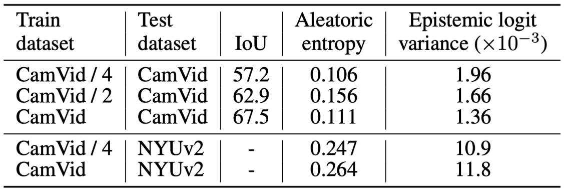
3.6 Decomposition of Uncertainties

As described in Section 1.1, decomposing a model's predictive uncertainty into aleatoric and epistemic uncertainty leads to more interpretable uncertainty estimates and benefits other applications, such as active learning or loss attenuation. Therefore, this section outlines two prominent approaches to uncertainty decomposition.

3.6.1 Kendall and Gal

Kendall and Gal (2017) decompose a model's prediction uncertainty into aleatoric and epistemic components by combining a BNN with additional heteroscedastic uncertainty neurons. The loss-attenuating neurons were already presented in Section 3.5.4. The authors build on the fact that aleatoric uncertainty is by definition irreducible, so models need not try to improve their performance in this regard. As a result, aleatoric uncertainty lends itself naturally to use in loss attenuation, which should result in the heteroscedastic neurons capturing this type of uncertainty. Meanwhile the remaining, i. e. epistemic, uncertainty is reflected in the variability of the BNN's MC samples. Their setup thus jointly models both uncertainty types by explicitly using two uncertainty quantification mechanisms inside a single model.

The authors found that modelling epistemic, aleatoric, as well as both uncertainties together improves segmentation performance over their baseline, a vanilla DenseNet (Huang et al., 2017). Regarding the quality of the uncertainty decomposition, the authors have shown that for varying training set sizes and inference on OOD data the decomposed uncertainties quantitatively behave in line with their definition, i. e. epistemic uncertainty rises for OOD samples and smaller training sets, while aleatoric uncertainty remains constant. Qualitatively, Kendall and Gal found aleatoric uncertainty to highlight object borders as well as objects far away from the camera, while epistemic uncertainty faintly highlights object borders and substantially shows up in visually challenging pixels as well as instances of rare classes. Their quantitative results regarding the quality of uncertainties are included for reference in Figure 3.2.



Train dataset	Test dataset	IoU	Aleatoric entropy	Epistemic logit variance ($\times 10^{-3}$)
CamVid / 4	CamVid	57.2	0.106	1.96
CamVid / 2	CamVid	62.9	0.156	1.66
CamVid	CamVid	67.5	0.111	1.36
CamVid / 4	NYUv2	-	0.247	10.9
CamVid	NYUv2	-	0.264	11.8

Figure 3.2: Kendall and Gal's results for semantic segmentation

3.6.2 Decomposition of a BNN's Predictive Distribution

Directly decomposing the predictive distribution of a BNN has the theoretical advantage that the relationship between aleatoric and epistemic uncertainty can be modelled. Also, from an implementation standpoint, solely building a BNN without additional output neurons is straightforward.

Variance Decomposition Kwon et al. (2018) derive both aleatoric and epistemic uncertainty estimates from the variance of the BNN's predictive distribution via:

$$Var_{p(y|x)}(y) = \mathbb{E}_{p(y|x)}\{y^{\otimes 2}\} - \mathbb{E}_{p(y|x)}\{y\}^{\otimes 2} \tag{3.15}$$

with $y^{\otimes 2} = yy^T$. The derivation makes use of a variant of the law of total variance. The aleatoric term computes the variance of the Bernoulli random variable, aka. the individual predictions, for each class. This means that aleatoric uncertainty is measured as the average variance of the individual softmax predictions while epistemic uncertainty is reflected in the variability of the network weights.

The authors directly compare their approach to that of Kendall and Gal, employing the activation of the heteroscedastic neuron as aleatoric estimate, in the context of an ischemic stroke lesion segmentation task. The resulting uncertainty maps show that the heteroscedastic uncertainty neurons produce next to none aleatoric uncertainty and diffuse epistemic uncertainty only appearing in the background of the image. Meanwhile, Kwon et al.'s aleatoric uncertainty estimates highlight class borders and incorrectly segmented regions, which largely conforms to human intuition, and their epistemic uncertainty estimates partially shade incorrectly segmented regions as well as the borders of segmented objects. Furthermore, Kwon et al. (2018) conducted a comparison between two different data sets with different training set sizes and stroke lesion proportions. Both epistemic and aleatoric uncertainty were significantly higher for the smaller data set, but the authors concluded that the smaller data set containing a higher percentage of lesion voxels was inherently ”noisier” and after deconfounding the aleatoric uncertainty, the conditional expectations of epistemic uncertainty were still slightly higher on the smaller data set. In later work, Kwon et al. (2020) perform an additional experiment comparing models trained on different numbers of training samples coming from the same data set. The authors find increased epistemic uncertainty for smaller training sets, while aleatoric uncertainty stays constant.

Mutual Information Decomposition The MI measure of a BNN's predictive distribution has already been described in Section 3.4 and is repeated here for the purpose of readability:

$$\mathrm{MI}[y_i, W \mid x_i, \mathcal{D}] \approx \mathrm{H}[y_i \mid x_i, \mathcal{D}] - \mathbb{E}[\mathrm{H}[y_i \mid x_i, W]] \,. \tag{3.16}$$

Since it computes the reduction in uncertainty for the network weights W given a sample x_i and the ground truth class label y_i, it can be interpreted as computing the model's epistemic uncertainty (Mobiny et al., 2021; Depeweg et al., 2018; Houlsby et al., 2011). Epistemic uncertainty potentially benefits from the reveal of the ground truth label, while aleatoric uncertainty, by definition, does not.

The minuend in Equation 3.16 constitutes the predictive entropy, as presented already in Section 3.4, which computes the overall predictive uncertainty of a model (Depeweg et al., 2018). The subtrahend eliminates the epistemic component, i. e. the weight uncertainty, from the predictive entropy by computing the expected value of the predictive entropy as conditioned on W. In practice, this translates to computing the average entropy of the individual predictions, which corresponds to measuring aleatoric uncertainty. The epistemic uncertainty estimate is then computed by subtracting the aleatoric uncertainty from the predictive entropy. Mutual information thus constitutes a valid decomposition method.

In Mobiny et al. (2021), qualitative observations of the resulting uncertainty maps of a BNN approximated via MC DropConnect for three different semantic segmentation tasks show clear correlations between epistemic uncertainty and class boundaries, class label frequency, and visual ambiguity of objects. To the best of the author's knowledge, no direct comparison between this decomposition method and Kendall and Gal's combined setup has been reported in the literature.

Material and Methods

This thesis aims to comprehensively compare two prominent uncertainty quantification approaches for deep learning models, with a focus on the quality of the decomposition of the predictive uncertainty into aleatoric and epistemic components. The comparison is set up in the context of binary liver segmentation on medical CT scans contained in the LiTS data set (Bilic et al., 2019).

This chapter first briefly introduces the hard- and software used for all experiments. Afterwards, the data set and preprocessing steps are described. The following section motivates the choices made throughout this work by first exposing the parallels between two loss attenuating methods and then describing the concrete setup of the models and experiments, including the choice of model, hyperparameters, and data sets. This also includes a description of the implementation of the two loss attenuation techniques. Afterwards, the choice of uncertainty metrics is explained. The last section describes the evaluation approaches and metrics that are used to answer this work's research question.

4.1 Hard- and Software

The following paragraphs briefly outline the frameworks used in this work, as well as the computational resources with which all neural network models are trained and tested.

MeVisLab This work loads, preprocesses, and visualises the data set with the MeVisLab framework, which provides user interfaces, programming environments, libraries and modules for medical image processing. MeVisLab modules are responsible for loading data, applying filters or transformations, or computing statistics about an incoming image stream. They can be arranged into a hierarchical data flow network by visual programming via the object-oriented GUI. MeVisLab supports scripting via Python and custom modules can be added in Python or C++. Because of its focus on medical imaging, MeVisLab supports common medical image formats, such as NIfTI and Dicom. This work uses MeVisLab version 3.5legacy.

RedLeaf For training and testing neural network models, this work employs the REmote Deep LEArning Framework (RedLeaf). RedLeaf is internally developed by Fraunhofer MEVIS. It is used in this work because of its tight integration with MeVisLab. The RedLeaf Python framework enables the implementation for training and testing neural networks using popular libraries, such as Keras, Tensorflow, and Pytorch. RedLeaf is also integrated into MeVisLab modules, which perform data preparation and patch extraction, before feeding the resulting streams of image data into NN models. RedLeaf offers predefined model architectures, such as u-net and au-net, the latter of which is used in this work.

Keras The deep learning framework Keras is an open-source Python library (Chollet et al., 2015). It provides a high-level programming interface for the machine learning library Tensorflow (Abadi et al., 2015), which is based on the concept of data flow graphs describing series of mathematical operations over multidimensional data arrays, or tensors. Keras aims to provide a simplistic, intuitive, and modular API for Tensorflow. While a wide range of commonly used layers, optimisers, activation and loss functions is already built-in, keras also facilitates the creation of custom variants, e.g. via layer subclassing.

GPU Cluster Training and inference of all neural networks in this work is performed on a cluster provided by Fraunhofer MEVIS, making heavy use of GPU compute accelerators. The cluster consists of eight nodes. Two nodes are equipped with four NVIDIA Geforce GTX 1080 Ti, Intel Xeon CPU E5-2620 v4, and 188.79 GiB RAM, each. The other six nodes each feature four NVIDIA Geforce RTX 2080 Ti, Intel Xeon Silver 4214 CPU, and 187.57 GiB RAM.

4.2 Data

The following sections describe the data that is used for training, validation, and testing in the experiments of this work, including the preprocessing steps used to prepare the data.

4.2.1 Liver Tumor Segmentation Data Set

All the experiments conducted in this work use the Liver Tumor Segmentation data set (LiTS; Bilic et al., 2019). The LiTS data was made available for the Liver Tumor Segmentation benchmark/challenge jointly organised by the IEEE International Symposium on Biomedical Imaging (ISBI 2017) and the 20th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2017). LiTS is a collection of 201 CT scans of patients' abdominal regions and includes 131 annotated images and 70 test images. The expert annotations include masks for liver tissue as well as primary and metastatic liver tumors. The scans were collected from seven research institutions and hospitals, and thus vary widely in terms of amount of lesions, tumor contrast levels, and tissue size abnormalities.

Of the 131 CT scans 65 were classified as ”poorly annotated” by an independent expert. These include more imprecise borders and halfway segmented vena cavas, which are likely due to automatic post-processing steps that interpolate between two liver positions. Because these cases contain a variety of poor-quality features, they are omitted from the training and test sets and only partially included in the validation set. In total 32 CT volumes are used as the default training set, henceforth called LiTS-full, another 33 cases are used as a test set, called LiTS-test from now on, and an additional 7 cases serve as validation set.

The training, validation, and test sets contain 320.844.582 voxels in total, with a mean ratio of 29% liver voxels and 0.17% tumor voxels. The mean image volume is 74.823.829 mm^3 with a minimum of 19.699.218 mm^3 and a maximum of 137.333.984 mm^3.

4.2.2 Data Preprocessing

The diversity of the LiTS data sources is reflected in the CT image data, with different volume sizes, voxel sizes, and different numbers of slices per case. In order to standardise the data for further experiments, all CT scans and liver masks are resampled to a uniform voxel size of $2.5 \times 2.5 \times 2.5$ mm^3 via a Lanczos filter with kernel size 3 for the images and trilinear filtering for the masks. While this voxel size is relatively coarse, the resolution of the resulting scans is found to be sufficiently high for the task of uncertainty estimation and decomposition, simultaneously allowing for a larger receptive field (in terms of mm) and heavily reducing the training time required to reach convergence.

Moreover, the ground truth annotations of tumors are cast to liver masks in order to conform to the binary segmentation task. All CT volumes are resampled to the transversal view direction. The preprocessing pipeline as implemented in MeVisLab is shown in Appendix A.

4.3 Approach

The binary segmentation of livers is a well explored problem and existing results achieve very good segmentation performance. For example, one instance from 2017 (Bilic et al., 2019) reported a Dice score of 0.96 on this task. Therefore, this work explicitly does not aim to find an even better architecture that increases state of the art performance or even calibration scores. Instead, this work focuses on the quality of different uncertainty decomposition methods. The setup described in the following is thus intended to yield a reasonably performing model that can be easily extended for uncertainty quantification.

The following section then explores the relationship between heteroscedastic uncertainty neurons (HUNs) and heteroscedastic logit smoothing (HLS) and provides insights on the choice of the heteroscedastic logit smoothing neuron (HLSN) as the basis for the comparison of vanilla BNNs with those augmented with an uncertainty estimating neuron. Afterwards, the architectural choices and hyperparameters of the models, as well as the details of the implementation of the different uncertainty layers are described.

4.3.1 Parallels between HUN and HLS

This subsection compares the mechanisms underlying the heteroscedastic uncertainty neuron by Kendall and Gal (2017) with those of the heteroscedastic logit smoothing approach by Neumann et al. (2018).

Heteroscedastic Uncertainty Neuron

Sampling noise with a learnt variance over the logits results in loss attenuation, because the softmax operation squashes logits, that were sampled with a high variance, more together in the extreme values. In the case of a wrong prediction, the loss is attenuated by the resulting predictive distribution which more closely follows a uniform distribution. This behaviour

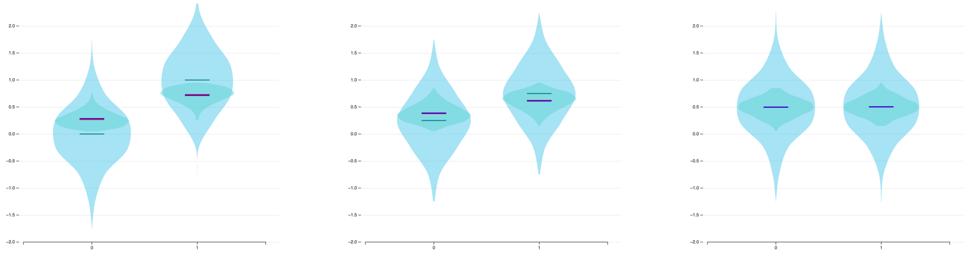


Figure 4.1: Numerical simulations of sampling logits through the softmax as proposed by Kendall and Gal (2017), σ is set to 0.5 for both classes, incoming logits are $[0, 1]$, $[0.25, 0.75]$, and $[0.5, 0.5]$, the number of samples is 1000.

can be observed in the numerical simulations shown in Figure 4.1. The teal-coloured lines indicate the two original logits, the larger violins represent the distribution of the sampled logits, and the smaller violins show the distribution of the sampled logits after being passed through the softmax operation. The red lines indicate the softmax scores of the two original logits and the blue lines finally indicate the mean of the sampled softmax scores, i. e. the model's final output. As can be seen in the graphic, the means of the sampled softmax scores (blue) lie closer towards 0.5 than the original softmax scores (red) would have. This phenomenon is explained by the non-linear transformation of the softmax operation. The previously symmetric Gaussian sampling distribution over the logits becomes an asymmetric distribution over the softmax scores of those same sampled logits, with higher density towards the respective extreme values 0 and 1. This asymmetric density illustrates the squashing behaviour of the softmax operation when applied on normally distributed logits. The whole process could also be described as ”smoothing” the softmax scores.

Heteroscedastic Logit Smoothing

This smoothing of the logits can also be explicitly and directly achieved by multiplying the original logits with a learnt confidence score in $[0, 1]$, as has been shown by Neumann et al. (2018). Although originally termed relaxed softmax, this approach is henceforth denoted as heteroscedastic logit smoothing (HLS), to capture its close relation to the mechanism behind the heteroscedastic uncertainty neuron (HUN).

While both approaches aim at pushing the predictive distribution towards a uniform distribution for difficult samples, Kendall and Gal's method does so via one heteroscedastic uncertainty neuron per class, whereas HLS naturally only works with a single smoothing factor.

Due to the overall similarity of the underlying behaviour of these loss attenuation mechanism, this work tries to employ HLS inside a BNN, similar to the setup of Kendall and Gal (2017). This work assumes that the combination achieves comparable uncertainty to the original approach including the HUN.

4.3.2 Network Architecture and Hyperparameters

Anisotropic U-net All models in this work use as their base architecture a 5-level 3D anisotropic u-net (au-net; Chlebus et al. (2022)), which constitutes a modification of the u-net (Ronneberger et al., 2015). The u-net was introduced as a fully-convolutional model architecture for biomedical image segmentation. It employs an encoder-decoder architecture, which gives the model a characteristic U-shape and hence its name. The near symmetry of the architecture is also due to the equal number of feature channels in both the encoder and decoder paths, allowing the model's higher resolution layers to receive more contextual information from lower ones. Instead of more traditional upsampling techniques, such as nearest neighbour interpolation or max-unpooling, the u-net makes use of learnable transposed 3D convolutions in its decoder. The au-net introduces anisotropy to the processing of the x, y, and z spatial dimensions. Instead of 3D convolutions operating on all three image axes, the upper two resolution levels of the au-net employ 3D convolutions working along the x and y dimensions only, and the other levels employ depthwise separable 3D convolutions, which significantly decreases the number of weights to be learnt. Moreover, the convolution layers consist of two convolutions and a ReLU activation function followed by a 2×2 max-pooling operator in the encoder, and a corresponding upsampling convolution in the decoder. The architecture of the au-net used in the experiments in this work is shown in Figure 4.2.

Moreover, this work employs MC dropout for the variational approximation of all BNNs due to its easy implementation and robust results, as outlined in Section 3.4.

Patch-based Image Processing Training and inference in this work are done patch-wise, meaning that image sub-portions of a specified size and

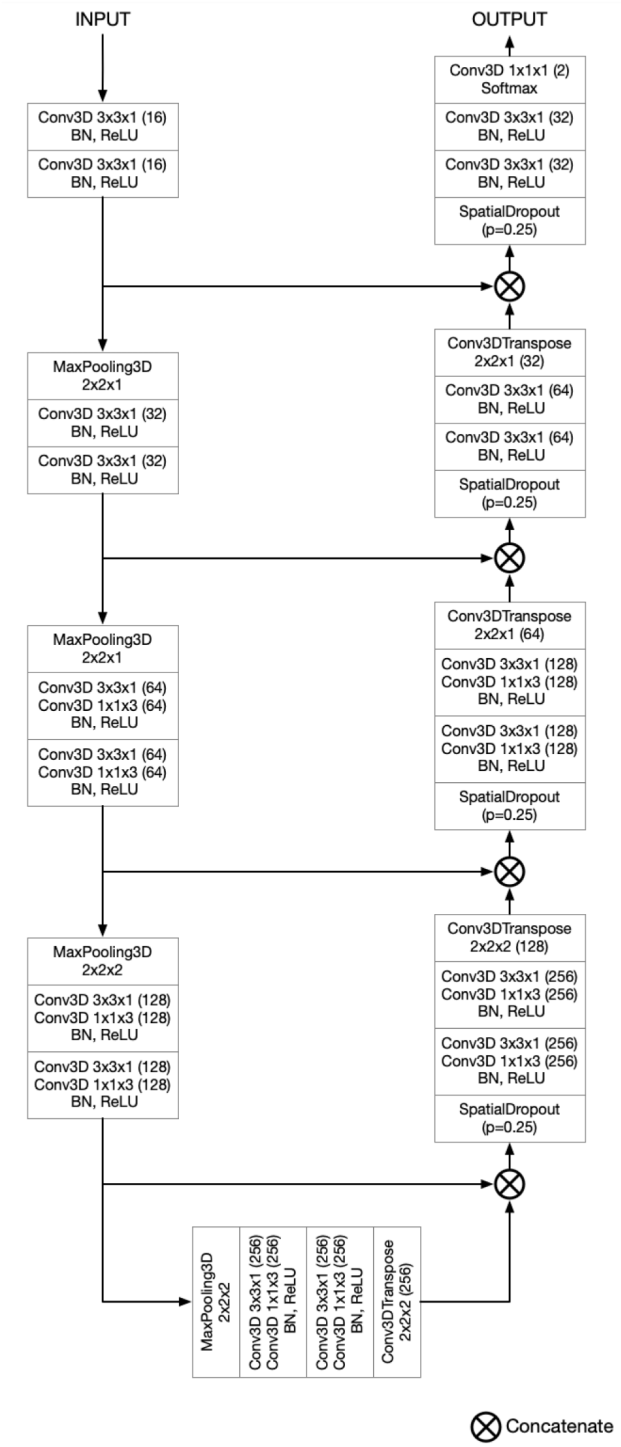


Figure 4.2: Au-net architecture with five resolution levels, image taken from Chlebus et al. (2022)

padding, instead of the whole image, are fed into the model. The use of patches is popular in medical image segmentation, where 3D images with extremely high resolution are common, and operations on whole images are prohibitively expensive. The patch size for the experiments is set to $52 \times 52 \times 52$ voxels with a padding of $92 \times 92 \times 20$ voxels, reflecting voxels at the patch border whenever the input image extent is exceeded. This size constitutes a good trade-off between memory usage and the model's receptive field. The receptive field allows the model to still learn global spatial properties from anatomic structures, such as the general position of the liver, thereby avoiding most false positive predictions. The use of patches also enables patch stratification, a technique to improve model training on imbalanced data sets. The LiTS-full training data, for example, is imbalanced in that 39% of the voxels belong to the foreground class. Such a separation of the training data into a common majority class and less common minority classes can easily result in biased models whose predictive accuracy is lacking for the minority class, i. e. the liver in this case. Patch stratification addresses this by oversampling the minority class, so that the classes are more evenly distributed in the training set. Balancing out the foreground-background voxel ratio of the original/actual distribution allows the model to learn from more liver voxels per batch than it would otherwise and has been shown to improve performance, especially on the more infrequent classes (Johnson and Khoshgoftaar, 2019). For the experiments in this work, a stratification ratio of 80% patches containing at least one foreground voxel and 20% containing only background voxels is chosen. Patches exclusively containing liver voxels are not available, since no liver mask completely fills a patch. The data preparation in this work was implemented using MeVisLab, where patch stratification can be straightforwardly included in the training via the StratifiedPatchSampler module. An example of this can be seen in the data-serving MeVisLab network in Appendix A.

