Looking at the posterior: on the origin of uncertainty in neural-network classification

H. Linander¹, O. Balabanov², H. Yang¹, B. Mehlig¹

¹ Department of Physics, Gothenburg University, 41296 Gothenburg, Sweden
² Department of Physics, Stockholm University, 10691 Stockholm, Sweden

Abstract. Bayesian inference can quantify uncertainty in the predictions of neural networks using posterior distributions for model parameters and network output. By looking at these posterior distributions, one can separate the origin of uncertainty into aleatoric and epistemic. We use the joint distribution of predictive uncertainty and epistemic uncertainty to quantify how this interpretation of uncertainty depends upon model architecture, dataset complexity, and data distributional shifts in image classification tasks. We conclude that the origin of uncertainty is subjective to each neural network and that the quantification of the induced uncertainty from data distributional shifts depends on the complexity of the underlying dataset. Furthermore, we show that the joint distribution of predictive and epistemic uncertainty can be used to identify data domains where the model is most accurate. To arrive at these results, we use two common posterior approximation methods, Monte-Carlo dropout and deep ensembles, for fully-connected, convolutional and attention-based neural networks.
1. Introduction

Recently there has been an intensive effort towards better understanding uncertainty of neural-network predictions [1, 2]. To quantify this uncertainty, and to identify its sources, is of key significance in many applications of machine-learning algorithms using neural networks, from real-time predictions to active learning [3]. Bayesian inference provides a theoretical framework to reason about uncertainties by quantifying probability distributions of the neural-network output given the model architecture and training data [4]. More precisely, given a neural network with parameters $\theta$ and a training dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots\}$ of pairs (input, target), Bayesian arguments determine a distribution over the neural-network parameters $p(\theta|\mathcal{D})$ [5]. This so-called posterior distribution tells us the probability of different model parameters given the training dataset. Using this posterior distribution for the parameters, a corresponding posterior distribution of the neural-network predictions can be derived, the posterior predictive distribution $p(y|x, \mathcal{D})$. The Bayesian posterior predictive distribution can be computed conditioned on a particular input, either previously unseen or contained in the training dataset. The posterior predictive distribution can be used to obtain a mean prediction, and more importantly to draw conclusions regarding the uncertainty of the model and the data.

In order to characterize the posterior predictive distribution, it is common to look at derived quantities, such as different measures of entropy, to quantify the uncertainty associated with predictions [1]. A fruitful distinction can be drawn between uncertainty stemming from the model or from the data, captured by classifying uncertainty as either epistemic or aleatoric [6].

The usefulness of this classification of uncertainty can be seen when asking specific questions such as “Should we trust the predictions of our neural network in this domain?” A region with low epistemic uncertainty instills confidence that the model has seen enough data to make a valid prediction but might still have a high aleatoric uncertainty due to, for example, the nature of the input. On the other hand, a region with high epistemic uncertainty is not expected to provide meaningful predictions. In terms of active learning [7], answers to questions such as “What data should we annotate to improve our model?” requires us to know of the origin of uncertainty: high epistemic uncertainty corresponds to regions where the model can improve by providing more training data.

Bayesian modeling makes the distinction between these sources clear. The posterior probability $p(\theta|\mathcal{D})$ of the neural-network parameters $\theta$ given the observations $\mathcal{D}$ contains the epistemic uncertainty, whereas the likelihood $p(y|\theta, x)$ given model parameters $\theta$ and input data $x$ contains the aleatoric part. The predictive distribution, i.e. the marginalization over model parameters $\theta$ using the posterior $p(\theta|\mathcal{D})$

$$p(y|\mathcal{D}, x) = \int_{\theta} p(y|\theta, x)p(\theta|\mathcal{D})d\theta,$$

contains, through the two factors in the integrand, a mixture of aleatoric and epistemic
Looking at the posterior: on the origin of uncertainty in neural-network classification

uncertainty. In very simple examples, the posterior can be computed in closed form (see Appendix A for an example) or with much computational effort [8], but usually we must rely on approximations introducing deviations from the true posterior. A computationally tractable approximation for the posterior predictive distribution is dropout inference [9], where the posterior is sampled by applying dropout in multiple inference passes for the same input, effectively sampling from different parameter configurations. It can be shown that this yields an approximation to the Bayesian posterior [10]. A more computationally demanding, but conceptually simple, alternative is deep ensembling [11] where a neural network is trained multiple times starting from different parameter initializations, creating an ensemble distribution to provide an approximation to the true posterior. Recent results show that deep ensembles provide a good approximation to the Bayesian posterior [8, 3, 12].