Training Configuration All models in this work are trained using the adam optimiser and a max patience-stopping criterion of 40 epochs, meaning that the training is finished once the Jaccard performance measure of the model on the validation set has not improved for 40 epochs. The batch size is set to 2, in order to allow for rapid training convergence without exhausting the available memory. Training-time data augmentation, in which additional training data is generated from the existing training data, is commonly employed to improve a model's accuracy and might even help decrease its epistemic uncertainty. However, as this work does not aim to produce the most well-calibrated liver segmentation model, but rather to evaluate the uncertainty quantification and decomposition techniques as such, no data augmentation was employed.

Model Hyperparameters For all BNNs trained in this work the dropout rate is set to 0.2 as in Kendall and Gal (2017). In preliminary experiments this value appeared to be a good compromise, the visibility of patch borders in the inferred images is minimal, while the model still converges robustly towards learning reasonable uncertainty in the BNN-HLS models. It is important to note, however, that many works employ 0.25 or 0.5, instead (DeVries and Taylor, 2018b; Gal and Ghahramani, 2016; Mobiny et al., 2021; Kwon et al., 2020).

The number of MC dropout samples is set to 20. In general, the number of stochastic forward passes used in the literature is diverse, ranging from 5 samples (Kwon et al., 2020) over 10 (Nair et al., 2020) and 20 (Chlebus et al., 2022) to 50 (Kendall and Gal, 2017). Mobiny et al. (2021) reported a prediction error one standard deviation from the best performance at 54 samples, but showed significant test error improvement after around 15 to 20 samples. Chlebus et al. (2022) also found 20 samples to be a "reasonable tradeoff between uncertainty resolution and computation speed".

Different dropout positions have been used in the literature. Kendall and Gal (2017) employ dropout throughout the whole net and Gal and Ghahramani (2016) showed the mathematical equivalence between a neural network with dropout in every layer and an approximation of a deep Gaussian process. However, Kendall et al. (2017) tested different dropout layer positions with regard to model performance and found that dropout throughout the whole model results in too strong regularisation, therefore leading to longer training times and decreased test performance. Moreover, their results demonstrate that performance is not improved when dropping out the first layers of a network, which is where basic image features are extracted (Zeiler and Fergus, 2014). In preliminary experiments dropout throughout the whole model, at the set rate of 0.2, resulted in highly visible patch borders of inferred images, drastically impairing their qualitative evaluation, while decoder-only dropout produced interpretable uncertainty estimates and much less visible patch borders. Dropout is therefore only applied in the decoder of the au-net.

Losses The choice of loss function is an important hyperparameter that has been shown to strongly influence a classifier's calibration. Cross-entropy loss, being a proper scoring rule, promotes calibrated predictions, whereas the soft Dice loss improves segmentation performance at the expense of model calibration (Guo et al., 2017). Mehrtash et al. (2020) also have shown that Dice loss leads to less well-calibrated BNNs than the cross-entropy loss. Bertels et al. (2021) demonstrated via numerical simulation that for true foreground probabilities with inherent aleatoric uncertainty, the local minimum of the expected value of the Dice loss lies at either 0 or 1, instead of the true foreground probability. Thus, the Dice loss cannot be interpreted as a proper scoring rule, and tends to promote under- and overconfident predictions.

Due to this both Kwon et al. (2018), Kendall and Gal (2017), Neumann et al. (2018), and Mobiny et al. (2021) employ cross-entropy loss in their experiments, since their research focus lies on producing calibrated uncertainty estimates. Keeping close to the original approaches, this work also uses the cross-entropy loss as the main objective function. Despite the mentioned problems, Dice loss is a popular choice for improving a model's overall segmentation quality in practical image segmentation settings. This work, therefore, also briefly examines whether the miscalibration induced by training with soft Dice loss similarly affects models that employ additional heteroscedastic neurons for calibration. Further, the quality of the uncertainty decomposition for models that are trained with the soft Dice loss is examined. This work puts forward the hypothesis that the aleatoric uncertainty estimates are heavily decreased, since the mutual information computation of aleatoric uncertainty uses the individual softmax prediction distributions, which have been shown to be overconfident when Dice loss is employed.

4.3.3 Employed Model Architectures

Altogether the following CNN architectures are trained and evaluated:

- A pure-NN model is trained with the au-net architecture, without dropout or additional heteroscedastic units.
- The pure-NN model is extended with two heteroscedastic uncertainty neurons (HUNs), no dropout is applied.
- The pure-NN model is extended with a heteroscedastic logit smoothing neuron (HLSN), no dropout is applied.

- A BNN is approximated via MC dropout, henceforth called BNN-MI.
- A BNN is approximated via MC dropout and extended with an HLSN, henceforth called BNN-HLS.

4.3.4 Heteroscedastic Uncertainty Neuron

σ activation As the heteroscedastic uncertainty neuron defines the variance (σ) of the Gaussian noise used to sample and by extension smooth the logits, the range of its output needs to match the domain of the variance in the Gaussian noise function. To achieve this, the HUNs' behaviour was investigated for implementations with a sigmoid activation function and a softplus activation function. Activations in the softplus case lie in the range of $[0, +\infty]$, while activations in the sigmoid case normalise to the range of $[0, 1]$, which seems appealing for subsequent activation map analyses of the uncertainty output. The convergence with a sigmoid towards an uncertainty-capturing unit turned out to be more difficult, since the constrained variance only provides the desired smoothing effect for smaller-valued logits. The range of the logit outputs, however, varies between training runs due to random weight initialisations and random mini-batches. Uncertainty estimates that passed through the softplus activation function are clearer and converge more robustly towards their intended meaning, i. e. capturing label noise. In a sense the softplus' σ should automatically correspond to the magnitude of their respective logits, since both are derived from the same layer, i.e. have the same data source.

Sampling Procedure Different procedures for sampling the noise also affect the performance of the HUN and were, therefore, also investigated. The original method employed by Kendall and Gal (2017) samples a fixed number of times, then computes the softmax over all sampled logits, and finally averages the resulting scores. The number of samples constitutes a trade-off between the computing time on the one hand and the sampling error on the other hand, with less samples leading to less robust loss attenuation. Kendall and Gal (2017) do not report the number of samples drawn for their Gaussian noise, the reproduction in DeVries and Taylor (2018b) uses 100 samples. In preliminary experiments 25 samples were found to be sufficient for inducing the loss attenuating behaviour, and more samples did not have a positive impact on the HUNs' convergence. In contrast, the strategy employed by Kwon et al. (2020) is to sample only once, i. e. $\mathbf{p} \sim \mathcal{N}(\mathbf{z}, \sigma)$. This work reproduced the same convergence problems encountered in their original study, where next to no uncertainty is captured by the HUN.

All subsequent experiments use the original method used by Kendall and Gal (2017), which yields good uncertainty estimates when combined with a vanilla NN.

Convergence To achieve robust capture of uncertainty in the HUN activation, the training of a HUN-model needs some guidance, similar to the learnt confidence estimate or abstention networks, presented in Section 3.5.2 and 3.5.3, respectively. Initialising the σ weights following a Gaussian distribution, i. e. $\mathbf{w} \sim (-1, 0.5)$, was found to enable reasonable convergence towards uncertainty in preliminary experiments. In combination with the previous ReLU activation, this weight initialisation translates to the network having very low uncertainty at the beginning of training, with uncertainty increasing as it encounters more samples that it has difficulties to learn. Without this soft training guidance, the HUN activations were found to frequently converge towards mimicking the logit values, which renders the HUN activation useless as an uncertainty estimate and slightly decreases model performance. Adding a HUN to a BNN, despite different weight initialisations and dropout rates, resulted in even more difficult convergence, with the HUN-activation frequently converging towards uniform zero.

4.3.5 Heteroscedastic Logit Smoothing

While its straightforward implementation is one of the attractive properties of HLS, not requiring any custom layers or changes to the loss function, there are still some practical considerations to be made to achieve reliable uncertainty estimates.

α **activation** The heteroscedastic logit smoothing model employs a sigmoid activation for α, so that the logits are scaled with a value in $[0, 1]$ prior to the application of the softmax. Although not explicitly specified by the original authors, constraining α to not take on values above 1 seems a natural choice. While temperature scaling is theoretically able to scale the logits both down and up, this form of learnt temperature scaling is expected to perform loss attenuation. Attenuating the loss of difficult samples by smoothing their predictive distribution whilst predicting a smoothing factor of 1 for the rest corresponds exactly to this behaviour. Furthermore, constraining the confidence values in this way also constrains the resulting uncertainty maps, thereby simplifying subsequent analyses.

Convergence In preliminary experiments the model's convergence towards the desired loss attenuation property was evaluated by qualitatively inspecting the activation of its HLSN. This convergence is not expected to be heavily reflected in the final segmentation performance or reliability, as the models' underlying architecture is quite complex, the input scans are resampled to a relatively large voxel size, and this work considers the binary segmentation problem of relatively large structures.

When trained naively, the models employing additional HLSNs were found to frequently converge towards very noisy HLSN-activations, in which not only the class boundaries, but also other structures as well as parts of the background are highlighted. An example of such an activation map is shown in Figure 4.3. As expected, the noisily applied smoothing factor does not strongly harm the final predictions (preliminary experiments revealed a segmentation decrease of around -0.015 for Dice and an increase in NLL of around 0.2% for unconverged models). However, the extent of the uncertain regions which do not correspond to human intuition about aleatoric uncertainty vary from model to model, which impedes subsequent comparisons between different training and inference setups.

Therefore, this work implements the HLSN with weights initialised following a Gaussian distribution, i. e. $\mathbf{w} \sim (1, 0.6)$. In combination with the previous ReLU activation, which leads to incoming values in the range of $[0, +\infty]$, this weight initialisation lets the model enter its training with higher confidence than when using keras' default Glorot initialiser, which samples from a uniform distribution $[-\text{limit}, \text{limit}]$ where $\text{limit} = \sqrt{6/(\text{fan\_in} + \text{fan\_out})}$. Over training, the model might then lower the confidence upon encountering difficult samples. This setup was found to more robustly lead to the aleatoric neuron capturing uncertainty. In particular, the amount of noise in the HLSN-activation is notably decreased, as shown on the right of Figure 4.3. This dynamic is reminiscent of the need for custom training regimes employed for several other loss attenuating models, as described in Sections 3.5.3 and 3.5.2.

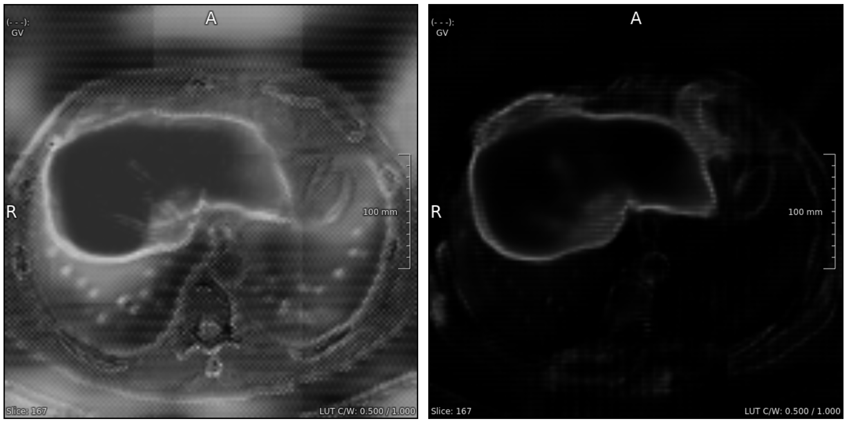


Figure 4.3: Activation maps for the HLSN in two different BNN-HLS models, illustrating convergence behaviour

4.3.6 Uncertainty Metrics

Interpretation of Uncertainty Metrics If one wants to separate aleatoric from epistemic uncertainty given a BNN's predictive distribution, it is helpful to consider how these uncertainty types are reflected in the model's output. To ease understanding, let us take a step back and make the following observations; If epistemic uncertainty is interpreted as model or weight uncertainty, then different MC forward passes, which sample from the model's posterior weight distribution, should result in diverse predictions. According to this interpretation, a model predicting $[[0, 1], [1, 0]]$ with two MC forward passes for one sample exhibits maximal epistemic uncertainty. Aleatoric uncertainty, on the other hand, is not model-specific but rather depends on the randomness of the underlying input data-generating process. Therefore, it should not be influenced by weight uncertainty. In order to reliably cope with label noise, a model should produce predictions whose class probabilities represent the ground truth distribution's stochasticity. Thus, the above model would exhibit zero aleatoric uncertainty, while a model predicting $[[0.5, 0.5], [0.5, 0.5]]$ would assign the maximum aleatoric uncertainty to the given sample.

With these observations in mind, the capability of predictive entropy, predictive variance, and MI for computing specific uncertainty types should be briefly reconsidered. More specifically, predictive entropy computes the entropy of the averaged predictions, thereby capturing both epistemic and aleatoric uncertainty, as interpreted above. Predictive variance, on the other hand, computes the variance between the samples, thus would assign zero uncertainty to the maximally aleatoric uncertain example above and maximal uncertainty to the epistemic example. MI essentially subtracts the aleatoric uncertainty, as computed by the averaged entropy of single samples, from the overall uncertainty, as measured by predictive entropy. Thus, MI computes epistemic uncertainty as well. This aligns well with its probabilistic interpretation of the question "how much can we learn about the model weights if we knew the target of a given sample?".

Aleatoric Uncertainty Metric for Loss-attenuating Neurons Intuitively, the activation of loss-attenuating neurons constitutes an (aleatoric) uncertainty estimate, since the model learns to produce high activation values when a difficult sample is encountered during training in order to decrease, i. e. attenuate, the loss. This general mechanism behind loss attenuation is presented in Section 3.5. Notably, all works that reproduce Kendall and Gal's HUN employ this straightforward metric (Nair et al., 2020; DeVries and Taylor, 2018b; Kwon et al., 2020).

However, the raw neuron activation does not correspond to any well-defined metric, and it does not constitute a probability since its values lie in the range of $[0, +\infty]$. This makes it harder to interpret the resulting aleatoric uncertainty estimates and complicates usage in subsequent downstream applications. Moreover, the degree of loss attenuation that is induced by the activation of the HUN and HLSNs depends both on the activation itself as well as the logits' magnitudes. In fact, the logits themselves inherently contain aleatoric uncertainty, which can be computed via the argmax, variance, or entropy of the final softmax distribution. This dependency and the separate amount of uncertainty are ignored when the uncertainty neuron's activation is interpreted as aleatoric uncertainty directly. In general, the HUN and HLS neurons constitute an additional calibration technique offering the model another "lever" to calibrate its predictions. Their activation directly modifies the softmax output distribution by smoothing the incoming logit values at both training and test time, as shown in Section 4.3.1. Therefore, a comprehensive aleatoric uncertainty measure can and should make use of one of the several well-defined metrics on the output distribution, such as variance or entropy. In fact, Kendall and Gal themselves employ "aleatoric entropy" in their work.

Interestingly, this observation produces a strong argument in favour of the approaches by Kendall and Gal (2017) and Neumann et al. (2018). It also motivates their combination with a BNN, because direct decomposition metrics applied to a BNN's predictive distribution, such as MI or variance decomposition, described in Sections 3.6.2 and 3.6.2, lend themselves naturally to this setup. Other approaches that employ some form of loss-attenuation, which are listed in Section 3.5, lack this straightforward aleatoric uncertainty measure, because their uncertainty values do not directly influence, i. e. calibrate, the model's predictive distribution at test time. They, instead, count only towards the loss function itself, as does a separate term involving the model's softmax scores. So, while the output of other uncertainty neurons does capture uncertainty, the softmax distribution remains an independent indicator for potentially remaining predictive uncertainty. A comprehensive aleatoric uncertainty measure would have to take into account the uncertainty encoded in both quantities, but evaluating the overall uncertainty over these two sources is not as simple as in the case of HUN and HLS.

Having made these observations, let us now revisit a critique voiced by Kwon et al. (2018) about the aleatoric uncertainty estimate in Kendall and Gal (2017). Kwon et al. found the aleatoric uncertainty estimate not to be a true random variable, because the predictive distribution's variance (as captured by the HUN's output) is not modelled as a function of its mean (the model output vector). This critique, however, rests on the assumption that the authors proposed to use the HUN's output as aleatoric uncertainty estimate, while Kendall and Gal actually used the entropy of the softmax distributions averaged over the MC samples. The aleatoric uncertainty in Kendall and Gal's approach can thus be calculated in a similar fashion as Kwon et al.'s aleatoric estimate.

Due to these considerations, this work reports both the output of the HLSN and the aleatoric component of the mutual information metric applied on the BNN's predictive distribution. Theoretically, both variance decomposition and mutual information metric, as described in Section 3.6.2, are valid choices for the direct uncertainty decomposition of the predictive distribution of BNN-HLS models. Because the two metrics are very similar and since the latter has already been successfully used by numerous authors (Mobiny et al., 2021; Depeweg et al., 2018; Nair et al., 2020), this work uses the mutual information decomposition, henceforth called MI, for all experiments. This also streamlines the main comparison in this work, minimising the number of differing variables between the decomposed uncertainties derived from

BNN-MI and BNN-HLS. Reporting both HLSN-activation and the aleatoric term of the MI decomposition enables a direct comparison between these two estimates. This work will compare the resulting uncertainty maps, as well as investigate how their behaviour for different training and inference data corresponds to the human intuition for aleatoric uncertainty. The specific experimental setups are described in detail in the following Section 4.4. The local MeVisLab macro for computing the MI decomposition from the predictive distribution of a BNN with and without additional heteroscedastic neuron is included in Appendix B.

4.4 Evaluation

This section describes the evaluation procedures and metrics that are to answer the research question. After outlining the overall segmentation and calibration metrics, some brief considerations about the expected separability into aleatoric and epistemic uncertainty are provided, followed by an explication of the experiments used to evaluate the quality of decomposed uncertainties. Lastly, the usage of statistical significance tests is described.

Performance and Calibration Employing a held-out test set, the segmentation performance is measured via Dice index and IoU for all experiment settings. Moreover, calibration is illustrated in reliability diagrams and quantitatively assessed via NLL, ECE, and MCE.

Qualitative Evaluation In order to qualitatively evaluate the decomposed uncertainties, the resulting uncertainty maps and their correspondence to the intuitive interpretation of aleatoric and epistemic uncertainty are investigated, as described in Section 2.5.2. Since LiTS-full has high-quality annotations, aleatoric uncertainty is expected to appear mainly at class boundaries. Moreover, due to the absence of instances of rare classes, given the binary liver segmentation task, epistemic uncertainty estimates are expected to have low magnitude.

Expected Separability of Uncertainties Let us briefly consider the degree of separability into aleatoric and epistemic uncertainty that is expected to be achievable in the experiments of this work.

Empirically, several authors have shown decomposed uncertainty values to be slightly "mixed" or "intertwined". The epistemic uncertainty estimates obtained via MI decomposition in Mobiny et al. (2021) occur in class boundaries, instances of infrequent classes, and visually difficult or ambiguous objects. Nair et al. (2020) find predictive entropy as well as MI to highlight class boundaries. While predictive entropy is expected to constitute the model's overall uncertainty, mutual information, as argued in Section 3.6.2, reflects epistemic uncertainty. Finally, both aleatoric and epistemic uncertainty estimates derived via variance decomposition highlight incorrectly segmented regions as well as class boundaries, the difference between the two being the overall magnitude and continuity exhibited in uncertainty maps (Kwon et al., 2020).

All in all, the investigated approaches frequently produce decomposed uncertainty estimates that do not perfectly correspond to the intuition about pure uncertainty types. This observation will be considered for the qualitative analyses in Section 6.

Varying Training Set Size Epistemic uncertainty indicates "what the model does not know", i. e. which patterns it has not encountered during training. Reducing the training data, thus, naturally constrains a model's knowledge and should therefore increase its epistemic uncertainty. Aleatoric uncertainty, on the other hand, is expected to not change much with varying training set sizes, since the underlying data distribution's label noise does not increase nor decrease, given that the training data is stratified accordingly. As the 32 LiTS cases that constitute the training data are already pre-filtered according to annotation quality, each case is assumed to have roughly the same amount of label noise and cases can be simply dropped to decrease the effective sample size during this experiment.

In this work, the different training sets for each experimental setup are the original training set LiTS-full, subsets of six cases (LiTS-6) , and subsets of two cases (LiTS-2), where LiTS-$2_{\mathrm{i}} \subset$ LiTS-$6_{\mathrm{i}} \subset$ LiTS-full for $i \in 1, 2, 3$. Different subsets for LiTS-6 and LiTS-2 are employed, for each of the three experiment runs, in order to minimise the possibility of any resulting uncertainty changes being caused by a subset-specific pattern in the data. The binary segmentation task is well learnable for the models in this work, even splitting the training data in half does not have much effect on either uncertainty estimates or segmentation performance, as preliminary experiments have shown. The ratio between the training case counts is therefore set to roughly $\frac{1}{5}$.

Both Kendall and Gal (2017) and Kwon et al. (2018) vary the effective training sample size to assess the quality of their uncertainties, as described in Sections 3.6.1 and 3.6.2, respectively. Akin to their results, this work reports the uncertainty as mean uncertainty over all voxels in the test set. Since this is a rather coarse metric, per-case uncertainty estimate distributions are also plotted and the results are validated by qualitatively analyzing the resulting uncertainty maps.

Artificial Label Noise Aleatoric uncertainty stems from label noise and is thus irreducible since it pertains to the underlying data distribution rather than the model itself, as already described in Section 2.5.2. This work therefore evaluates the quality of the uncertainty decomposition methods by comparing the uncertainties of two models which are trained on the same data set but with different amounts of label noise. More specifically, one model is trained on the standard training set of 32 LiTS cases (LiTS-full), whereas the other model is trained on the same set, but artificial label noise is added to one half of the label masks. This artificial noise is created by dilating the liver masks with a max-kernel of size 7x7x1. This altered training set will be henceforth called LiTS-noisy. Figure 4.4 illustrates an example liver mask that is dilated, with the dilated region visible as a pronounced border in a dark green colour. Thus, one half of the training cases in LiTS-noisy have an enlarged liver annotation as ground truth, the other half remains as-is. This setting does not reflect natural label noise as stemming from visual ambiguity at the border of objects, since the voxels added for dilation are highly correlated, occurring either all together or not at all. Instead, the setting mimics training on annotations that were produced with two different annotation regimes and/or labelling post-processing steps. In practice such situations are common in medical data sets that were labelled in different hospitals and where re-labelling by experts would be costly. The model trained on LiTS-noisy is expected to produce higher aleatoric uncertainty around the border of the liver during inference, while the epistemic uncertainty reported by both models should be similar. Furthermore, the same dilation is applied to one half of LiTS-test liver masks in order to compute segmentation performance and calibration. Inferring on vanilla LiTS-test would mean discarding label noise, which is assumed to be inherent in the data itself. Also, the test

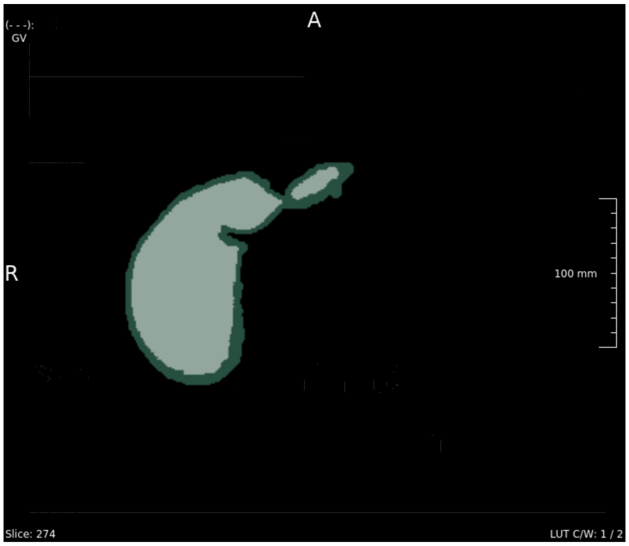


Figure 4.4: Original and dilated liver mask shown in light and dark green, respectively

set is commonly selected to accurately represent the training data.

Thulasidasan et al. (2019) have also added artificial label noise in order to assess the capabilities of predictive uncertainty, as described in Section 3.5.3. However, their evaluation approach differs in that they make a distinction between structured and unstructured noise, where noise is the randomisation of image labels. The authors also did not decompose their predictive uncertainty into uncertainty types. In this work, the artificial label noise-setup is used specifically to evaluate the quality of uncertainty decomposition methods. As with the varying training size experiments described above, the mean uncertainty over all voxels in the noisy test set is reported and the quality of the resulting uncertainty maps will be assessed.

Inference on OOD Medical image segmentation models should indicate at test time via their epistemic uncertainty whether a given CT scan is in distribution (ID) or out of distribution (OOD), i. e. the likeliness of being derived from the same underlying distribution as the model's training data. There are numerous variants of OOD samples that might occur in a practical clinical setting, for example CT scans whose anatomical plane, use of contrast, or imaging technique/protocol diverges from what was seen during training.

Even a simple 2D rotation of a CT volume represents an OOD sample, given that the model was not trained with a corresponding data augmentation regime. Identifying such samples is important, since the model learnt the typical form, angle, and approximate position of certain objects, so that inferring on a rotated image might result in wildly different segmentations than inference on the same image with proper rotation. Epistemic uncertainty is expected to filter out such cases, as this type of uncertainty indicates the model's degree of familiarity with patterns in the image. Aleatoric uncertainty, conversely, represents label noise and is, thus, expected to stay the same. Since expert annotators are generally able to recognise e. g. rotated OOD CT scans, they can adapt and robustly label these cases regardless, as such they typically contain the same amount of label noise as ID cases.

For this experimental setting all models trained on LiTS-full are evaluated on 33 test cases that were rotated 180 degrees, henceforth called LiTS-rot. This situation is not too far-fetched for the clinical setting, since this rotation can be achieved by simply leaving out the resampling step which normally transforms the CT scans to a specific anatomical plane. In addition to this experimental setting, the models trained on LiTS-full are also used to infer on one CT scan taken from a Fraunhofer-MEVIS-internal data set which is employed for therapy planning of selective internal radiation therapy (SIRT). The case displays a severe form of ascites, which is not encountered to this extent in the LiTS data set. As such, it constitutes another type of OOD sample by reflecting yet another shift in distribution.

If the predictive uncertainty is properly decomposed, inferring on an OOD sample should naturally result in higher epistemic uncertainty, while aleatoric uncertainty remains largely constant. Note that aleatoric uncertainty is not expected to be perfectly stable, since the proportion of predicted class boundaries might be higher for certain OOD images, in which case an increase in aleatoric uncertainty might occur.

4.4.1 Statistical Significance

In order to reduce the impact of the stochasticity of the training due to randomly initialised weights and random ordering of mini-batches, three models are trained for each network architecture and experimental setting. The three resulting per-case uncertainty outcomes are then averaged across the three trained models for plotting the distributions of uncertainty estimates per architecture and experimental setting as well as for computing the statistical significance of their differences.

In order to compute the statistical significance of performance and calibration differences as well as increases or decreases in a model's predicted uncertainty, the Wilcoxon signed-rank test and the paired student's t-test are employed. Pairwise comparisons are performed for the per-case mean model results as averaged across the three models per architecture. Thus, both mean results are for the same individual, i. e. same test case, which is a good fit for paired difference tests. The Wilcoxon signed-rank test is performed for most uncertainty comparisons, since the uncertainty output distributions of estimates derived via MI decomposition are not normally distributed. The assumption of non-normality is validated via the Shapiro-Wilk test, the results of which are depicted in Tables 1 and 2. Within this work, the null hypothesis of normality for all distributions is rejected at the 0.05 significance level. Most mean activations of the HLSN follow a normal distribution. The statistical significance of differences between those values is thus computed via the student's t-test.

The Wilcoxon signed-rank test statistics reveal whether a sample from one population is greater than one from the other population with a chance of more than 50%. Within this work, the null hypothesis of the median of the differences between the distributions being zero is rejected, if the p value is smaller than 0.05. The student's t-test computes the difference between two population means for pairs of random population samples following a normal distribution. Within this work, if the p value is smaller than 0.05 the null hypothesis of the population means being equal is rejected.

Results

5.1 Segmentation Performance

Table 5.1 shows the segmentation performance computed over the 33 cases of the held-out test set LiTS-test for the pure-NN models, HUN- and HLS models, BNN-MI, and BNN-HLS models trained on LiTS-full employing the cross-entropy loss. Segmentation performance is computed via the mean Dice coefficient and IoU across three models per architecture, all individual segmentation results are listed in the Appendix 4. The reported differences in segmentation performance are relatively small and the ranges of the segmentation performances of three models which are trained per architecture overlap considerably, thus no statistically significant performance differences are found, as shown in Table 5.2.

Table 5.3 shows the segmentation performance as computed on LiTS-test for the BNN-MI and BNN-HLS models trained on LiTS-full employing the soft Dice loss. Segmentation performance is slightly lower than for models trained with cross-entropy loss. The difference in segmentation performance between BNN-MI and BNN-HLS models is negligible. Note, that in this case only one model per architecture was trained, thus, no statistical significance was computed for the mean segmentation performance in this case.

Table 5.1: Mean Dice coefficient and IoU as measured on the test set with the pure-NN model, HUN- and HLS models, BNN-MI, and BNN-HLS models trained with cross-entropy loss on LiTS-full

	Dice	IoU
pure-NN	0.941	0.89
HUN	0.946	0.90
HLS	0.945	0.90
BNN-MI	0.939	0.89
BNN-HLS	0.947	0.90

Table 5.2: Wilcoxon signed-rank test results for the difference in segmentation performance between the BNN-MI and BNN-HLS models computed for several experiment settings. The respective training/test sets are indicated in the first column.

	t	p
LiTS	2.0	0.75
LiTS-6	3.0	1.0
LiTS-2	0.0	0.25
LiTS-noisy	2.0	0.75
LiTS-rot	1.0	0.5

Table 5.3: Mean Dice coefficient and IoU as measured on the test set with BNN-MI and BNN-HLS models trained with soft Dice loss

	Dice	IoU
BNN-MI	0.922	0.855
BNN-HLS	0.939	0.885

Table 5.4 shows the segmentation performance for the BNN-MI and BNN-HLS models trained with cross-entropy loss for all other experimental settings, namely with varying training set sizes (LiTS-6 and LiTS-2), added artificial label noise in the training and test data (LiTS-noisy), and trained on LiTS-full but inferring on the test set LiTS-rot.

Segmentation performance for BNN-MI and BNN-HLS models:

- is extremely poor when inferring on OOD samples from LiTS-rot , even worse than naively predicting the background class for all voxels.

- positively correlates with the size of the training set, with highest performance when trained on LiTS-full.

- is lower when trained on LiTS-noisy than for models trained on LiTS-full.

Across experiment settings, BNN-MI and BNN-HLS models perform largely

Table 5.4: Mean Dice coefficient and IoU for the performance of the BNN-MI and BNN-HLS models trained with cross-entropy loss on different training sets and inferring on different test sets, as indicated in the first row.

	LiTS-rot		LiTS-6		LiTS-2		LiTS-noisy	
	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU
BNN-MI	0.041	0.02	0.853	0.74	0.78	0.64	0.744	0.59
BNN-HLS	0.071	0.04	0.864	0.76	0.75	0.6	0.835	0.72

similarly with only marginal differences between their mean performances. There are no statistically significant differences, as reported in Table 5.2.

5.2 Calibration

Table 5.5 shows the mean calibration computed over the 33 cases of the held-out test set LiTS-test for the pure-NN model, HUN- and HLS models, BNN-MI, and BNN-HLS trained on LiTS-full employing the cross-entropy loss. Calibration is computed via the mean NLL, ECE, and MCE across three models per architecture. Overall, calibration for models trained with cross-entropy loss is extremely high. The differences in calibration between model architectures are marginal, with a slightly better calibration for BNN-MI and BNN-HLS. The ranges in calibration of the three models which are trained per architecture overlap considerably, thus no statistically significant performance differences are found, see also Table 5.6. The reliability diagrams of the "representative median models", i. e. the model with the median NLL across the three trained models, for BNN-MI and BNN-HLS are plotted in Figure 5.1. The reliability diagrams of all models trained on LiTS-full can be found in Appendix E.

The calibration of the BNN-MI and BNN-HLS models trained with soft Dice loss is noticeably worse, as can be seen in Table 5.7. Meanwhile, both BNN-MI and BNN-HLS are markedly better calibrated than the pure-NN model, as can be seen in the reliability diagrams in Figure 5.2. There are no statistically significant differences between the calibration of BNN-MI and BNN-HLS, as reported in Table 5.6.

Table 5.5: Mean calibration metrics as measured on the test set for the pure-NN model, HUN- and HLS models, BNN-MI, and BNN-HLS models trained with cross-entropy loss on LiTS-full. All measures are given in percent.

	NLL	ECE	MCE
pure-NN	0.7	0.1	6.6
HUN	0.7	0.7	4.8
HLS	0.7	0.07	6.0
BNN-MI	0.67	0.07	3.8
BNN-HLS	0.6	0.00	2.7

Table 5.6: Wilcoxon signed-rank test results for the difference in calibration between the BNN-MI and BNN-HLS models as measured via NLL. The different training and test sets are indicated in the first column.

	t	p
LiTS-full	1.0	0.5
LiTS-6	2.0	0.75
LiTS-2	1.0	0.65
LiTS-noisy	2.0	0.75
LiTS-rot	2.0	0.75

Tables 5.8 and 5.9 show the mean calibration for BNN-MI and BNN-HLS models employing cross-entropy loss for the remaining experiments, namely with varying training set sizes (LiTS-6 and LiTS-2), added artificial label noise in the training and test data (LiTS-noisy), and trained on LiTS-full but inferring on the test set LiTS-rot.

The calibration of BNN-MI and BNN-HLS models:

- is poor when inferring on OOD samples from LiTS-rot.
- negatively correlates with the size of the training set, with best calibration when trained on LiTS-full.
- worse for models trained on LiTS-noisy than when training and inferring on LiTS-full and LiTS-test.

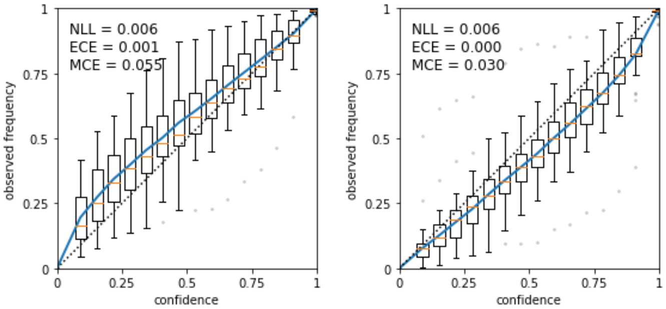


Figure 5.1: Reliability diagrams for a BNN-MI (L) and a BNN-HLS (R) model, trained with cross-entropy loss on LiTS-full

Table 5.7: Mean calibration metrics as measured on the test set for BNN-MI and BNN-HLS models trained with soft Dice loss on LiTS-full. All measures are given in percent.

	NLL	ECE	MCE
BNN-MI	1.8	0.1	38
BNN-HLS	1.6	0.2	18.1

Across experiment settings, the calibration of BNN-MI and BNN-HLS are largely similar with only marginal differences between their mean calibration. There are no statistically significant differences, as reported in Table 5.6.

In Appendix F, several example predictive uncertainty maps are shown for all model architectures, i. e. pure-NN, HUN- and HLS models, BNN-MI, and BNN-HLS. For the HUN- and HLS models, uncertainty maps are provided for both the entropy of the softmax scores as well as the activation of the HLSN and the HUN. For BNN-MI and BNN-HLS the predictive entropy is used as overall predictive uncertainty measure. Overall, all uncertainty measures for all model architectures highlight segmentation class boundaries and show up slightly in tumor regions for which the model is also unsure in its general prediction.

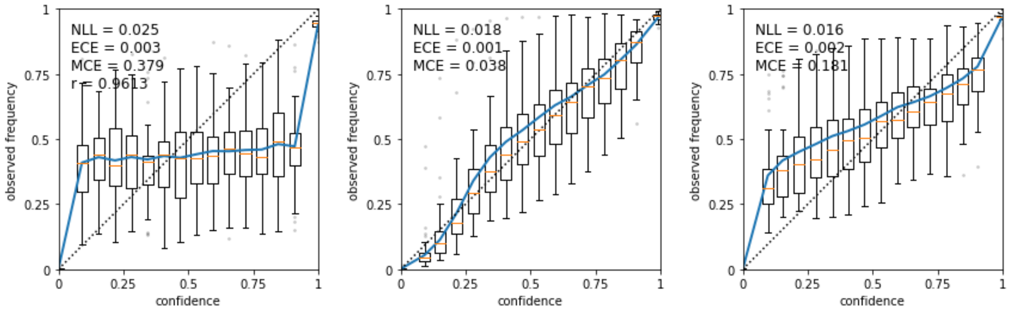


Figure 5.2: Reliability diagrams for a pure-NN (L), a BNN-MI (M), and a BNN-HLS (R) model, trained with soft Dice loss on LiTS-full

Table 5.8: Mean calibration metrics as measured on the test set for BNN-MI and BNN-HLS models trained with cross-entropy loss on LiTS-6 and LiTS-2. All measures are given in percent.

	LiTS-6			LiTS-2		
	NLL	ECE	MCE	NLL	ECE	MCE
BNN-MI	2.2	0.2	8.7	3.2	0.4	10.3
BNN-HLS	2.0	0.1	8.4	3.8	0.5	15.0

5.3 Decomposed Uncertainties

In the following, the resulting aleatoric and epistemic estimates derived via uncertainty decomposition performed on the BNN-MI and BNN-HLS models are reported for all experiment settings.

Please note that the HLSN-derived aleatoric uncertainty is plotted as $1 - \text{HLSN-activation}$, in order to streamline the discussion of the uncertainty maps by consistently using the term ”uncertainty” instead of ”confidence”. Also, the liver masks used for training were thresholded so that liver tumors were assigned the same class as liver. The original unthresholded ground truth maps are shown here in order to be able to investigate the interaction between liver tumors and corresponding segmentation and uncertainty estimates. All reported uncertainty maps are produced by the representative, median models, as chosen according to the NLL calibration metric. Due to

Table 5.9: Mean calibration metrics as measured on the LiTS-rot test set for BNN-MI and BNN-HLS models trained with cross-entropy loss on LiTS-full. All measures are given in percent.

	LiTS-rot			LiTS-noisy		
	NLL	ECE	MCE	NLL	ECE	MCE
BNN-MI	18.7	2.3	16.9	2.1	0.6	16.4
BNN-HLS	19.2	2.3	22.6	1.4	0.5	14.7

MC dropout, patch borders might appear inside an activation or segmentation map. This is a consequence of using different dropout probabilities for each patch, and the mean of those forward passes producing slightly different results. The default VOI LUT (used to transform voxel values prior into more human-readable ranges prior to rendering) of 0.5/1 is changed whenever the highlighted structures would otherwise not be visible. LUT values are annotated at the bottom right of each image.

5.3.1 Overall Qualitative Results

Figure 5.3 shows example uncertainty maps derived from the BNN-MI and the BNN-HLS models alongside the corresponding original image and ground truth liver mask. Further examples are included in Appendix G.

All aleatoric estimates, i.e. aleatoric uncertainty as computed via MI decomposition on BNN-MI and BNN-HLS, as well as the activation of the HLSN consistently highlight the segmentation class boundaries, as can be seen in the provided uncertainty maps. Epistemic uncertainty is frequently similar to the aleatoric estimates, though less focal and appearing only partially around segmented object borders, and having vastly lower magnitude overall. Moreover, the aleatoric uncertainty maps from BNN-MI and BNN-HLS as derived via MI decomposition are ”cleaner” in contrast to those estimates derived from the HLSN-activation. The former mostly constitute a thin, high-valued border around the segmented objects, whereas the latter sometimes dimly highlights the borders of other structures, as well. The magnitude of aleatoric uncertainty values derived via MI decomposition is generally higher than that of the HLSN-activation.

For several tumor regions, segmentation accuracy drops slightly, as can

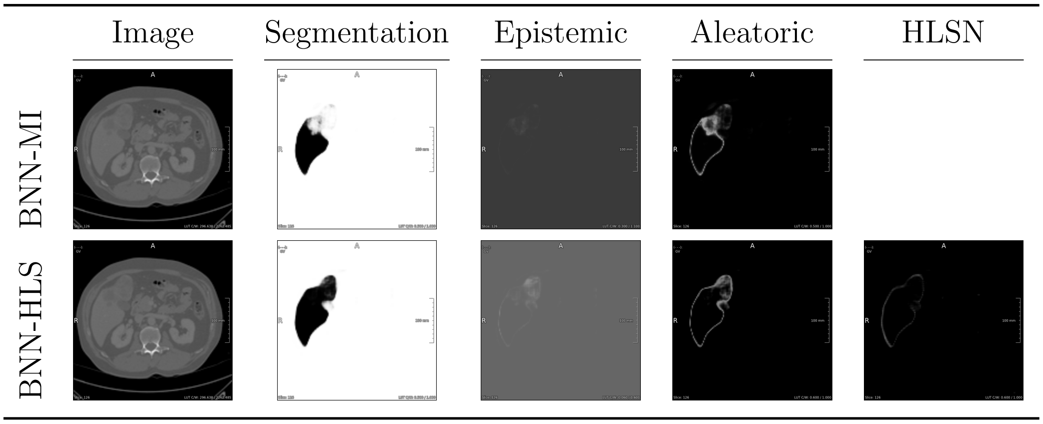


Figure 5.3: Example uncertainty maps derived from BNN-MI and BNN-HLS models trained on LiTS-full

be seen in the top two rows for both BNN-MI and BNN-HLS. In those cases, both aleatoric and epistemic uncertainty estimates indicate an increase in uncertainty.

5.3.2 Experiment Results

Varying Training Set Size

Varying the training set size visibly alters the resulting uncertainty maps. Some examples are shown in Figures 5.9 and 5.10. Aleatoric uncertainty as derived via MI decomposition becomes less delineated with less training data. The activation of the HLSN becomes more spread out, as well, projecting notably into the segmented liver region. Epistemic uncertainty becomes significantly larger, mostly around the class boundaries (of which there are more for models trained on LiTS-6 and LiTS-2 because the upper liver lesions are segmented as non-liver). Additionally, the epistemic uncertainty of both model architectures tends to highlight the spleen or kidneys when trained on LiTS-2, an example of which is shown in Figure 5.10.

Quantitatively, the different training set sizes have a significant impact on the epistemic uncertainties of models from both architectures, as shown in Tables 5 and 6. Smaller training sets correlate with higher epistemic uncertainty, as is reflected in the overall test set voxel means, as seen in Table 5.10, as well as in the violin plots of the per-case uncertainty estimates

Table 5.10: Mean aleatoric and epistemic uncertainty estimates averaged over all voxels in the test set and across the three respectively trained models for BNN-MI and BNN-HLS. The first column denotes the respective experiment setting, indicating the training/test sets used.

	BNN-MI		BNN-HLS		
	aleatoric	epistemic	HLSN	aleatoric	epistemic
LiTS-full	0.0080	8.40e-4	0.0203	0.0055	6.62e-4
ascites (1 case)	0.0181	2.82e-3	0.0273	0.0195	3.39e-3
LiTS-noisy	0.0276	1.70e-3	0.0153	0.0301	1.54e-3
LiTS-6	0.0092	1.84e-3	0.0258	0.0119	1.99e-3
LiTS-2	0.0129	2.52e-3	0.0305	0.0073	2.35e-3
LiTS-rot	0.0069	1.15e-3	0.0161	0.0051	1.49e-3

in Figures 5.4 and 5.6. While the aleatoric uncertainty estimates as computed via MI decomposition increase for LiTS-2 as well, the uncertainty derived from the HLSN emits significantly larger values for models trained on both LiTS-6 and LiTS-2.