Although uncertainty quantification for neural networks in terms of aleatoric and epistemic uncertainty has been the subject of much work, application and analysis of the different uncertainties continues to be an active field of research. The large set of possible neural-network architectures and data domains present many concrete settings where an understanding of the uncertainty content is important for a wide range of applications. Some of the most important open questions in this context are: How does the separation of uncertainty into aleatoric and epistemic parts depend on the neural-network architecture? For a given model, how does the complexity of the dataset impact the perceived uncertainty for different forms of data distributional shifts? For which models and data domains can uncertainty quantification help us improve our model predictions?

To answer these questions we evaluate uncertainty quantification using the joint distribution of total predictive uncertainty and epistemic uncertainty. Our results show that different neural-network architectures can disagree about the aleatoric and epistemic uncertainty of the same data. In other words, the origin of uncertainty is not objective. Using the joint distribution of total predictive uncertainty and epistemic uncertainty, we quantify how the approximate posteriors of three common neural-network architectures for image classification differ from each other and how they depend on data distributional shifts in the form of impulse noise [13] for MNIST [14] and CIFAR [15]. Furthermore, we show that the joint distribution of total predictive uncertainty and epistemic uncertainty can be used to identify data domains where a model is accurate, and that it is difficult to do so using the predictive or epistemic uncertainty alone. We use two common approximation methods for the posterior predictive distribution, dropout inference and deep ensembles, to evaluate predictive and epistemic uncertainty.

2. Contributions

• We use the joint distribution of predictive entropy and mutual information of the neural-network parameter posterior to quantify the variability of uncertainty measures over different model architectures, data distributional shifts and posterior
Looking at the posterior: on the origin of uncertainty in neural-network classification

approximation methods.

- We show that neural networks with different architectures can disagree about the origin of uncertainty for data distributional shifts in image classification tasks.
- We show that the complexity of the dataset impacts the sensitivity of the uncertainty measures to data distributional shifts.
- We quantify the correlation between model accuracy and the joint distribution of total and epistemic uncertainty.
- We evaluate our results by using two different posterior approximation schemes, Monte-Carlo dropout and deep ensembles, in order to quantify the extent to which our conclusions depend on the posterior approximation.

3. Related work

Ref. [16] uses both uncertainty measures separately in connection with active learning, concluding that epistemic uncertainty provides a better selection criterion than the total predictive uncertainty. In the context of reinforcement learning Ref. [16] use a combination of aleatoric and epistemic uncertainty to find balanced policies. The joint distribution, architecture dependence, and dataset dependence is not considered. In Ref. [17], the correlation between accuracy and uncertainty is quantified for image classification. The correlation with accuracy is evaluated for total predictive uncertainty and epistemic uncertainty separately, the architecture and dataset dependence is not discussed. The joint distribution of aleatoric and epistemic uncertainty is considered in Ref. [18], where the correlation between predictive probabilities and the joint uncertainty distribution is quantified in the context of medical image semantic segmentation. Using a fixed residual U-Net architecture and datasets for semantic segmentation it was shown that there is a correlation between predictive probabilities and the uncertainty measures, and that accuracy for their semantic segmentation model on the considered datasets shows correlation with epistemic uncertainty. Ref. [18] conclude that open questions include the effect of data distributional shifts, model architecture, and data modality on the quality of uncertainty quantification. These are in line with two of our target questions on how the perceived origin of uncertainty depends on model architecture and how dataset complexity impacts uncertainty quantification. Quantifying uncertainty under data distributional shifts was investigated in Ref. [19] where accuracy, calibration and entropy of the posterior predictive is evaluated for different shifts introduced in Ref. [20]. Here, the epistemic uncertainty is not considered and the model dependence of the uncertainty measures is not analyzed.
4. Background

4.1. Bayesian inference

An artificial neural network $f$ with parameters $\theta \in \mathbb{R}$ can be seen as a map from input space $X$ the output space $Y$ [21], where we take $Y$ to be the space of distributions over possible outcomes so that the likelihood is given by $p(y|\theta) = f(y; \theta)$. Bayesian inference [22] allows us to reason about uncertainty in terms of posterior distributions for the parameters of a model. With a training dataset $D \in X \times Y$ corresponding to observations $(x_t, y_t)$ with $t \in \{1 \cdots N\}$, the posterior distribution for the neural-network parameters $p(\theta|D)$ can be calculated using Bayes theorem in terms of the likelihood as

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)},$$

(2)

where $p(D|\theta)$ (assuming independent samples in the dataset $D$) can be expressed as $p(D|\theta) = \prod_{i=1}^{N} f(y_i, x_i; \theta)$, $p(\theta)$ the parameter prior and the evidence $p(D)$ is the marginalization over the parameter prior.

A prediction, or more generally a distribution over possible outputs, is computed using the posterior predictive distribution in (1) by marginalizing over the parameters. Each parameter configuration is weighted by its posterior probability given the training data. For large neural-network architectures, the integral over the parameter space $\theta$, which is typically some subset of $\mathbb{R}^n$ with large $n$, is both analytically and computationally [8] infeasible, see Appendix A for a simple toy example where the Bayesian posterior is computed in closed form and evaluated numerically.