Artificial Label Noise

Dilating the liver mask of one half of the training samples, as described in Section 4.4, visibly alters the resulting aleatoric uncertainty maps. Two examples are shown in Figure 5.11 for the aleatoric uncertainty as derived via MI decomposition for both BNN-MI and BNN-HLS. The aleatoric uncertainty estimates are slightly higher and considerably more spread out around the boundary of the liver class, visibly projecting into the liver in some parts, for both models when trained on LiTS-noisy. Moreover, the models trained on LiTS-noisy are visibly more uncertain about smaller liver parts that are sticking out of the main structure, blurrying most into a more homogeneous liver mass. On the other hand, the aleatoric uncertainty computed via the activation of the HLSN, as shown in Figure 5.12, exhibits no clear tendencies towards more or less aleatoric uncertainty. The training-inherent stochasticity due to random initialisation of weights produces considerably different HLSN-activation patterns. Across randomly initialised models, the resulting HLSN-activation ranges from less pronounced than when trained on LiTS-

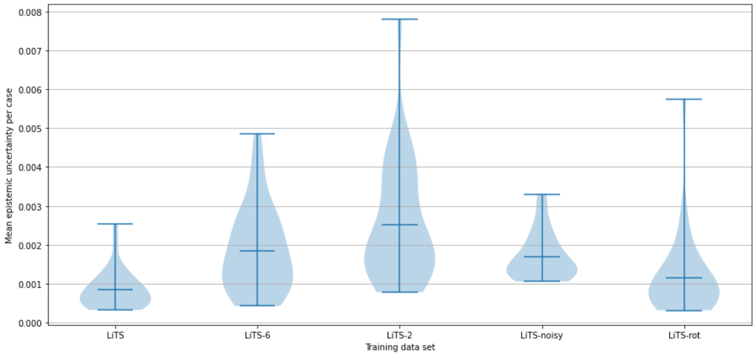


Figure 5.4: Mean per-case epistemic uncertainty estimates of BNN-MI models trained on different training set sizes and artificial label noise levels, evaluated on the test set.

full, over showing a definitive region of uncertainty around the segmented liver, to not only producing a wide border of uncertainty, but the aleatoric uncertainty also projecting into the liver itself. Epistemic uncertainty for both BNN-MI and BNN-HLS shows no strong difference when training on LiTS-full and LiTS-noisy. The epistemic estimates, which line the border of the segmented liver, appear more blurred overall for models trained on LiTS-noisy. Additionally, a tiny amount of epistemic uncertainty encompasses the broadened segmentation and aleatoric uncertainty.

Quantitatively evaluating all predicted uncertainties on the test set confirms the qualitatively observed rise in aleatoric uncertainty as computed via MI decomposition. The mean uncertainty values across all voxels in the test set are shown in Table 5.10. They show notably increased aleatoric uncertainty and a slightly raised mean of the epistemic uncertainty, for both models when trained on LiTS-noisy. The uncertainty derived from the activation of the HLSN, on the other hand, significantly decreases for models trained on LiTS-noisy. Figures 5.4 to 5.7 provide a more fine-grained view on the resulting decomposed uncertainty estimates when averaged per-case over the three respectively trained models for both architectures BNN-MI and BNN-HLS. Both BNN-MI and BNN-HLS produce not only higher mean aleatoric uncertainties, but also more spread-out aleatoric estimates with a considerable number of cases in the upper quartile when trained with LiTS-noisy as compared to training in the normal setting. This increase in aleatoric

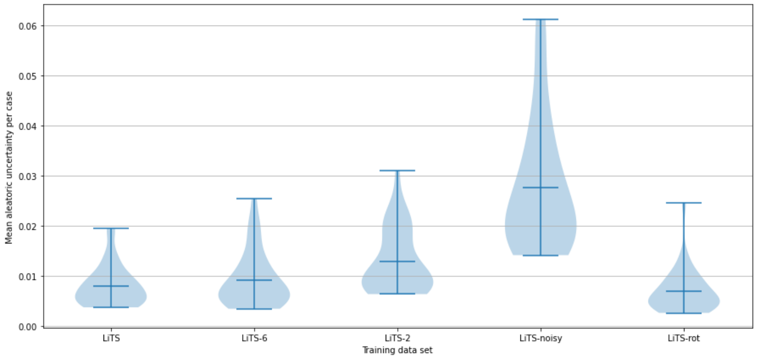


Figure 5.5: Mean per-case aleatoric uncertainty estimates of BNN-MI models trained on different training set sizes and artificial label noise levels, evaluated on the test set.

uncertainty is statistically significant for both models, as shown in Tables 5 and 6. While the mean of the per-case epistemic uncertainty estimates is also higher for LiTS-noisy, the distribution of epistemic estimates is a lot less variable, as can be seen in the respective violin plots in Figures 5.4 and 5.6.

Inference on OOD

Inferring on rotated CT scans in LiTS-rot with models from both architectures, BNN-MI and BNN-HLS, results in extremely poor performance, as already seen in Table 5.3. The models predict several regions, most notably the spleen, as liver, which can be seen in Figure 5.13. Most segmented regions are located on the left-hand side of the patient, which corresponds to the liver's original location in the unrotated training CT scans. The reliability of all models is accordingly bad, as seen in Table 5.9.

Quantitatively, epistemic uncertainty estimates are slightly elevated in comparison to models which both train on LiTS-full and infer on the normal LiTS-test set, while all aleatoric uncertainty estimates remain largely the same, as seen in Table 5.10.

This rise in epistemic uncertainty is reflected in the uncertainty maps, as shown in Figure 5.13. Aleatoric and epistemic uncertainty for BNN-MI and BNN-HLS both highlight regions of the original liver, with only aleatoric uncertainty still strongly appearing around the border of the confidently segmented spleen, as well. As opposed to the preceding uncertainty maps

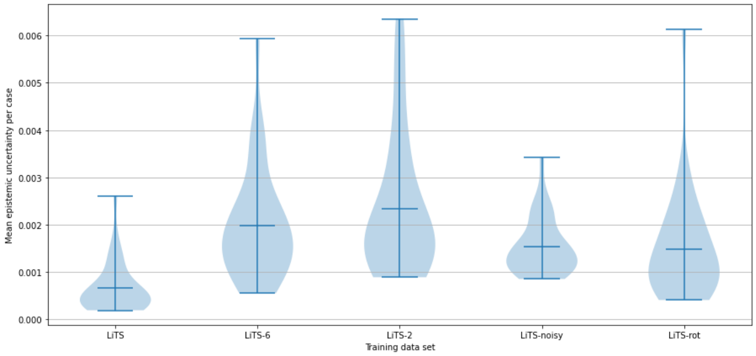


Figure 5.6: Mean per-case epistemic uncertainty estimates of BNN-HLS models trained on different training set sizes and artificial label noise levels, evaluated on the test set

for inference on ID data, the increased epistemic uncertainty estimates are overall higher in magnitude.

The inference of both BNN-MI and BNN-HLS models on the CT scan of a patient with severe ascites, which was not included in the training set, results in strong overestimation of the liver region, with all models predicting large parts of fluid in the abdomen as well as a separate part of the lower abdomen, as belonging to the liver. The segmentations also demonstrate noticeably less delineated borders, when compared to segmented livers on the ID test set, as can be seen in the 3D visualisation in Figure 5.14 for two segmentations of an ID and the OOD ascites-case, respectively.

Figure 5.15 shows that aleatoric and epistemic uncertainty estimates highlight the boundary of the segmented liver region, as well as shading the part of the fluid that is incorrectly predicted as liver by both models.

Quantitatively, inferring on this OOD case slightly raises all aleatoric uncertainty estimates, and drastically increases the epistemic uncertainty estimates of both BNN-MI and BNN-HLS, as shown in Table 5.10.

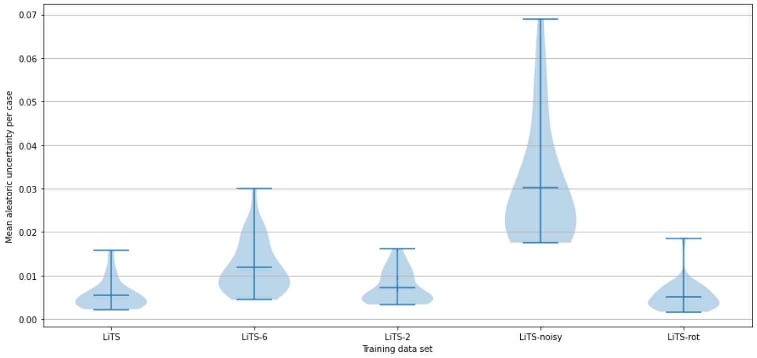


Figure 5.7: Mean per-case aleatoric uncertainty estimates of BNN-HLS models trained on different training set sizes and artificial label noise levels, evaluated on the test set.

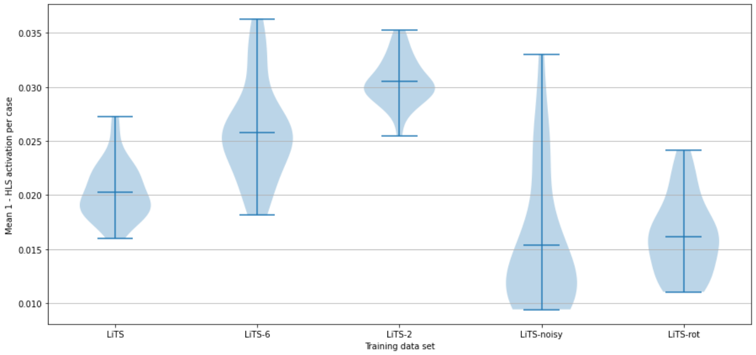


Figure 5.8: Mean per-case aleatoric uncertainty estimates derived from the HLSN-activation of BNN-HLS models trained on different training set sizes and artificial label noise levels, evaluated on the test set.

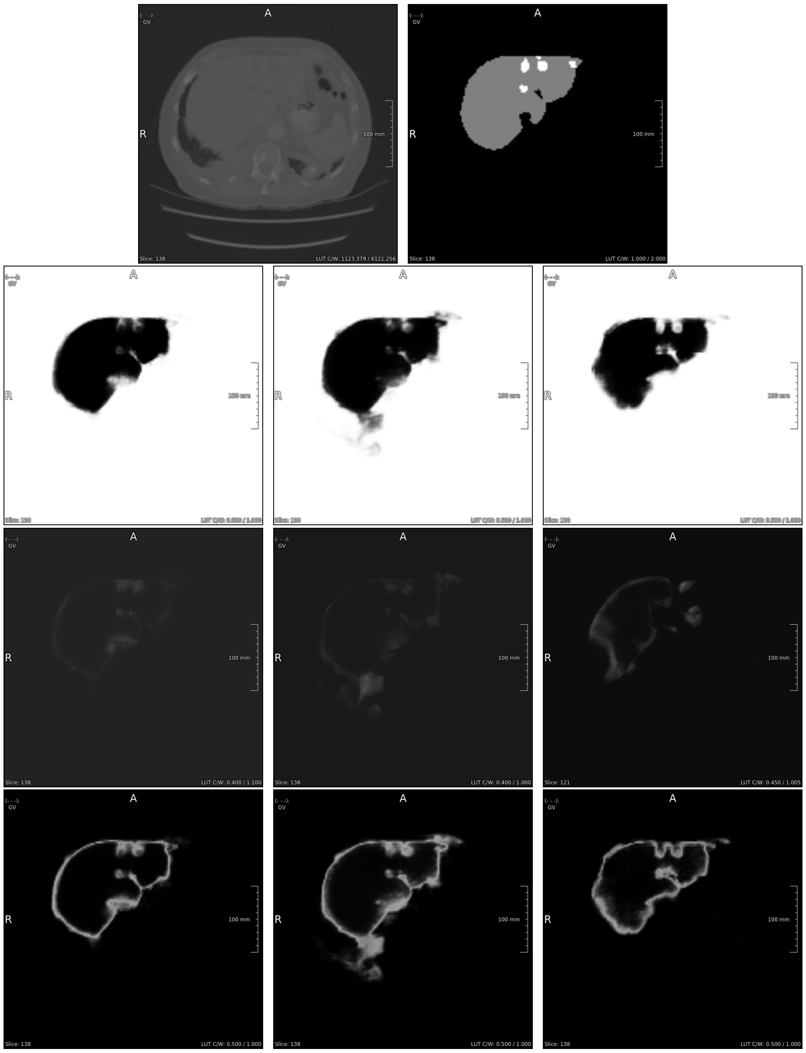


Figure 5.9: Example uncertainty maps for BNN-MI trained on LiTS-full, LiTS-6, and LiTS-2 (L to R). Original image and ground truth liver mask are shown in the top row, then, from top to bottom: segmentation, epistemic, and aleatoric uncertainty maps as computed via MI decomposition.

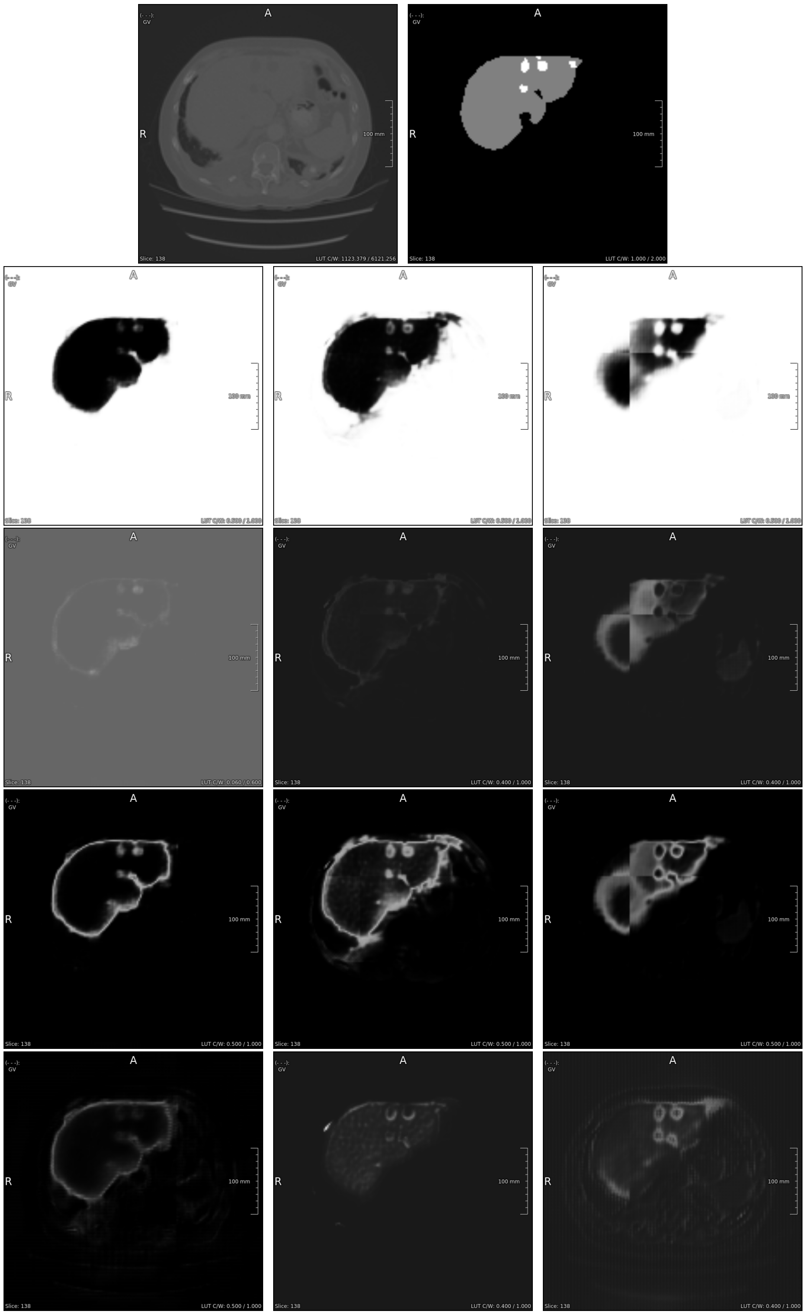


Figure 5.10: Example uncertainty maps for BNN-HLS trained on LiTS-full, LiTS-6, and LiTS-2 (L to R). Original image and ground truth liver mask are shown in the top row, then, from top to bottom: segmentation, epistemic and aleatoric uncertainty as computed via MI decomposition, and HLSN-derived aleatoric activation maps.

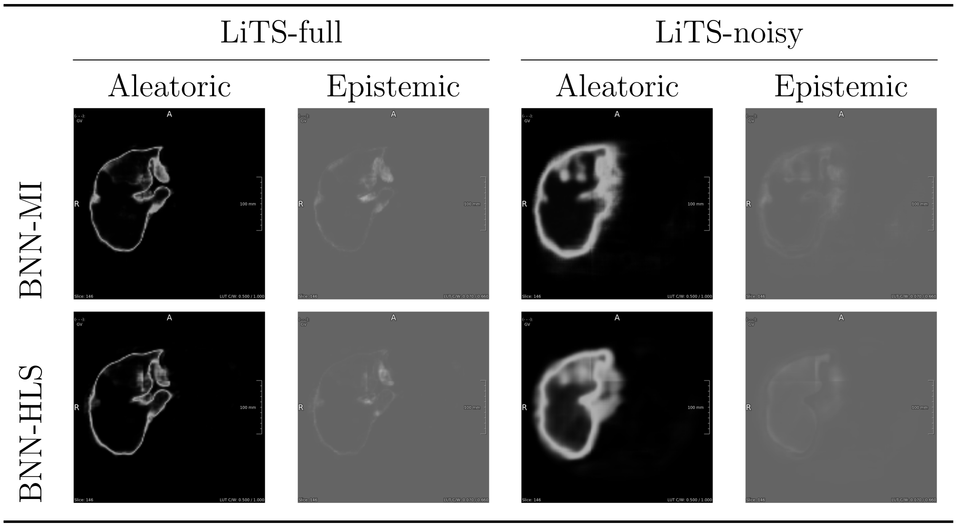


Figure 5.11: Example uncertainty maps for BNN-MI and BNN-HLS models when trained on LiTS-full vs. LiTS-noisy

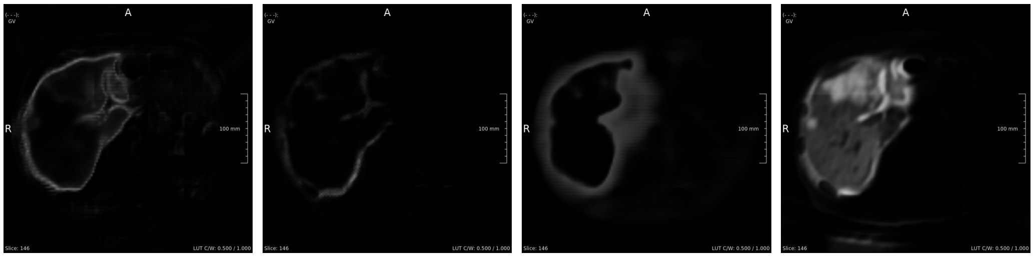


Figure 5.12: Example activation map of the HLSN in BNN-HLS models when trained on LiTS-full (left) vs. 3 different variants when trained on LiTS-noisy

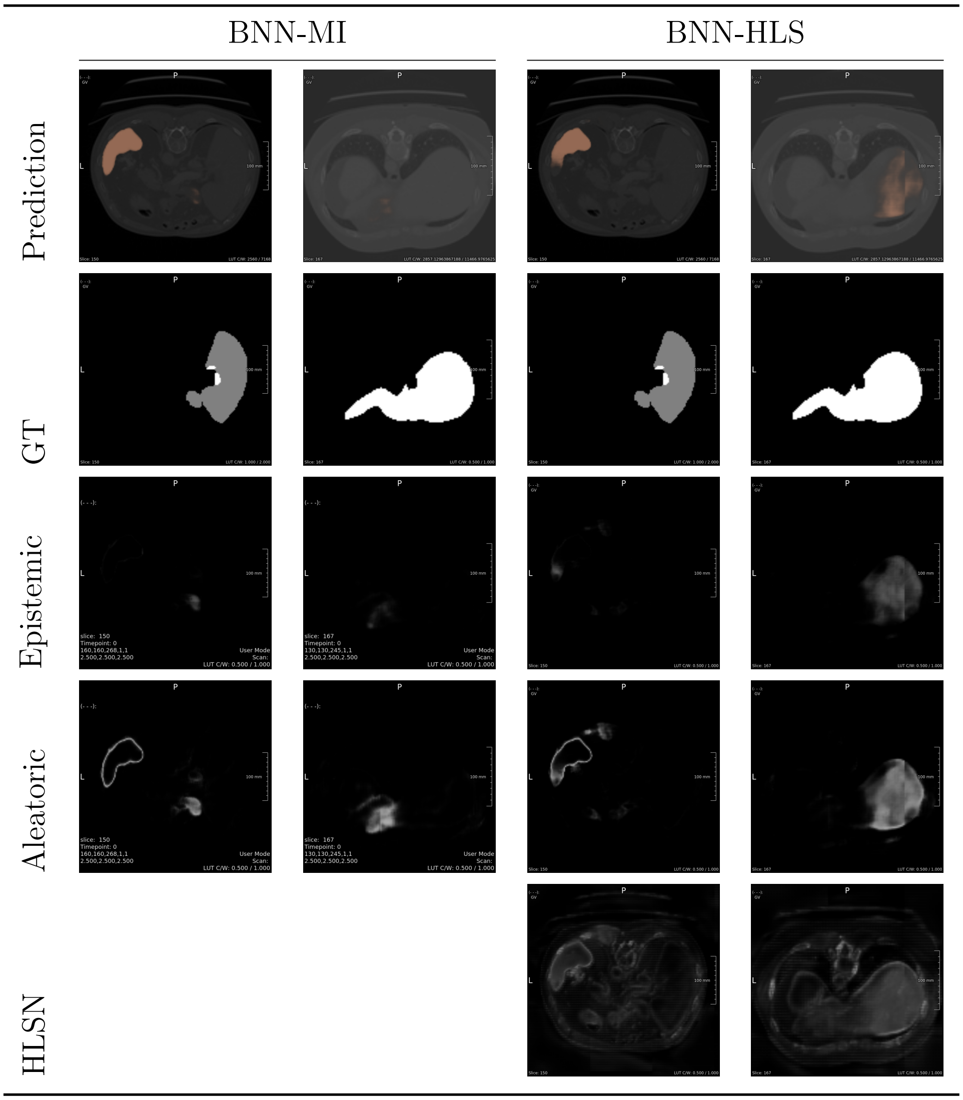


Figure 5.13: Aleatoric and epistemic uncertainty maps derived from BNN-MI and BNN-HLS when trained on LiTS-full and inferring on LiTS-rot.

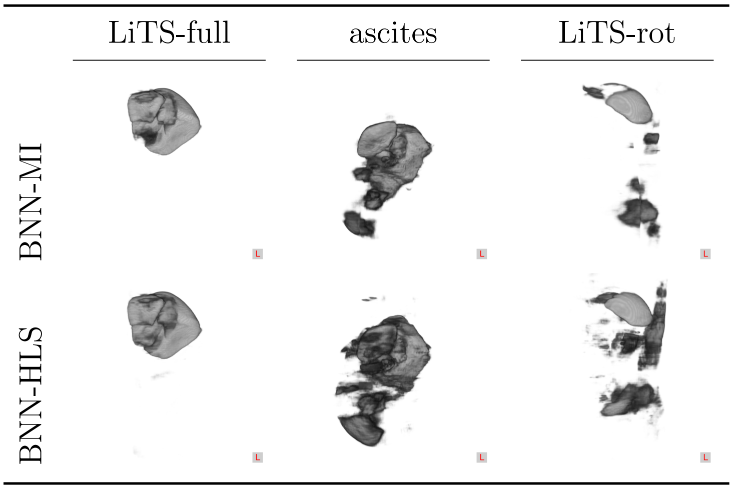


Figure 5.14: Example segmentations for BNN-MI and BNN-HLS on an ID and two OOD test cases

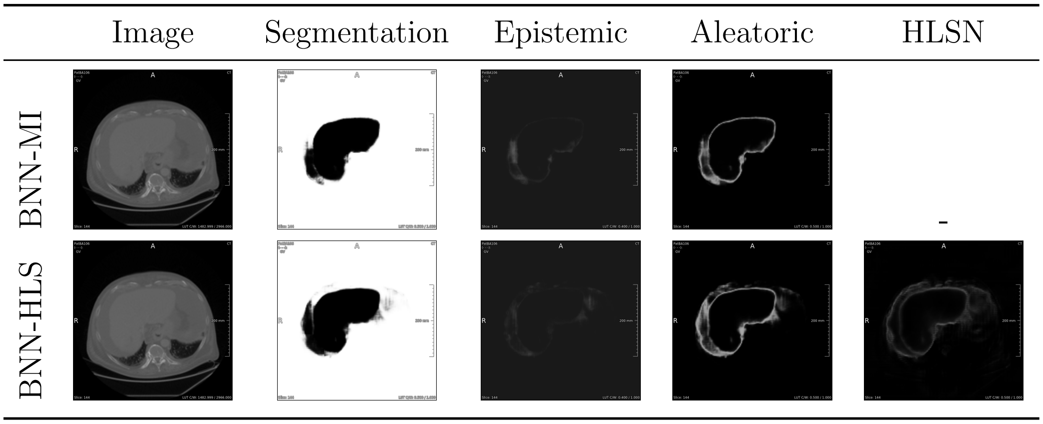


Figure 5.15: Aleatoric and epistemic uncertainty maps derived from BNN-MI and BNN-HLS models when trained on LiTS-full and inferring on an OOD case with severe ascites.

Discussion

This chapter analyses the results of the experiments detailed in the previous chapter, relates them to existing literature, and uses them to answer this work's research question. As per the latter, the main focus lies on the quality of the decomposition of the overall predictive uncertainty into aleatoric and epistemic components given two different architectures:

- BNN-MI, which uses the MI decomposition on a regular BNN's predictive distribution, and
- BNN-HLS, whose predictive distribution can be decomposed in the same way, but whose HLSN provides an additional indicator for aleatoric uncertainty, independent of the epistemic MC sample variance.

The main portion of this chapter is a thorough analysis of the decomposed uncertainties and their behaviour for different experiment settings, with some further remarks about the use of the HLSN-activation as an aleatoric uncertainty estimate. Observations concerning the behaviour of uncertainty decomposition for models that were trained with soft Dice loss instead of cross-entropy loss are also included at the end.

6.1 Overall Performance and Calibration

Overall, the segmentation performance and calibration of all models trained on LiTS-full is very high. This result is not surprising given the relatively simple binary segmentation task, and is consistent with the results of previous works (Bilic et al., 2019). As expected, both reliability and segmentation performance degrade for all models with decreasing training set size and increasing artificial label noise. In contrast to Kendall and Gal (2017), this work found no statistically significant performance difference for either BNN-MI nor BNN-HLS compared to the pure-NN model. However, Kendall and Gal found larger performance improvements for their more challenging data set. Thus, the missing segmentation performance improvement in this work might be explained by the given problem setup, with much less objects to segment, two instead of 11 or 40 classes, and larger segmented objects in general. These factors simplify the segmentation task, leading to very high baseline performance, which in turn renders performance improvements harder to achieve.

The total uncertainty of pure-NN, HUN- and HLS models occurring mostly at class boundaries conforms to the expectation of it mainly reflecting aleatoric uncertainty. The probabilistic predictions of vanilla NNs encode learnt label noise but lack a model confidence estimate, which could act as epistemic uncertainty, as explained in Section 3.1. The predictive entropy maps for BNN-MI and BNN-HLS are also heavily dominated by the aleatoric uncertainty values, being both more focal and larger in magnitude.

6.2 Decomposed Uncertainties

Overall, the expected differences in uncertainty are reflected in the mean uncertainties when averaged over all voxels in the test set, as shown in Table 5.10. While the uncertainty estimates derived via MI decomposition largely correspond to the intuitive behaviour of the respective uncertainty types, the activation of the heteroscedastic logit smoothing neuron (HLSN) does not do so consistently. Furthermore, the variance of the per-case estimates over the test set for MI-decomposed uncertainty types, as reported in Figures 5.4 to 5.7, provides another, even clearer distinction that matches the intuition about the behaviour of aleatoric and epistemic uncertainty estimates.

Epistemic uncertainty estimates are generally low which can be explained by the binary classification setting. The multi-class road and indoor scene segmentation tasks in Kendall and Gal (2017) illustratively provide epistemic uncertainty estimates for instances of rare classes and visually difficult pixels, of which there are next to none in the binary liver segmentation setting. Moreover, epistemic uncertainty highlights segmented class boundaries across experiment settings. While not fully conforming to the definition of epistemic uncertainty, since one would expect this region to be difficult due to label noise, i.e. data-inherent aleatoric uncertainty, these findings are consistent with other works and have been anticipated, as described in Section 4.4. Both Mobiny et al. (2021) and Kendall and Gal (2017) derive epistemic estimates that also outline the borders of segmented objects.

6.2.1 Varying Training Set Size

Decreasing the number of cases in the training set is expected to raise epistemic uncertainty with little impact on aleatoric uncertainty. However, slightly raised aleatoric uncertainty on smaller training sets does not necessarily signify the employed uncertainty decomposition is inaccurate. Aleatoric uncertainty might occur around the borders of incorrectly or over-segmented regions, whose number increases with decreasing training set sizes. Conversely, slightly lowered aleatoric uncertainty might occur, since smaller data sets contain in fact less label noise, as there are simply less instances over which annotators might disagree.

In the experiments performed in this work, decreasing the number of cases in the training set significantly raises epistemic uncertainty while having little impact on aleatoric uncertainty for both BNN-MI and BNN-HLS models, as shown in Figures 5.4, 5.5 and 5.6, 5.7, respectively. The experiments also show that aleatoric uncertainty is significantly increased for the BNN-MI models trained on smaller training data, as are the aleatoric uncertainty estimates of BNN-HLS models trained on LiTS-2, while BNN-HLS on LiTS-6 exhibits slightly decreased aleatoric uncertainty. This observation fits in with the situation described previously. Training on smaller data sets results in models that are less accurate e .g. segmenting liver lesions as non-liver, as shown in Figures 5.9 and 5.10. The irregular shape in turn increases the amount of segmented class boundaries, which raises aleatoric uncertainty.

Interestingly, for both model architectures trained on LiTS-2, epistemic uncertainty sometimes highlights the spleen or kidney while the aleatoric uncertainty estimates are clearly located solely in the liver region. This finding demonstrates that the model has not yet accurately learnt the concept and features of the liver differentiating it from other organs with similar radiodensity. Thus, it mistakes the similar-looking spleen for a liver instead in some forward passes. This observation conforms to the definition of epistemic uncertainty as indicating model uncertainty and is in line with the results by Kendall and Gal (2017) who found epistemic uncertainty to identify ”visually challenging” pixels.

6.2.2 Artificial Label Noise

Both BNN-MI and BNN-HLS predict significantly higher aleatoric uncertainty when trained on LiTS-noisy, a data set in which one half of the liver masks is artificially dilated, than when trained on the original LiTS-full, as shown in Table 5.10. Qualitatively, the resulting aleatoric uncertainty maps confirm these results, with broader bands of uncertainty at class boundaries for aleatoric uncertainty estimates derived via MI decomposition. The models' behaviour, thus, conforms to the intuition of aleatoric uncertainty as capturing label noise. The activation of the HLSN, on the other hand, does not consistently lead to increased aleatoric uncertainty at class boundaries, and in fact its mean over the LiTS-test voxels is significantly decreased. Epistemic uncertainty, while quantitatively slightly increased, appears more blurred and outlines both the original, undilated liver and the broadened outline of aleatoric uncertainty. The latter finding is unexpected, since it does not directly translate to the intuition about epistemic uncertainty, since no visually challenging or rare patterns were introduced. However, considering that epistemic uncertainty does occur lightly around objects in other works as well (Mobiny et al., 2021; Kendall and Gal, 2017), and given that the models were trained with essentially two kinds of liver outlines, observing epistemic uncertainty around both learnt contours appears reasonable. Meanwhile, the epistemic estimates largely leaving out regions, where artificial label noise was introduced during training, does reflect a clean separation between the two uncertainty types.

In sum, the aleatoric uncertainty as derived via MI decomposition reliably indicates the artificially introduced inter-annotator disagreement. This finding agrees with the work conducted by Thulasidasan et al. (2019), in which the loss-attenuating abstention networks, as described in Section 3.5.3, act as effective data cleaners for artificially introduced label noise. In a real-world setting, increased aleatoric uncertainty might thus be seen as a warning sign and warrant a manual investigation of the annotations in order to evaluate their suitability for the given task.

6.2.3 Inference on OOD

When inferring on LiTS-rot, the test set of 180 degree rotated CT scans, all models fail to perform proper segmentation. The reliability of the predictions from all models is equally poor, which is to be expected, as calibration is negatively affected by incorrect segmentations. The models confidently segment the spleen and other regions in the patient's left upper abdominal quadrant. This indicates that the models rely on the location of the liver as a reliable source of information, which could consistently be used during training to differentiate it from other organs.

Furthermore, mean aleatoric uncertainty of BNN-MI and BNN-HLS is slightly decreased while epistemic uncertainty is considerably increased. The small decrease in aleatoric uncertainty cannot indicate less label noise in the images, since label noise in LiTS-rot is by construction equal to that in LiTS-test. Instead, it is a sign of the model's failed segmentation, as qualitative investigation reveals that the segmented regions are noticeably smaller and more numerous than the liver itself, leading to more object borders lined with aleatoric uncertainty. In this case, the shift in distribution causes the models' reliability to decrease considerably, rendering the decomposed uncertainty estimates less meaningful. Meanwhile, the general rise in epistemic uncertainty as compared to the uncertainty estimates reported for the normal LiTS-test is statistically significant for all models. The difference between the distributions for the per-case epistemic uncertainty estimates is also clearly visible in Figures 5.4 and 5.6. While these results demonstrate a strong correlation between OOD samples and epistemic uncertainty, the uncertainty estimates are not generally reliable OOD detectors, since a considerable number of LiTS-rot cases are actually assigned epistemic uncertainty estimates that lie in the distribution of the normal LiTS-test, as shown in Figures 5.4 and 5.6.

These findings are in line with work conducted by Ovadia et al. (2019), who showed that segmentation performance as well as calibration significantly degrade with distribution shifts. The reader is also referred to Graham et al. (2022), who emphasise the difference between uncertainty quantification and OOD detection, and instead propose to explicitly estimate the training data-likelihood of a given sample.

The strong ascites case arguably presents a less extreme shift in distribution, while still significantly deviating from the LiTS-full training data. The BNN-MI and BNN-HLS models perform more meaningful segmentation of the liver, while producing visibly fuzzy segmentation borders as well as mistaking large parts of the intraperitoneal fluid as liver. The decomposed uncertainties conform to expectations. Mean aleatoric uncertainty is slightly increased and mean epistemic uncertainty is considerably increased, exceeding that of models trained on only two training cases (LiTS-2). The slightly raised aleatoric estimates account for the increase in segmented class boundaries, while epistemic uncertainty successfully indicates the shift in distribution.

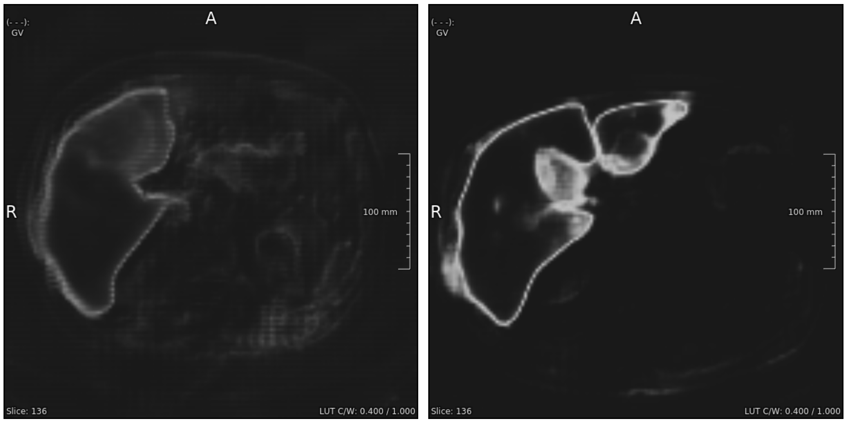


Figure 6.1: Aleatoric uncertainty as estimated via HLSN (L) and mean entropy (R) for a BNN-HLS model

6.3 Aleatoric Uncertainty Metric for Loss-attenuating Neurons

Two approaches to aleatoric uncertainty estimation afforded by a model equipped with heteroscedastic loss-attenuating neurons are introduced in Section 4.3.6: using the output of the heteroscedastic neuron and deriving an estimate from the model's predictive distribution. Recall that the same section also reported that all reproductions of Kendall and Gal (2017) differed from the original work by construing the HUN's output as aleatoric uncertainty estimate. It was decided to derive both aleatoric uncertainty estimates in order to be able to compare the resulting uncertainties in this work.

Let us briefly compare visually how the output of the heteroscedastic neuron varies from the entropy of the softmax distribution in terms of encoding aleatoric uncertainty in the setting of this work. Figure 6.1 depicts two uncertainty maps from the HLS model derived via the output of the HLSN and the entropy of the predictive distribution. The maps show a qualitative difference between the two aleatoric uncertainty estimates, with the entropy-based uncertainty being more pronounced around class boundaries and containing less noise around structures other than the segmented object, i. e. the entropy-based uncertainty is more robust towards noise. Similar qualitative differences are found for the HUN model, as shown in Appendix F.

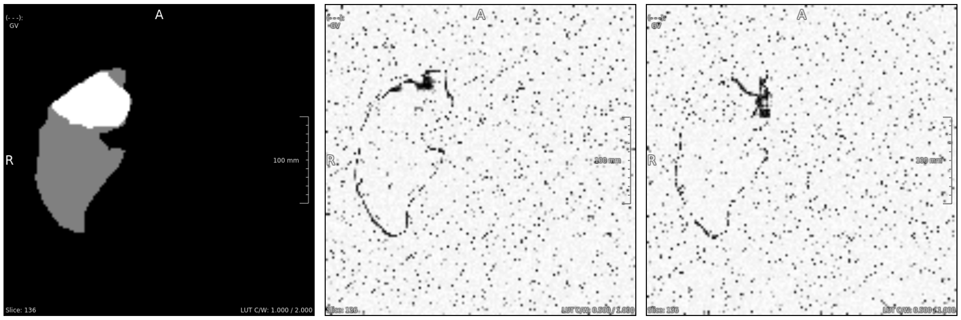


Figure 6.2: Ground truth liver mask (L), variance of logits for the pure-NN model (M) and the HLS model (R)

This observation is in line with the results by Nair et al. (2020), in which the authors found the HUN activation to highlight the borders of not only the segmented objects but also of other structures in the image. In addition to this qualitative observation, the quantitative evaluation of the behaviour of both aleatoric uncertainty metrics, as described in Sections 5.3.2 and 5.3.2, showed that the mean entropy of the softmax output distributions corresponds precisely to the intuition about aleatoric uncertainty while the activation of the HLSN is less cleanly interpretable.

Furthermore, the models' logits were found to inherently contain a significant amount of uncertainty which can only be captured when deriving aleatoric uncertainty from the predictive distribution, as explained in Section 4.3.6. Figure 6.2 depicts the variance of the logits of an HLS model alongside its corresponding ground-truth liver mask. The images demonstrate the informativeness of the logits' inherent uncertainty, since they clearly and consistently indicate less confidence in a thin border around the class boundary, which corresponds exactly to the intuition about aleatoric uncertainty. Thus, the logits clearly still encode some amount of aleatoric uncertainty, despite the added heteroscedastic loss-attenuating neuron. In fact, the logits output by the pure-NN model look similar in form and magnitude to the logits of models that employ HLS. This suggests that the calibration lever introduced by the HLSN is used *in addition to* and not *as a replacement for* the logit-inherent uncertainty. When interpreting the output of the HLSN as aleatoric uncertainty, this considerable amount of logit-inherent uncertainty is discarded. This finding, thus, corroborates the theoretical discussion in Section 4.3.6, and motivates the use of comprehensive aleatoric uncertainty metrics.

This also reinforces the argument in favour of the approaches by Kendall and Gal (2017) and Neumann et al. (2018), which afford straightforward uncertainty decomposition methods applied on the predictive distribution, because the loss-attenuating terms directly translate into a smoother output distribution at both training and test time.

6.4 Loss Interactions

Models of each uncertainty-decomposing architecture, BNN-MI and BNN-HLS, were also trained employing the soft Dice loss as objective function, as motivated in Section 4.3.2. The following investigation of the resulting uncertainty maps aims to briefly answer two questions. The first question pertains to the quality of decomposed uncertainty estimates of models trained with Dice loss, with the hypothesis being that aleatoric uncertainty will be reduced, because the individual softmax score distributions are promoted to be overconfident. The other question is, whether the HLSN is generally capable of capturing meaningful uncertainty, leading to a better-calibrated model than without loss attenuation, when trained with Dice loss.

Example uncertainty maps in Figure 6.3 juxtapose the aleatoric uncertainty estimates computed for both BNN-MI and BNN-HLS models when trained with cross-entropy and with soft Dice loss. For both architectures, the choice of loss function leads to a stark qualitative difference in the resulting uncertainty maps. Models trained with cross-entropy exhibit the expected aleatoric uncertainty at class boundaries, and show very faint epistemic uncertainty overall. For the models trained with soft Dice loss, on the other hand, it appears as though the roles of aleatoric and epistemic uncertainty were reversed. The epistemic uncertainty now prominently appears at class boundaries while aleatoric uncertainty is reduced to a partial and faint border around the segmentation.

The already observed miscalibration of the models trained with soft Dice loss, shown in Table 5.7, aligns with the well-known fact that soft Dice loss promotes overconfident models. The changes observed for the individual uncertainty type estimates, on the other hand, have to the best of the authors knowledge not have been studied before. The apparent role-reversal of epistemic and aleatoric uncertainty can be explained by understanding aleatoric uncertainty as the mean entropy of the single MC predictions while epistemic uncertainty is represented by the variance over multiple MC predictions, as described in Section 4.3.6. The mutual information decomposition corresponds precisely to this interpretation. Soft Dice loss, simultaneously, pushes the network to be confident in its individual predictions, thereby suppressing aleatoric uncertainty estimates. These results confirm the initial hypothesis regarding the quality of uncertainty decomposition. The soft Dice loss not only leads to worse reliability, but it also drastically reduces the amount of aleatoric uncertainty obtained via the mean entropy of the predictive distribution.

The miscalibration exhibited in the reliability diagrams in Figure 5.2 also answers the other question, demonstrating that models trained with loss-attenuating neurons also suffer from overconfidence when trained with the soft Dice loss. Figure 6.3 includes an activation map of the HLSN, in order to more closely investigate the role of the heteroscedastic neuron. The map reveals that effectively no uncertainty is captured by the additional heteroscedastic neuron. With the logit-smoothing neuron rendered unusable by the soft Dice loss, the BNN-HLS model therefore calibrates its output scores solely via the logit values.