To implement Bayesian inference for neural networks, the posterior needs to be approximated, as mentioned in the introduction. There are a number of different methods available that range from computationally expensive Monte-Carlo simulations [4] to more efficient dropout approximations [9] and simpler ensembling methods [11]. The most accurate approximation to the true posterior available are the Hamiltonian Monte-Carlo methods. Through intensive computational efforts these have recently been used to compute the posteriors of larger convolutional neural networks such as a 20-layer ResNet [8]. The HMC computations show that simpler approximation schemes such as ensembling and variational inference can fail to accurately describe the true posterior, but that ensembles often provide more accurate posteriors than more advanced methods.

4.2. Dropout posterior

The posterior predictive distribution can be approximated using dropout [9]. In this scheme, the posterior predictive distribution for a sample $x$ is calculated as

$$p(y|x, D) = \int_{\theta} f(y, x; \theta)p(\theta|D) \, \text{d}\theta \approx \frac{1}{N} \sum_{i=0}^{N} f(y, x; \theta^{(i)}),$$

(3)
Looking at the posterior: on the origin of uncertainty in neural-network classification

where $\theta^{(i)}$ are $N$ samples of parameters using dropout indexed by $i$, of a neural network trained with dropout [23] and $L^2$-regularization [24] and $N$ is the ensemble size.

For a classification problem over a finite number of classes $c \in \{1, \cdots, M\}$, the variable $y$ is discrete, so that (3) becomes

$$p(y = c|x, D) \approx \frac{1}{N} \sum_{i=0}^{N} f_c(x; \theta^{(i)}).$$

(4)

The dropout approximation is equivalent to variational inference when the prior distribution over the weights together with the choice of distribution family $q(\theta)$ satisfies certain conditions [9, 25]. Variational inference is itself an approximation scheme to the true posterior, so that even if dropout approaches variational inference for a particular choice of distributional family $q(\theta)$, it may still be a bad approximation to the true posterior.

4.3. Ensemble posterior

A second, frequently employed method to sample from the model space is to train an ensemble [11] of $N$ identical neural networks using different initial parameter values $\theta^{(i)}_{\text{initial}}$ sampled from some prior distribution. Training these neural networks to maximize the likelihood of the training data gives a set of parameters $\theta^{(i)}$, that then provides an approximation to the Bayesian mean by equation (4), where $\theta^{(i)}$ are now the ensemble member parameters instead of the dropout samplings. Here it is assumed that the likelihood of the minima attained by $\theta^{(i)}$ are equally probable. Note that it is not clear a priori whether the minima $\theta^{(i)}$ are degenerate, but for the regime of neural networks for visual perception it is typically the case that they are not [26].

The difference between the dropout posterior and ensemble posterior lies in the way the parameters are sampled. For dropout, the parameters are sampled by dropout inference whereas for ensembles the different parameter sets are trained independently. The ensemble members are local maximums of the likelihood function whereas a dropout sampling tends to sample around such points [25].

4.4. Uncertainty quantification

For classification over a discrete set of $M$ classes, the entropy [27] of the predictive distribution

$$H(p) = - \sum_{c=1}^{M} p(c) \log(p(c)),$$

(5)

provides a measure of the information content and thus its uncertainty.

In terms of the dropout and ensemble approximations of the posterior predictive distribution in (4), this takes the form

$$H(y|x, D) = - \sum_{c=1}^{M} \left( \frac{1}{N} \sum_{i=1}^{N} f_c(x; \theta^{(i)}) \right) \log \left( \frac{1}{N} \sum_{i=1}^{N} f_c(x; \theta^{(i)}) \right),$$

(6)
Looking at the posterior: on the origin of uncertainty in neural-network classification

where $H(y|x, D) \equiv H(p(y|x, D))$ is the entropy of the posterior predictive distribution, referred to here as total predictive uncertainty, and $H(\theta|D, (x, y))$ is the entropy of the posterior given a training set containing the union $D \cup (x, y)$. Looking at the posterior predictive distribution in (1), we see that the entropy of $p(y|x, D)$ in equation (6), as a measure of uncertainty, contains contributions that are both epistemic and aleatoric.