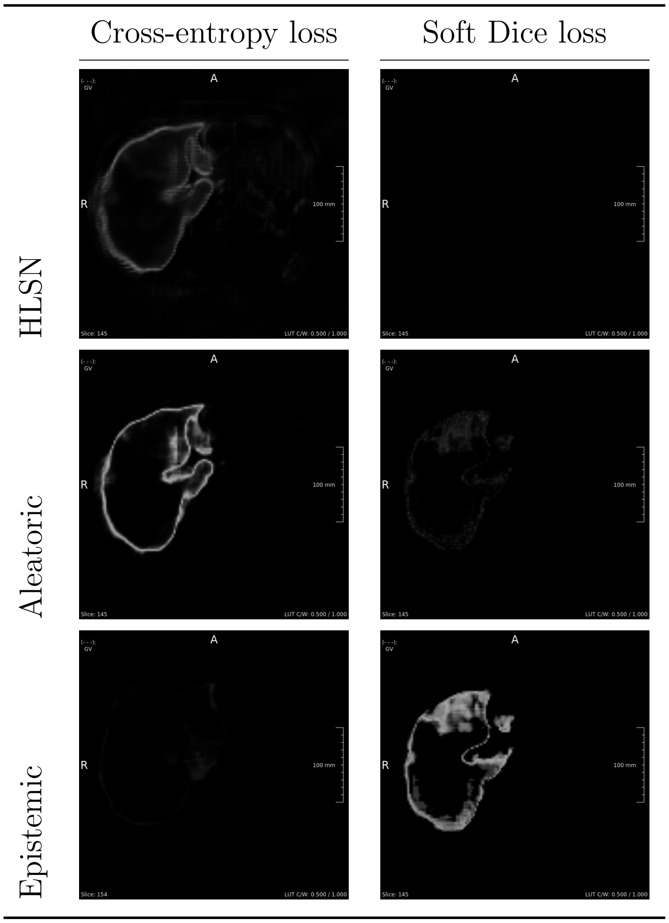


Figure 6.3: Uncertainty maps for BNN-HLS models: one is trained with cross-entropy loss and one is trained with soft Dice loss.

Conclusion

This work provides a comprehensive comparison between the direct decomposition of the predictive uncertainty of a BNN via the MI metric and the explicitly separate modelling of epistemic and aleatoric uncertainty in a BNN with an additional heteroscedastic loss-attenuating neuron. BNNs in this work are implemented via MC dropout variational inference and use a heteroscedastic logit smoothing neuron as loss attenuator. The underlying mechanism of logit smoothing is shown to be similar to that of the heteroscedastic uncertainty neuron which was originally introduced for use in the joint architecture by Kendall and Gal.

The comparison performed in the context of medical image segmentation of the liver from CT scans, employing a 3D au-net as base architecture.

In order to evaluate the quality of the uncertainty decomposition, the resulting uncertainty maps were evaluated qualitatively and the mean uncertainty values per test case were systematically compared quantitatively for different experiment settings with varying training set sizes, label noise, and distribution shifts.

The comparison revealed that the output of the added heteroscedastic neuron, while correctly indicating aleatoric uncertainty at the class boundaries, contains a lot of noise around other structures and is generally less robust in converging towards capturing uncertainty during training than aleatoric uncertainty estimates derived via the MI decomposition. Meanwhile, deriving aleatoric uncertainty via the mutual information decomposition for both models produces the desired label noise-capturing estimates and cleaner uncertainty maps. Adding artificial noise by dilating half of the ground truth masks in the training data leads to significantly higher mean aleatoric uncertainty per case and visibly broader aleatoric uncertainty estimates at class boundaries for both model architectures. This demonstrates the robust ability of the mutual information metric to derive meaningful aleatoric uncertainty from predictive distributions.

Epistemic uncertainty as computed via the mutual information metric is overall lower in magnitude than aleatoric uncertainty for the employed data set, which reflects the high number of training samples and the absence of rare classes in the binary liver segmentation task. The epistemic estimates frequently occur around the outline of segmented objects, which corresponds to expected label noise. Smaller training sets significantly increase epistemic uncertainty and diffusely highlight structures that are visually similar to the liver, such as the spleen. The overall behaviour is consistent with the definition of epistemic uncertainty and the results of previous works deriving epistemic uncertainty in image segmentation.

Segmentation performance and reliability of the models is very high for ID data and extremely poor for OOD data. This work found no statistically significant difference in the performance, reliability, or quality of uncertainty decomposition between the BNN and the joint architecture combining a BNN with heteroscedastic logit smoothing.

Adding a heteroscedastic logit smoothing neuron (HLSN) to a BNN does indeed result in the neuron learning uncertainty, as seen in its activation, however this work found it to not statistically significantly improve the quality of the uncertainty estimates when decomposed via the MI metric. Noisiness in the activation of the loss-attenuating neuron in both the uncertainty maps as well as in its quantitative properties across different settings leads to the conclusion that the MI decomposition remains significantly more suited for uncertainty decomposition of BNNs even with loss-attenuating neurons.

This work also demonstrates a strong influence of the choice of loss function on the quality of uncertainty decomposition. The soft Dice loss essentially disables loss-attenuating neurons and heavily deteriorates the quality of the decomposed uncertainties. In particular, the roles of aleatoric and epistemic uncertainty show reversed behaviour, a finding which might help future practitioners choose an adequate loss function when aiming to decompose their models' uncertainty.

Limitations and Future Work

A strong limitation of this work is the task setting itself. Binary tasks inherently do not induce much epistemic uncertainty and the overall performance and reliability were extremely high even for the vanilla neural network models. The very high baseline performance renders performance improvements harder to achieve and quantify. Conducting this comparison exhaustively on more diverse and challenging multi-class data sets would therefore yield a more comprehensive understanding of the differences between the uncertainty decomposition in BNNs with and without loss-attenuating neurons.

Models with additional loss-attenuating neurons, such as heteroscedastic uncertainty neurons (HUNs) and heteroscedastic logit smoothing neurons (HLSNs), were found to have non-trivial convergence properties regarding the neurons' tendency to capture uncertainty. Successful loss attenuation required carefully tuning the initialisation of their incoming weights. Finding a stable and more generally applicable training regime reliably inducing convergence would be a crucial prerequisite for widespread adoption of heteroscedastic loss-attenuating neurons.

Though MC Dropout provides a cheap and straightforward approximation to BNNs, one could compare the quality of decomposed uncertainty estimates for additional approximation approaches. Especially interesting are methods supporting more complex posteriors and their interaction with estimates derived from the predictive distribution. Different approximation approaches might also have less influence on the convergence of the HUN.

The evaluation of hyper-parameters, such as dropout rate and the position of the dropout layers, were not in the scope of this work, with their values set to sane defaults adopted from prior work or determined by preliminary experiments. Future works should investigate the influence of these parameters on the quality of uncertainty estimates derived from a BNN's predictive distribution and their interaction with loss-attenuating neurons.

It would be interesting to investigate if one can fine-tune the calibration of an already trained (B)NN by adding a heteroscedastic output neuron. This could essentially constitute a novel post-hoc calibration method with straightforward implementation and no requirements to otherwise alter the network architecture or loss function.

Another interesting question regards the differences between uncertainty estimates yielded by 2D and 3D models. Given that the uncertainty estimates are comparable one could envision a setup, where active learning is performed with cheap 2D models, producing a training set for a much more computationally expensive high performance model on 3D data.

Bleeding-edge models in image processing make increasing use of transformer-based architectures operating via attention mechanisms (Vaswani et al., 2017). Investigating the calibration properties of attention-based models and how to best estimate their uncertainty would be an interesting next step. Fan et al. (2020) propose a first approach for quantifying uncertainty of attention-based models from a Bayesian viewpoint.

Bibliography

Abadi, Martín et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Software available from tensorflow.org.

Abdar, Moloud, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi (2021). "A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges". In: *Information Fusion* 76.C, 243–297. DOI: `10.1016/j.inffus.2021.05.008`.

Alabousi, Mostafa, Matthew DF McInnes, Jean-Paul Salameh, Janakan Satkunasingham, Yoan K. Kagoma, Leyo Ruo, Brandon M. Meyers, Tariq Aziz, and Christian B. van der Pol (2021). "MRI vs. CT for the Detection of Liver Metastases in Patients With Pancreatic Carcinoma: A Comparative Diagnostic Test Accuracy Systematic Review and Meta-Analysis". In: *Journal of Magnetic Resonance Imaging* 53.1, pp. 38–48. DOI: `https://doi.org/10.1002/jmri.27056`.

Bertels, Jeroen, David Robben, Dirk Vandermeulen, and Paul Suetens (Jan. 2021). "Theoretical analysis and experimental validation of volume bias of soft Dice optimized segmentation maps in the context of inherent uncertainty". In: *Medical Image Analysis* 67, p. 101833. DOI: `10.1016/j.media.2020.101833`.

Betts, J Gordon, P Desaix, E Johnson, JE Johnson, O Korol, D Kruse, B Poe, JA Wise, Mark Womble, and KA Young (2013). "Anatomy and physiology". In: *Rice University: Houston, TX, USA.*

Bilic, Patrick et al. (Jan. 2019). *The Liver Tumor Segmentation Benchmark (LiTS).* Tech. rep.

Blundell, Charles, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra (2015). "Weight Uncertainty in Neural Networks". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37.* ICML'15. Lille, France: JMLR.org, 1613–1622.

Carter, Shan, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah (Mar. 2019). "Activation Atlas". In: *Distill* 4.3, 10.23915/distill.00015.

Chlebus, Grzegorz, Andrea Schenk, Horst K. Hahn, Bram Van Ginneken, and Hans Meine (2022). "Robust Segmentation Models Using an Uncertainty Slice Sampling-Based Annotation Workflow". In: *IEEE Access* 10, pp. 4728–4738. DOI: `10.1109/ACCESS.2022.3141021`.

Chollet, François et al. (2015). *Keras*. `https://keras.io`.

Chougula, Basavaraj, Arun Tigadi, Prabhakar Manage, and Sadanand Kulkarni (2020). "Road segmentation for autonomous vehicle: A review". In: *2020 3rd International Conference on Intelligent Sustainable Systems*, pp. 362–365. DOI: `10.1109/ICISS49785.2020.9316090`.

DeGroot, Morris H. and Stephen E. Fienberg (1983). "The Comparison and Evaluation of Forecasters". In: *Journal of the Royal Statistical Society. Series D (The Statistician)* 32.1/2, pp. 12–22. DOI: `10.2307/2987588`.

Depeweg, Stefan, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft (2018). "Decomposition of Uncertainty in Bayesian Deep Learning for Efficient and Risk-sensitive Learning". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 1184–1193.

DeVries, Terrance and Graham W. Taylor (2018a). "Learning Confidence for Out-of-Distribution Detection in Neural Networks". In: *CoRR* abs/1802.04865.

— (2018b). "Leveraging Uncertainty Estimates for Predicting Segmentation Quality". In: *CoRR* abs/1807.00502.

Dice, Lee R. (July 1945). "Measures of the Amount of Ecologic Association Between Species". In: *Ecology* 26.3, pp. 297–302. DOI: `10.2307/1932409`.

Ding, Zhipeng, Xu Han, Peirong Liu, and Marc Niethammer (Oct. 2021). "Local Temperature Scaling for Probability Calibration". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, pp. 6869–6879. DOI: `10.1109/ICCV48922.2021.00681`.

Eisenhauer, E A et al. (Jan. 2009). "New response evaluation criteria in solid tumours: revised RECIST guideline (version 1.1)". In: *Eur J Cancer* 45.2, pp. 228–247.

Fan, Xinjie, Shujian Zhang, Bo Chen, and Mingyuan Zhou (2020). "Bayesian Attention Modules". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 16362–16376.

Gal, Yarin and Zoubin Ghahramani (2016). "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pp. 1050–1059.

Gamerman, Dani. (1997). *Markov chain Monte Carlo : stochastic simulation for Bayesian inference*. London: Chapman & Hall.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (June 2011). "Deep Sparse Rectifier Neural Networks". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. ISSN: 1938-7228. JMLR Workshop and Conference Proceedings, pp. 315–323.

Goodfellow, Ian, Jonathon Shlens, and Christian Szegedy (2015). "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations*.

Graham, Mark S., Petru-Daniel Tudosiu, Paul Wright, Walter Hugo Lopez Pinaya, James Teo, Jean-Marie U-King-Im, Yee Mah, Rolf H. Jäger, David Werring, Parashkev Nachev, Sebastien Ourselin, and M. Jorge Cardoso (2022). "Transformer-based out-of-distribution detection for clinically safe segmentation". In: *International Conference on Medical Imaging with Deep Learning, MIDL 2022, 6-8 July 2022, Zürich, Switzerland*. Proceedings of Machine Learning Research. Forthcoming.

Guo, Chuan, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger (Aug. 2017). "On calibration of modern neural networks". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. Sydney, NSW, Australia: JMLR.org, pp. 1321–1330.

Hann, L. E., C. B. Winston, K. T. Brown, and T. Akhurst (2000). "Diagnostic imaging approaches and relationship to hepatobiliary cancer staging and therapy." In: *Seminars in surgical oncology* 19.2, pp. 94–115.

Hendrycks, Dan and Kevin Gimpel (2016). "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *CoRR* abs/1610.02136.

Houlsby, Neil, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel (2011). "Bayesian Active Learning for Classification and Preference Learning". In: *CoRR* abs/1112.5745.

Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger (July 2017). "Densely Connected Convolutional Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. ISSN: 1063-6919, pp. 2261–2269. DOI: 10.1109/CVPR.2017.243.

Huang, Rui, Andrew Geng, and Yixuan Li (2021). "On the Importance of Gradients for Detecting Distributional Shifts in the Wild". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., pp. 677–689.

Hüllermeier, Eyke and Willem Waegeman (Mar. 2021). "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods". In: *Machine Learning* 110.3, pp. 457–506. DOI: 10.1007/s10994-021-05946-3.

Iantsen, Andrei, Dimitris Visvikis, and Mathieu Hatt (2021). "Squeeze-and-Excitation Normalization for Automated Delineation of Head and Neck Primary Tumors in Combined PET and CT Images". In: *Head and Neck Tumor Segmentation*. Ed. by Vincent Andrearczyk, Valentin Oreiller, and Adrien Depeursinge. Cham: Springer International Publishing, pp. 37–43.

Jaccard, Paul (Feb. 1912). "The Distribution of the Flora in the Alpine Zone". In: *New Phytologist* 11.2, pp. 37–50. DOI: 10.1111/j.1469-8137.1912.tb05611.x.

Johnson, Justin M. and Taghi M. Khoshgoftaar (Mar. 2019). "Survey on deep learning with class imbalance". In: *Journal of Big Data* 6.1, p. 27. DOI: 10.1186/s40537-019-0192-5.

Kendall, Alex, Vijay Badrinarayanan, and Roberto Cipolla (Jan. 2017). "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding". In: DOI: 10.5244/C.31.57.

Kendall, Alex and Yarin Gal (2017). "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc.

Kingma, Diederik P. and Max Welling (2014). "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun.

Kock, Farina, Felix Thielke, Grzegorz Chlebus, and Hans Meine (2022). "Confidence Histograms for Model Reliability Analysis and Temperature Calibration". In: *International Conference on Medical Imaging with Deep Learning, MIDL 2022, 6-8 July 2022, Zürich, Switzerland*. Proceedings of Machine Learning Research. Forthcoming.

Kull, Meelis and Peter Flach (2015). "Novel Decompositions of Proper Scoring Rules for Classification: Score Adjustment as Precursor to Calibration". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Annalisa Appice, Pedro Pereira Rodrigues, Vítor Santos Costa, Carlos Soares, João Gama, and Alípio Jorge. Vol. 9284. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 68–85. DOI: 10.1007/978-3-319-23528-8_5.

Kwon, Yongchan, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik (2018). "Uncertainty quantification using Bayesian neural networks in classification: Application to ischemic stroke lesion segmentation". In: p. 13. DOI: https://doi.org/10.1016/j.csda.2019.106816.

— (2020). "Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation". In: *Computational Statistics & Data Analysis* 142, p. 106816. DOI: https://doi.org/10.1016/j.csda.2019.106816.

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell (2017). "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc.

Leibig, Christian, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl (Dec. 2017). "Leveraging uncertainty information from deep neural networks for disease detection". In: *Scientific Reports* 7.1, p. 17816. DOI: 10.1038/s41598-017-17876-z.

Liang, Shiyu, Yixuan Li, and R. Srikant (2017). "Principled Detection of Out-of-Distribution Examples in Neural Networks". In: *CoRR* abs/1706.02690.

Lin, Eugene and Adam Alessio (Nov. 2009). "What are the basic concepts of temporal, contrast, and spatial resolution in cardiac CT?" In: *J. Cardiovasc. Comput. Tomogr.* 3.6, pp. 403–408.

Ma, Jun (2021). "Cutting-edge 3D Medical Image Segmentation Methods in 2020: Are Happy Families All Alike?" In: *arXiv preprint arXiv:2101.00232*.

Maier, Andreas, Stefan Steidl, Vincent Christlein, and Joachim Hornegger, eds. (2018). *Medical Imaging Systems - An Introductory Guide*. Springer International Publishing. DOI: 10.1007/978-3-319-96520-8.

Manisha, Pallath, Rabindranath Jayadevan, and Vayakkattil Sidharthan Sheeba (2020). "Content-based image retrieval through semantic image segmentation". In: *AIP Conference Proceedings* 2222.1, p. 030008. DOI: 10.1063/5.0004087.

McKinley, Richard, Michael Rebsamen, Raphael Meier, Mauricio Reyes, Christian Rummel, and Roland Wiest (2019). "Few-shot brain segmentation from weakly labeled data with deep heteroscedastic multi-task networks". In: *CoRR* abs/1904.02436.

Mehrtash, Alireza, William M. Wells III, Clare M. Tempany, Purang Abolmaesumi, and Tina Kapur (Dec. 2020). "Confidence Calibration and Predictive Uncertainty Estimation for Deep Medical Image Segmentation".

In: *IEEE Transactions on Medical Imaging* 39.12, pp. 3868–3878. DOI: 10.1109/TMI.2020.3006437.

Mihail, Radu Paul, Gongbo Liang, and Nathan Jacobs (July 2019). “Automatic Hand Skeletal Shape Estimation From Radiographs”. In: *IEEE Transactions on NanoBioscience* 18.3, pp. 296–305. DOI: 10.1109/TNB.2019.2911026.

Minderer, Matthias, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic (2021). “Revisiting the Calibration of Modern Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., pp. 15682–15694.

Mobiny, Aryan, Pengyu Yuan, Supratik K. Moulik, Naveen Garg, Carol C. Wu, and Hien Van Nguyen (Dec. 2021). “DropConnect is effective in modeling uncertainty of Bayesian deep networks”. In: *Scientific Reports* 11.1, p. 5458. DOI: 10.1038/s41598-021-84854-x.

Monteith, Kristine and Tony Martinez (July 2010). “Using multiple measures to predict confidence in instance classification”. In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*. ISSN: 2161-4407, pp. 1–8. DOI: 10.1109/IJCNN.2010.5596550.

Naeini, Mahdi Pakdaman, Gregory F. Cooper, and Milos Hauskrecht (2015). “Obtaining Well Calibrated Probabilities Using Bayesian Binning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence* 2015, pp. 2901–2907.

Nair, Tanya, Doina Precup, Douglas L. Arnold, and Tal Arbel (Jan. 2020). “Exploring uncertainty measures in deep networks for Multiple sclerosis lesion detection and segmentation”. In: *Medical Image Analysis* 59, p. 101557. DOI: 10.1016/j.media.2019.101557.

Neumann, Lukas, Andrew Zisserman, and Andrea Vedaldi (2018). “Relaxed Softmax: Efficient Confidence Auto-Calibration for Safe Pedestrian Detection”. In: *Advances in Neural Information Processing Systems*, p. 8.

Ng, Matthew, Fumin Guo, Labonny Biswas, Steffen E. Petersen, Stefan K. Piechnik, Stefan Neubauer, and Graham A. Wright (2020). “Estimating Uncertainty in Neural Networks for Cardiac MRI Segmentation: A Benchmark Study”. In: *CoRR* abs/2012.15772.

Nguyen, Vu-Linh, Sebastien Destercke, and Eyke Hüllermeier (Oct. 2019). “Epistemic Uncertainty Sampling”. In: pp. 72–86. DOI: 10.1007/978-3-030-33778-0_7.

Ovadia, Yaniv, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek (2019). "Can you trust your model' s uncertainty? Evaluating predictive uncertainty under dataset shift". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.

O'Neill, Erin K., Jonathan R. Cogley, and Frank H. Miller (Feb. 2015). "The Ins and Outs of Liver Imaging". In: *Clinics in Liver Disease* 19.1, pp. 99–121. DOI: 10.1016/j.cld.2014.09.006.

Papamarkou, Theodore, Jacob Hinkle, M. Todd Young, and David Womble (2022). "Challenges in Markov Chain Monte Carlo for Bayesian Neural Networks". In: *Statistical Science* 37.3, pp. 425 –442. DOI: 10.1214/21-STS840.

Platt, John C. (1999). "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods". In: *Advances in Large Margin Classifiers*. MIT Press, pp. 61–74.

Qin, Lihui and James M. Crawford (2018). "1 - Anatomy and Cellular Functions of the Liver". In: *Zakim and Boyer's Hepatology (Seventh Edition)*. Ed. by Arun J. Sanyal, Thomas D. Boyer, Keith D. Lindor, and Norah A. Terrault. Seventh Edition. Philadelphia: Elsevier, 2–19.e4. DOI: https://doi.org/10.1016/B978-0-323-37591-7.00001-X.

Ritter, Felix, Tobias Boskamp, André Homeyer, Hendrik Laue, Michael Schwier, Florian Link, and Heinz-Otto Peitgen (Nov. 2011). "Medical image analysis". In: *IEEE Pulse* 2.6, pp. 60–70.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

Sander, Jörg, Bob D. de Vos, Jelmer M. Wolterink, and Ivana Išgum (Mar. 2019). "Towards increased trustworthiness of deep learning segmentation methods on cardiac MRI". In: *Medical Imaging 2019: Image Processing*, p. 44. DOI: 10.1117/12.2511699.

Sanyal, Arun J., Thomas D. Boyer, Keith D. Lindor, and Norah A. Terrault, eds. (2018). *Zakim and Boyer's Hepatology*. Elsevier Inc. DOI: 10.1016/C2013-0-19055-1.

Senge, Robin, Stefan Bösner, Krzysztof Dembczyński, Jörg Haasenritter, Oliver Hirsch, Norbert Donner-Banzhoff, and Eyke Hüllermeier (2014).

"Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty". In: *Information Sciences* 255, pp. 16–29. DOI: https://doi.org/10.1016/j.ins.2013.07.030.

Sutton, Reed T, David Pincock, Daniel C Baumgart, Daniel C Sadowski, Richard N Fedorak, and Karen I Kroeker (Feb. 2020). "An overview of clinical decision support systems: benefits, risks, and strategies for success". In: *npj Digital Medicine* 3.1, p. 17.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna (June 2016). "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. ISSN: 1063-6919, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus (2014). "Intriguing properties of neural networks". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun.

Sørensen, Thorvals (1948). *A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons.*

Thulasidasan, Sunil, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof (May 2019). "Combating Label Noise in Deep Learning using Abstention". In: *Proceedings of the 36th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, pp. 6234–6243.

Timmeren, Janita E van, Davide Cester, Stephanie Tanadini-Lang, Hatem Alkadhi, and Bettina Baessler (Aug. 2020). "Radiomics in medical imaging—"how-to" guide and critical reflection". In: *Insights into Imaging* 11.1, p. 91.

Vander Kooi, Douglas C., Bobby T. Kalb, and Diego R. Martin (2018). "10 - Imaging in Assessment of Liver Disease and Lesions". In: *Zakim and Boyer's Hepatology (Seventh Edition)*. Ed. by Arun J. Sanyal, Thomas D. Boyer, Keith D. Lindor, and Norah A. Terrault. Seventh Edition. Philadelphia: Elsevier, 136–156.e2. DOI: https://doi.org/10.1016/B978-0-323-37591-7.00010-0.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems.* Vol. 30. Curran Associates, Inc.

Williams, G. and S. Renals (1997). "Confidence measures for hybrid HMM / ANN speech recognition". In: *EUROSPEECH.*

Wilson, Andrew G and Pavel Izmailov (2020). "Bayesian Deep Learning and a Probabilistic Perspective of Generalization". In: *Advances in Neural Information Processing Systems.* Vol. 33. Curran Associates, Inc., pp. 4697–4708.

Yang, Xiao, Roland Kwitt, and Marc Niethammer (2016). "Fast Predictive Image Registration". In: *Deep Learning and Data Labeling for Medical Applications.* Ed. by Gustavo Carneiro, Diana Mateus, Loïc Peter, Andrew Bradley, João Manuel R. S. Tavares, Vasileios Belagiannis, João Paulo Papa, Jacinto C. Nascimento, Marco Loog, Zhi Lu, Jaime S. Cardoso, and Julien Cornebise. Cham: Springer International Publishing, pp. 48–57.

Zadrozny, Bianca and Charles Elkan (Aug. 2002). "Transforming Classifier Scores into Accurate Multiclass Probability Estimates". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* DOI: 10.1145/775047.775151.

Zeiler, Matthew D. and Rob Fergus (2014). "Visualizing and Understanding Convolutional Networks". In: *Computer Vision – ECCV 2014.* Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Vol. 8689. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 818–833. DOI: 10.1007/978-3-319-10590-1_53.

Appendices

A Data Preprocessing

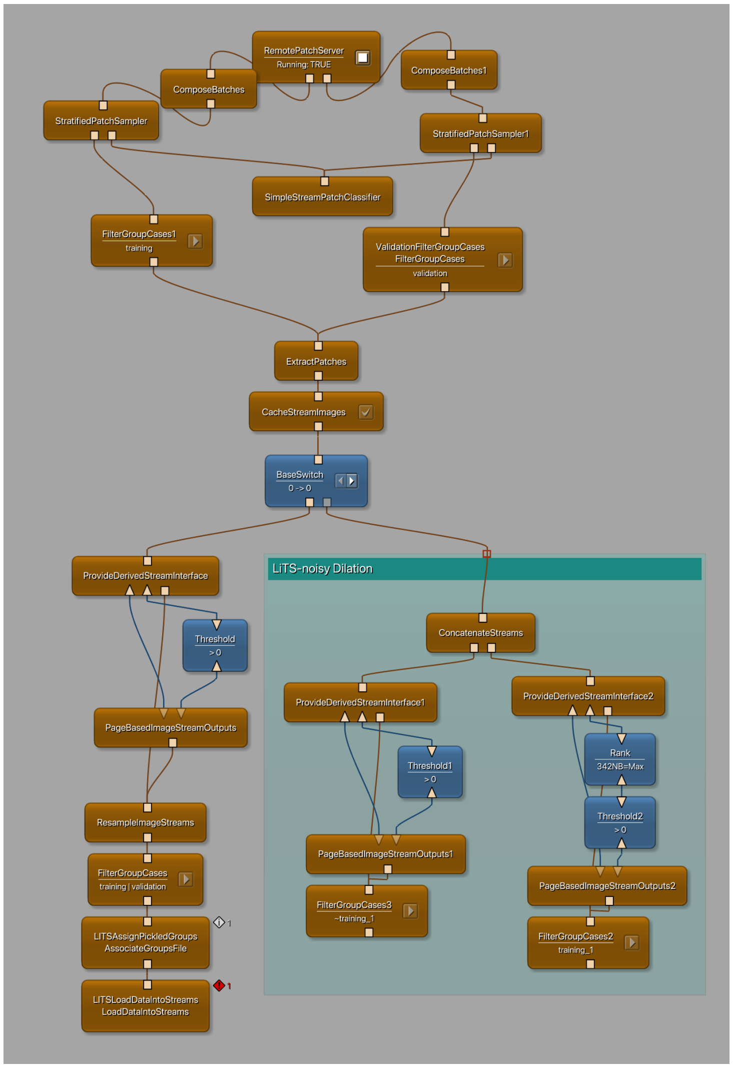


Figure 1: Data preprocessing MeVisLab network

B Mutual Information Computation

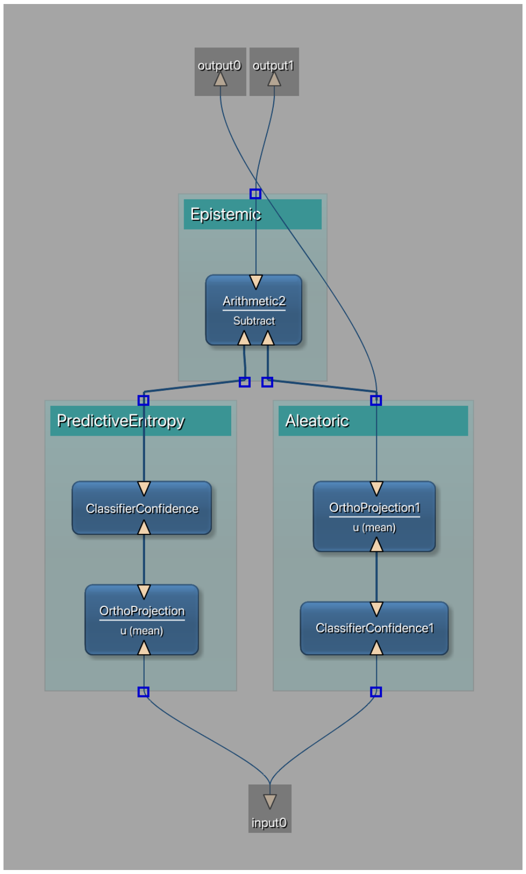


Figure 2: MeVisLab Macro Module for computing decomposed uncertainty estimates via the mutual information metric given a predictive distribution as input.

C Shapiro-Wilk Tests

Table 1: Shapiro-Wilk test results for the mean epistemic uncertainty per case as computed by BNN-MI and BNN-HLS on the test set.

	BNN-MI		BNN-HLS	
	W	p	W	p
LiTS	0.84	2.2e-4	0.78	1.5e-5
LiTS-noisy	0.86	7.0e-4	0.87	9.0e-4
LiTS-6	0.92	2.2e-2	0.87	1.4e-3
LiTS-2	0.85	4.1e-4	0.83	1.4e-4
LiTS-rot	0.67	2.3e-7	0.77	7.7e-6

Table 2: Shapiro-Wilk test results for the mean aleatoric uncertainty per case as computed by BNN-MI and BNN-HLS via MI decomposition and activation of the HLSN on LiTS-test.

	BNN-MI (MI)		BNN-HLS (MI)		BNN-HLS (HLSN)	
	W	p	W	p	W	p
LiTS	0.85	3.8e-4	0.83	1.2e-4	0.95	0.16
LiTS-noisy	0.85	2.8e-4	0.84	1.7e-4	0.83	1.1e-4
LiTS-6	0.84	2.6e-4	0.90	4.9e-3	0.94	0.16
LiTS-2	0.86	7.0e-4	0.87	1.0e-3	0.98	0.64
LiTS-rot	0.78	1.2e-5	0.79	1.7e-5	0.95	0.18

D Test Performance

Table 3: Dice coefficients for all three pure-NN, HUN- and HLS-models trained with cross-entropy loss on LiTS-full and inferring on LiTS-test.

	Pure-NN			HUN-model			HLS-model		
	I	II	III	I	II	III	I	II	III
LiTS-full	0.944	0.932	0.949	0.958	0.938	0.953	0.946	0.936	0.995

Table 4: Dice coefficients for all three BNN-MI and BNN-HLS models trained with cross-entropy loss for different training and test sets, as indicated in the first row.

		LiTS-full	LiTS-rot	LiTS6	LiTS2	LiTS-noisy
BNN-MI	I	0.919	0.013	0.894	0.736	0.532
	II	0.945	0.054	0.893	0.826	0.869
	III	0.952	0.056	0.802	0.785	0.831
BNN-HLS	I	0.946	0.066	0.876	0.716	0.828
	II	0.947	0.027	0.891	0.784	0.818
	III	0.947	0.119	0.826	0.744	0.858

E Reliability Diagrams

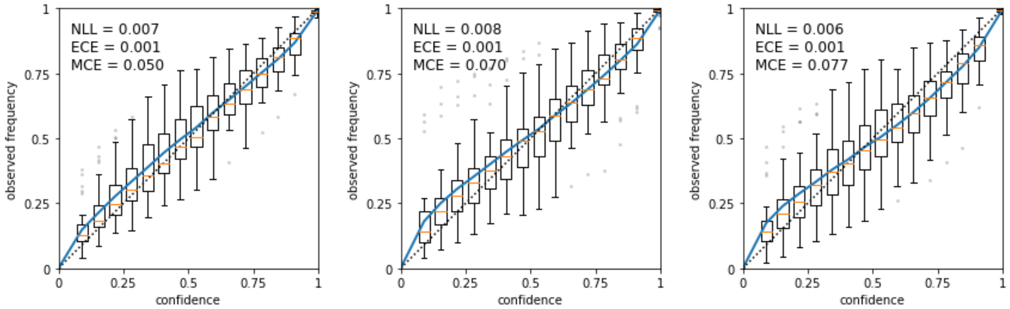


Figure 3: Reliability diagrams for all three pure-NN models trained with cross-entropy loss on LiTS-full

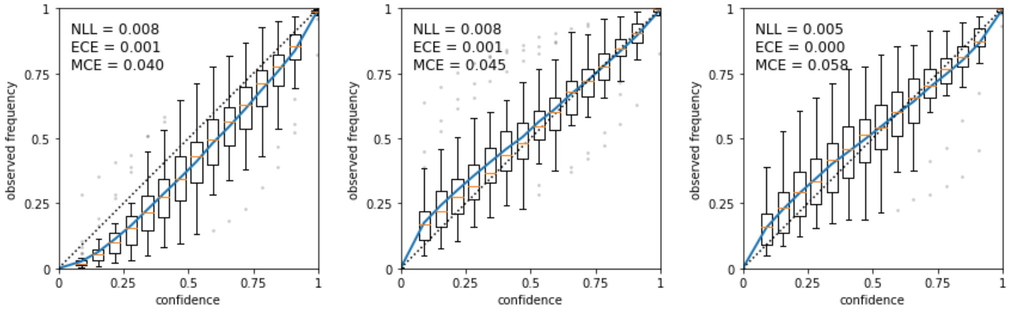


Figure 4: Reliability diagrams for all three HUN models trained with cross-entropy loss on LiTS-full

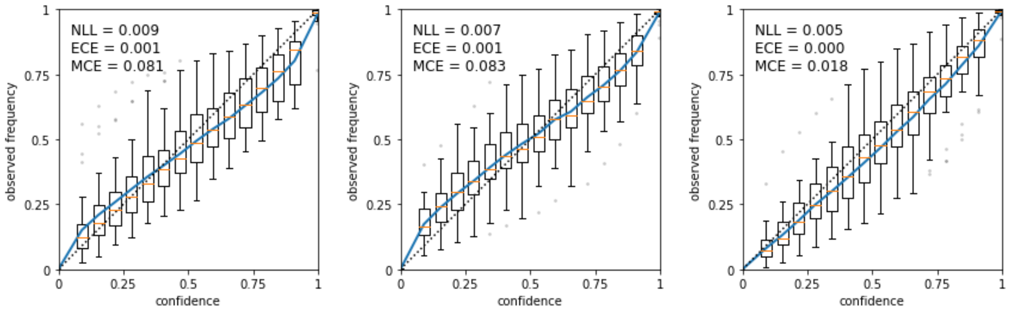


Figure 5: Reliability diagrams for all three HLS models trained with cross-entropy loss on LiTS-full

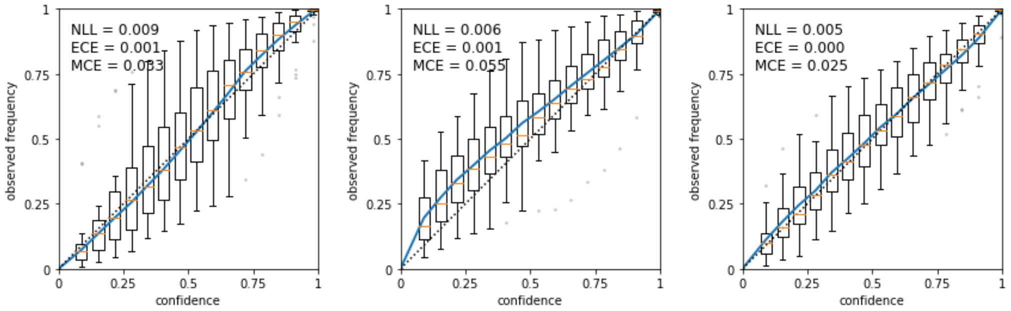


Figure 6: Reliability diagrams for all three BNN-MI models trained with cross-entropy loss on LiTS-full

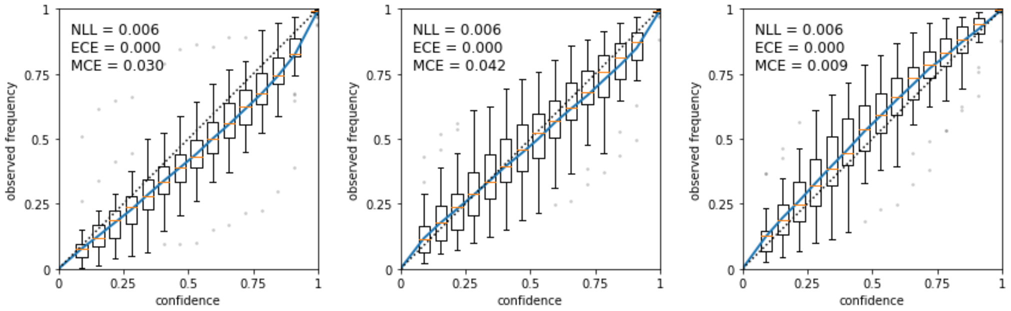


Figure 7: Reliability diagrams for all three BNN-HLS models trained with cross-entropy loss on LiTS-full

F Predictive Uncertainty Maps

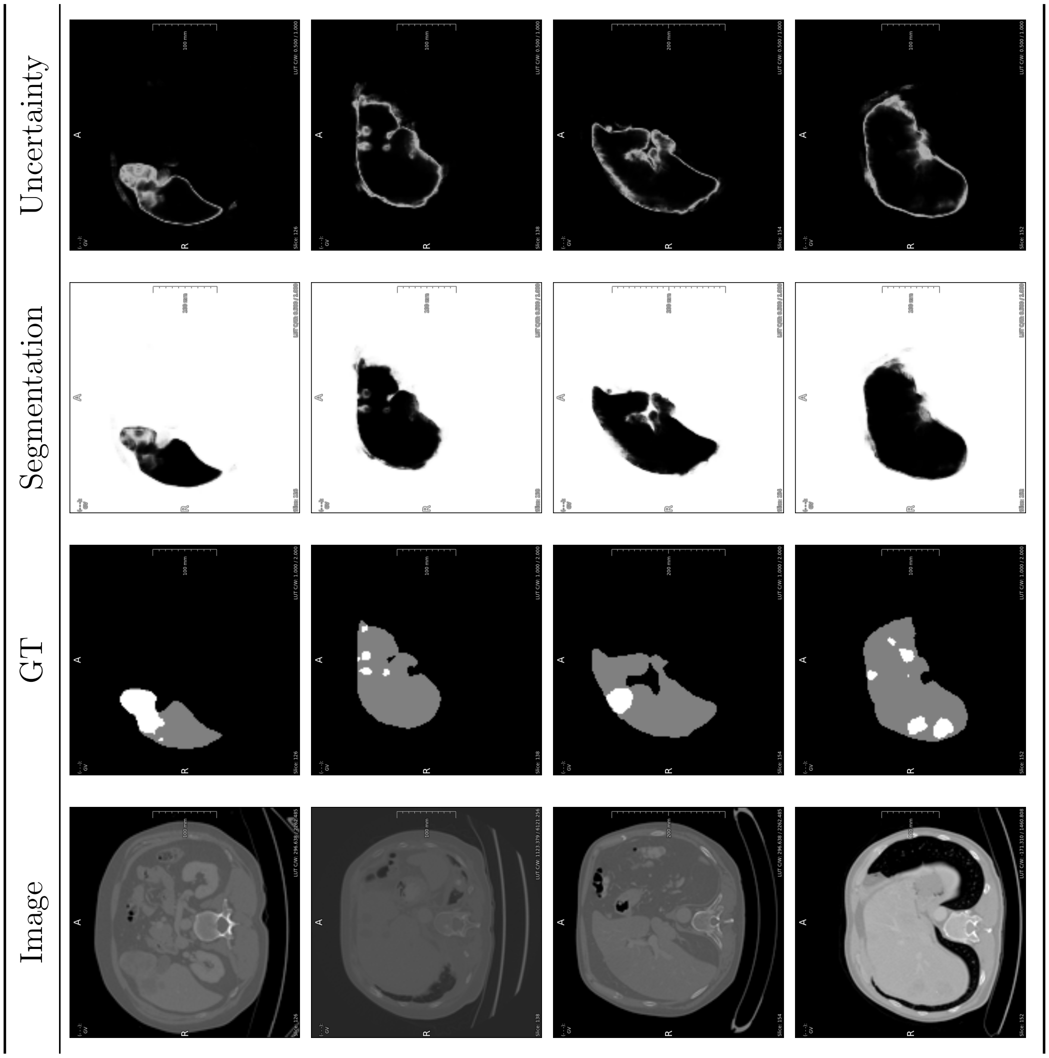


Figure 8: Example output for a pure-NN model trained on LiTS-full, uncertainty is computed via entropy of the softmax distribution.

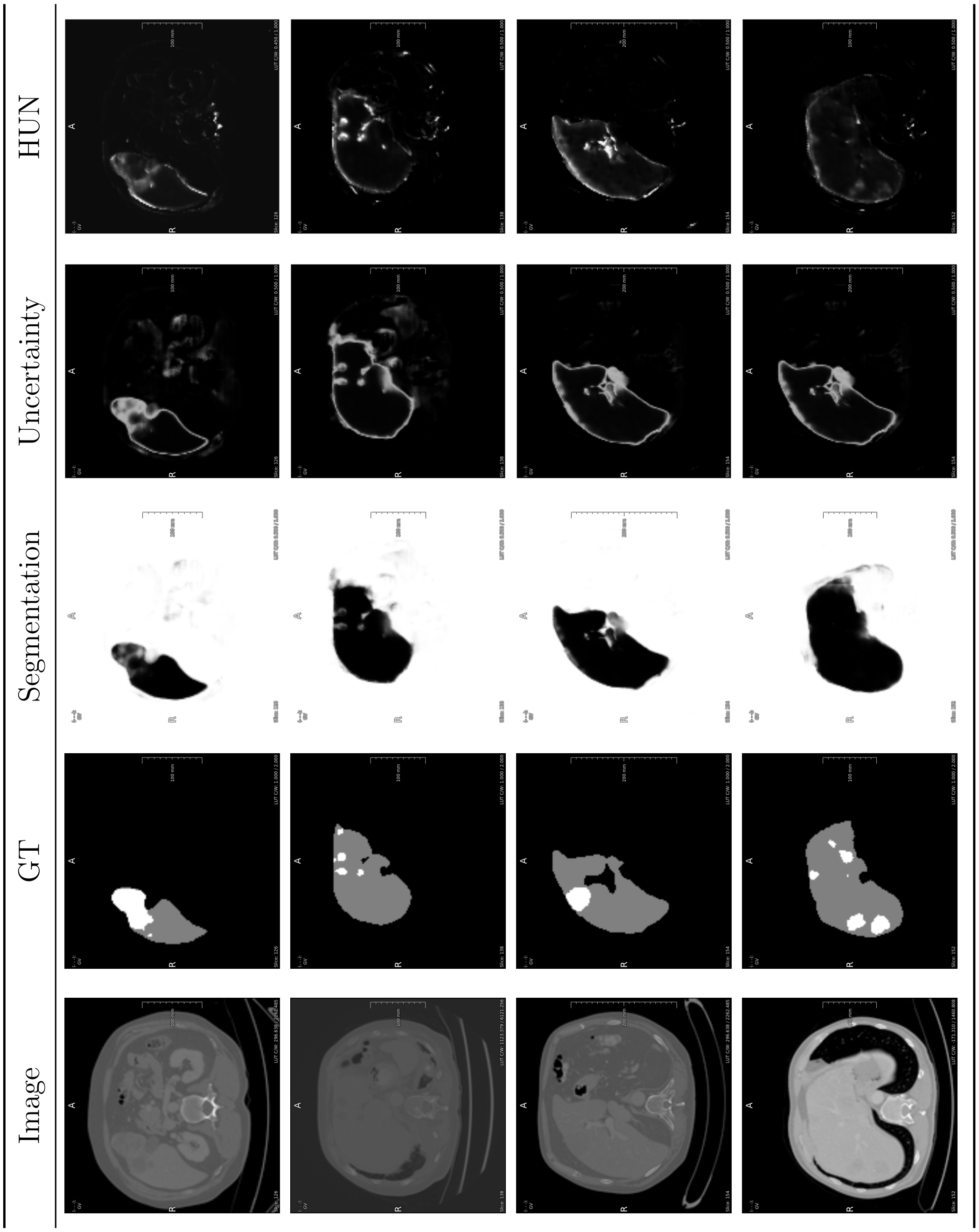


Figure 9: Example output for an HUN model trained on LiTS-full, uncertainty is computed via entropy of the softmax distribution and the activation of the HUN.

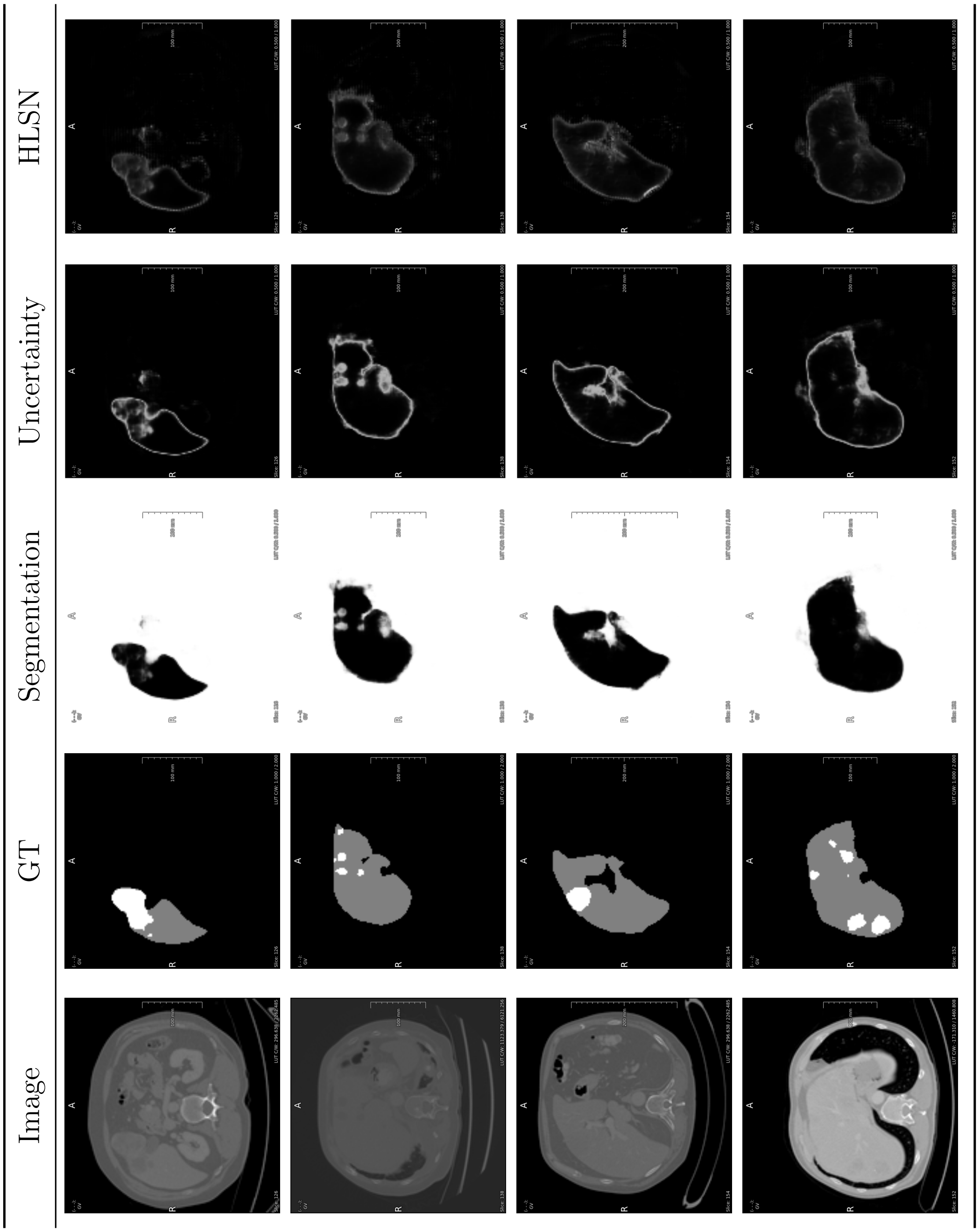


Figure 10: Example output for an HLS model trained on LiTS-full, uncertainty is computed via entropy of the softmax distribution and the activation of the HLSN.

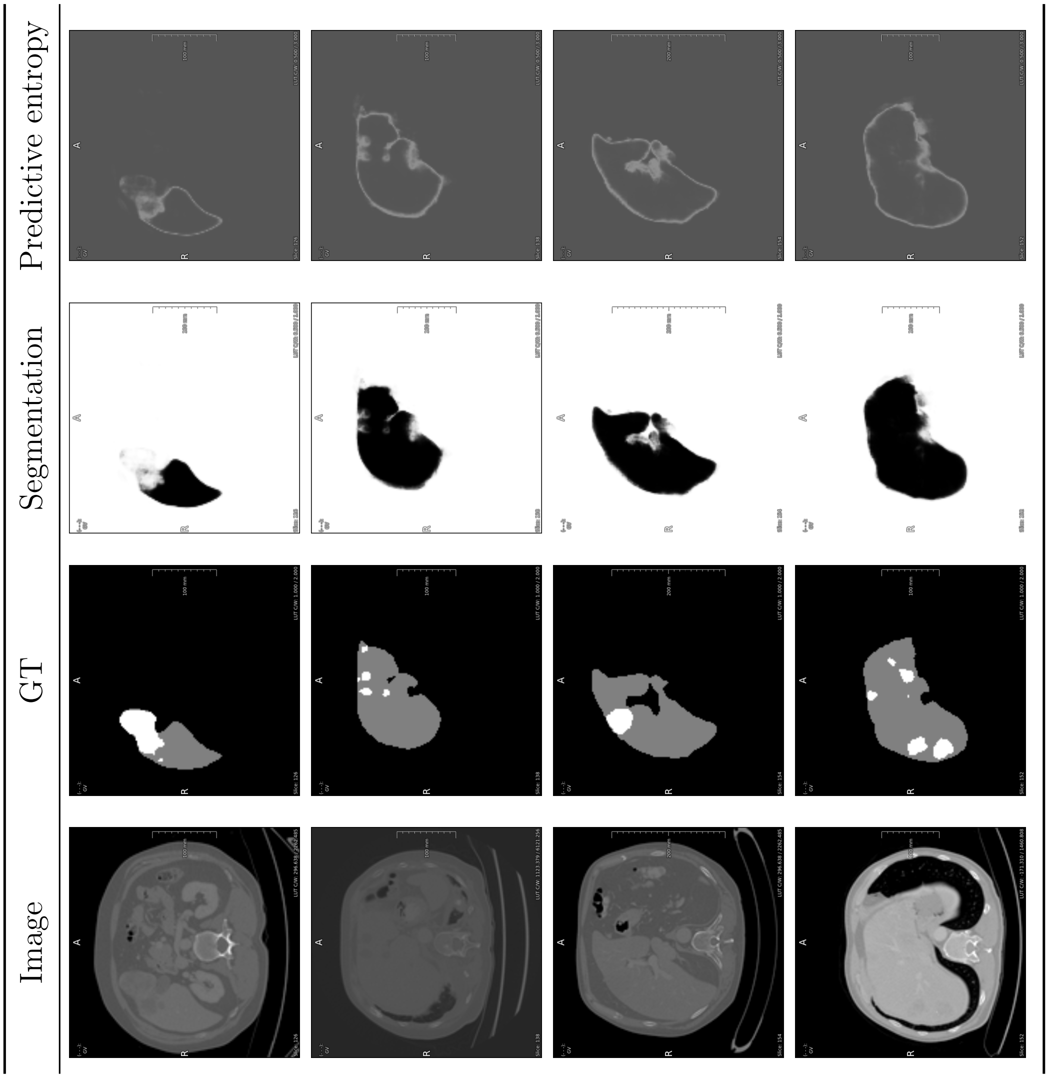


Figure 11: Example output for a BNN-MI model trained on LiTS-full, the overall uncertainty is computed via predictive entropy.

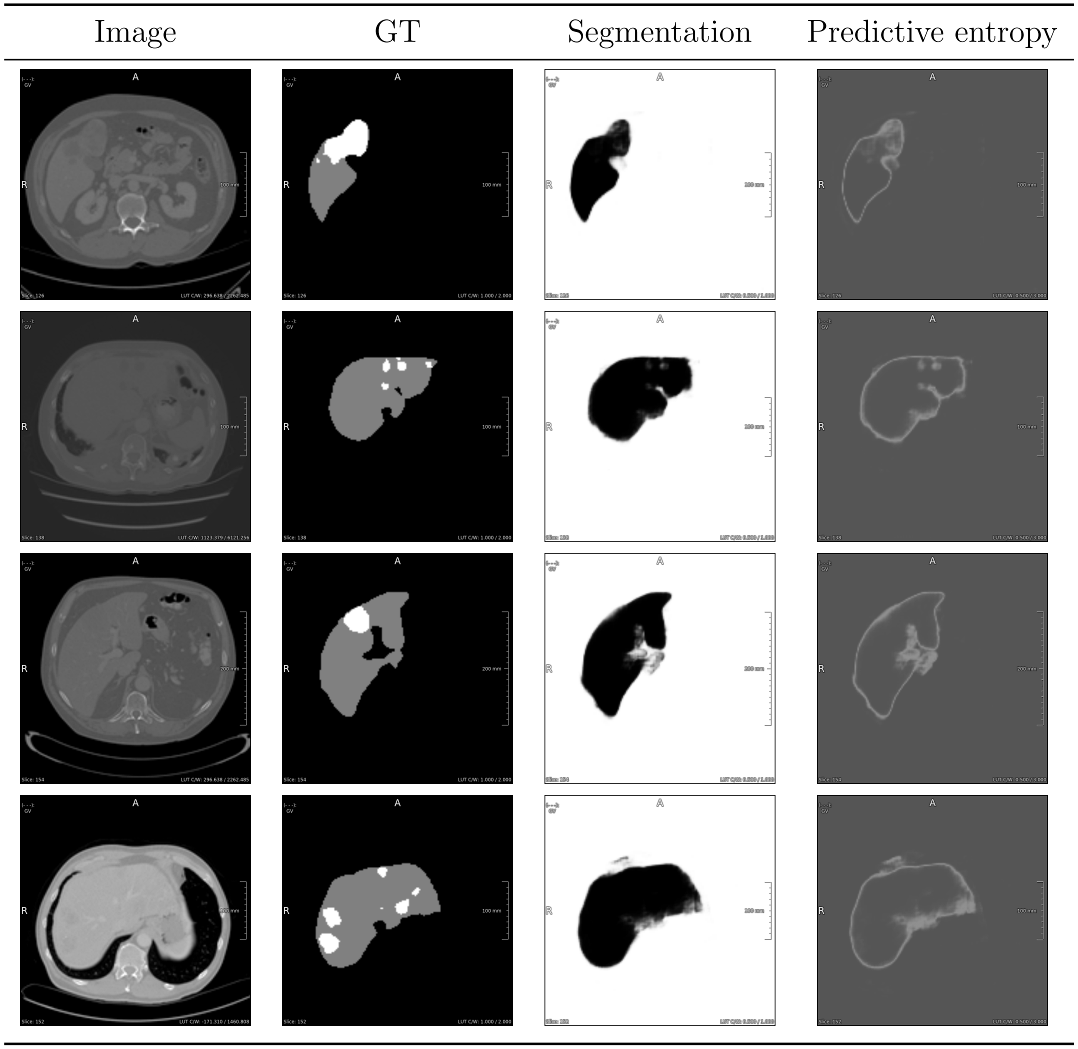


Figure 12: Example output for a BNN-HLS model trained on LiTS-full, the overall uncertainty is computed via predictive entropy.

G Decomposed Uncertainty Maps

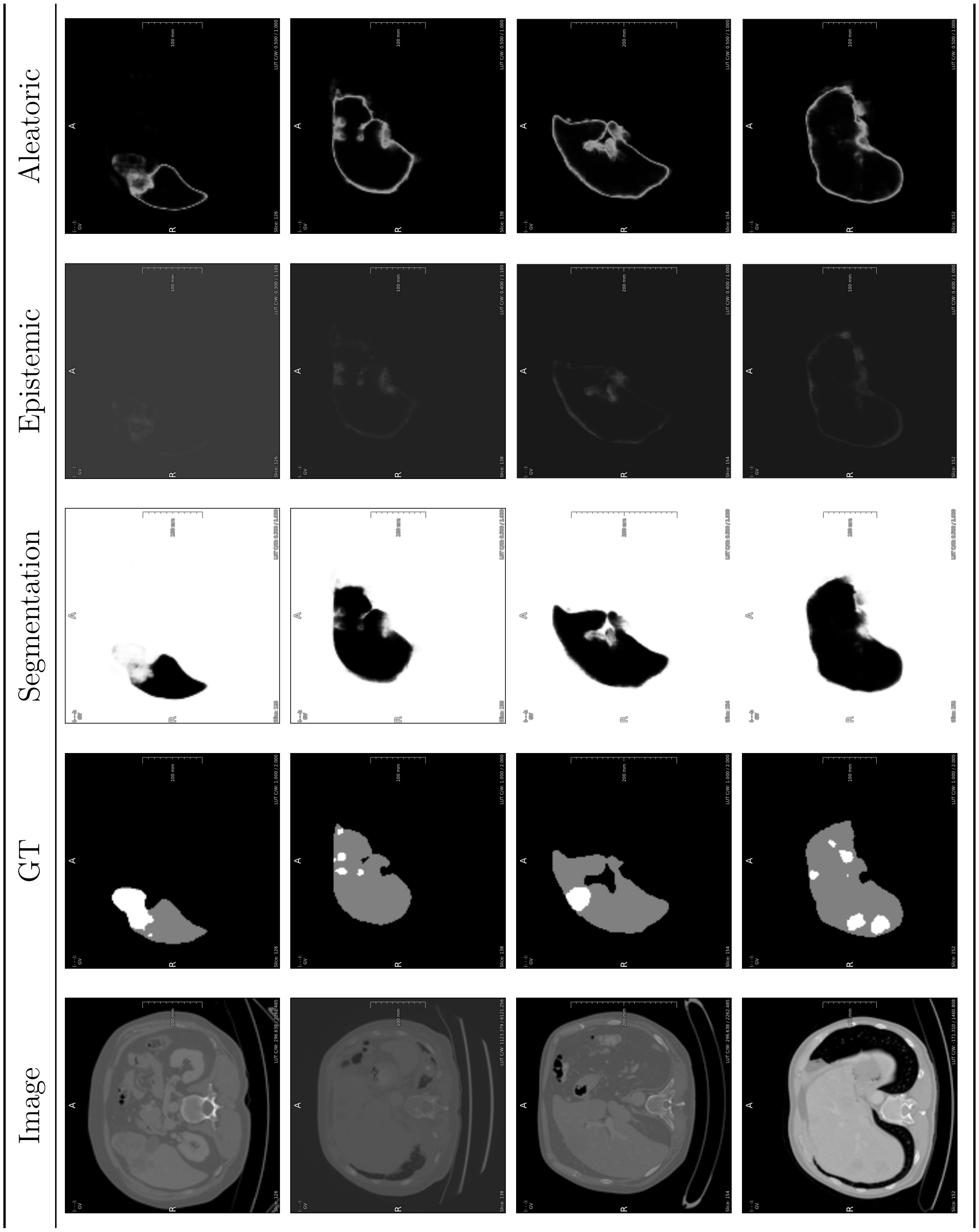


Figure 13: Example segmentation and uncertainty maps for a BNN-MI model trained on LiTS-full

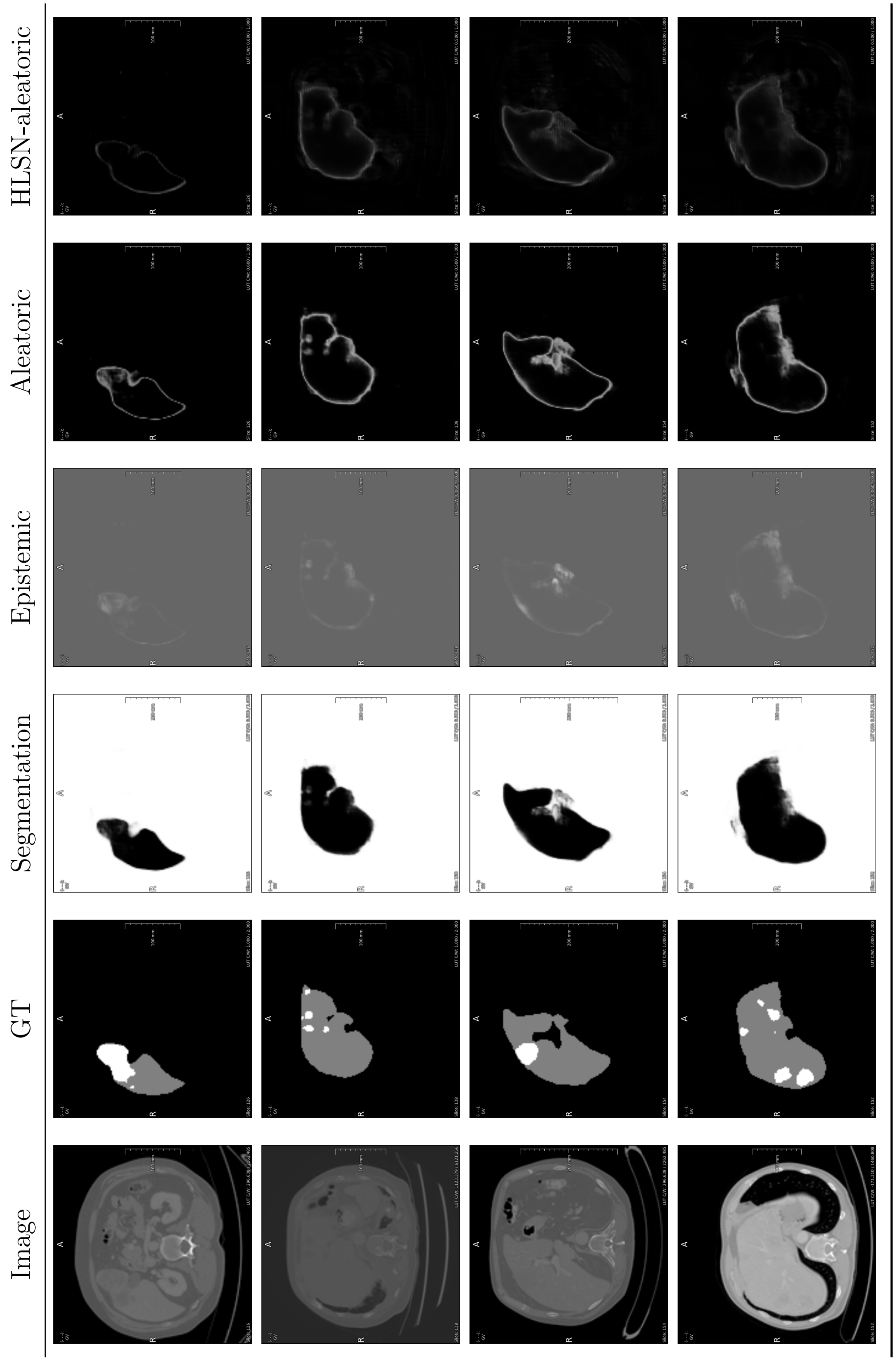


Figure 14: Example segmentation and uncertainty maps for a BNN-HLS model trained on LiTS-full

H Statistical Significance Tests for Decomposed Uncertainties

Table 5: Wilcoxon signed-rank test results for difference in epistemic and aleatoric uncertainty derived from the BNN-MI models when trained on LiTS-full and inferring on LiTS-test vs. other training and test set settings, as denoted in the first column.

	Epistemic		Aleatoric	
	t	p	t	p
LiTS-6	1.0	5.9e-7	97.0	0.001
LiTS-2	1.0	5.9e-7	0.0	5.4e-7
LiTS-noisy	0.0	5.3e-7	0.0	5.4e-7
LiTS-rot	153.0	0.02	148.0	0.02

Table 6: Wilcoxon signed-rank test results for the difference in epistemic and aleatoric uncertainty as derived via MI decomposition of the BNN-HLS models when trained on LiTS-full and inferring on LiTS-test vs. other training and test set settings, as denoted in the first column.

	Epistemic		Aleatoric	
	t	p	t	p
LiTS-6	0.0	5.4e-7	0.0	5.4e-7
LiTS-2	0.0	5.4e-7	51.0	4.1e-5
LiTS-noisy	0.0	5.4e-7	0.0	5.4e-7
LiTS-rot	14.0	1.9e-6	231.0	0.38

Table 7: Wilcoxon signed-rank test results for the difference in HLSN-derived aleatoric uncertainty of the BNN-HLS models when trained on LiTS-full vs. when trained on LiTS-noisy.

	HLSN activation	
	t	p
LiTS-noisy	61.0	8.8e-5

Table 8: Paired student's t-test results for the difference in HLSN-derived aleatoric uncertainty of the BNN-HLS models when trained on LiTS-full and inferring on LiTS-test vs. other training and test set settings, as denoted in the first column.

	HLSN activation	
	t	p
LiTS-6	-5.2	1.2e-5
LiTS-2	-32.9	3.2e-26
LiTS-rot	11.6	5.4e-13