Epistemic uncertainty stems from uncertainty in model parameters. This is captured by the shape of the posterior distribution of the model parameters. To quantify the epistemic uncertainty associated with a single data sample $x$ we can ask how this shape changes [5]. One says that the epistemic uncertainty associated with a data sample $x$ is measured by the expected change in entropy of the model parameter posterior distribution $p(\theta|D)$ when $x$ is added to the observations [5],

$$I(x) = E_{p(y|x, D)}[H(\theta|D, (x, y)) - H(\theta|D)],$$

where $H(\theta|D) = H(p(\theta|D))$. The entropy difference in equation (7) can be related to the more easily computable entropy of the posterior predictive distribution [28]:

$$I(x) = H(y|x, D) - E_{p(\theta|D)}[H(y|\theta, x)],$$

where $H(y|\theta, x)$ is the entropy of the likelihood $p(y|\theta, x)$. See Appendix A for an illustration of this relation in a toy model. The first term in (8) is the entropy of the posterior predictive distribution given the dataset $D$, whereas the second term is the expected value of the likelihood entropy over the model posterior distribution. An equivalent way of arriving at equation (8) is compute the conditional mutual information $I(\theta, y|x, D)$ between the model parameters and target $y$ conditioned on the dataset $D$ and the new sample $x$ [29], motivating our simplified notation $I(x)$ for the same quantity as a function of the new data sample $x$.

Using the approximate posteriors from the dropout and ensemble methods, the first term in (8) is given by (6) and the second term is given by

$$E_{p(\theta|D)}[H(y|\theta, x)] = \frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{M} f_c(x; \theta^{(i)}) \log (f_c(x; \theta^{(i)})].$$

Together, equations (6) and (9) provide a concrete way to evaluate epistemic uncertainty as defined by (8) in practice.

The epistemic uncertainty in (8) is large when the posterior predictive entropy is large and the mean likelihood entropy is small. In terms of the ensemble members this corresponds to the situation where each member has a sharp distribution but they disagree about the mean. Large aleatoric uncertainty is ascribed to broad output distributions from the individual members $f(\cdot; \theta^{(i)})$, also implying a large posterior predictive entropy. Since entropy is positive, both the term $H(y|x, D)$ and the expected entropy of the likelihood in (8) are positive, and so the entropy difference in (8) is bounded by the posterior entropy in (6), resulting in the inequality

$$I(x) \leq H(x).$$
Looking at the posterior: on the origin of uncertainty in neural-network classification

This means that a large epistemic uncertainty implies a large posterior predictive entropy. This can be seen by noting that a collection of sharp member distributions that disagree about the mean necessarily adds up to a broad mean distribution. One way to describe a sample with large epistemic uncertainty is that the model might fit the data well, but in many different ways. The parameters with high posterior probability could all result in sharp distributions, whereas the full posterior can be broad [30].

Since the epistemic uncertainty in (8) measures the change in posterior entropy, a model can have some irrelevant parameters that contribute to a broad posterior distribution, but the change in posterior entropy can still be small for a given sample. If the parameter posterior for some irrelevant parameter is equally broad after we add sample \( x \), then we do not want to consider this as a point of high epistemic uncertainty.

In summary, the entropy \( H(y|x, D) \) of the posterior predictive distribution \( p(y|x, D) \) in (6) and the mutual information in (8) quantify the total predictive uncertainty and epistemic uncertainty respectively.

5. Methods

5.1. Datasets

Two datasets are used for all numerical experiments, MNIST [14] and a grayscale version of CIFAR10 [31]. To be able to compare relative shifts in uncertainty measures between the two datasets we choose to work with a grayscale version of CIFAR denoted CIFAR10G so that the data distributional shift acts in exactly the same way for both MNIST and CIFAR10G. MNIST is an example of an image classification dataset with minimal complexity, whereas CIFAR10 provides a more realistic data distribution for image classification. The grayscale conversion for the RGB data from CIFAR10 is given by the standard BT.601 luminance \( Y = 0.2989r + 0.5870g + 0.1140b \). We apply impulse noise [13], a common corruption present in digital images, to the original data sets (MNIST, CIFAR10G) where the strength of the perturbation is controlled by a noise parameter \( \alpha \). For “salted” noise with parameter \( \alpha > 0 \) a random sample of pixels of size \( \alpha N_{\text{pixels}} \) are given the maximum value 1.0, and for “peppered” noise with \( \alpha < 0 \) a corresponding amount of pixels are set to 0. We pick \( N_{\alpha} \) distinct values between \( \alpha_{\min} = -0.3 \) and \( \alpha_{\max} = 0.3 \).

MNIST consists of 60000 grayscale images with resolution 28×28. The original CIFAR10 dataset consists of 60000 RGB images with resolution 32×32 that we resample to a single grayscale channel. See figure 1 for examples of the different noise levels.

5.2. Network architectures

In order to quantify how the architecture affects the uncertainty estimates, we use three neural-network architectures: fully connected (dense), convolutional (cnn) and attention-based (swin) neural networks. For the fully connected neural network we use a simple three-layer architecture with two 128-neuron hidden layers using ReLU
Looking at the posterior: on the origin of uncertainty in neural-network classification

Figure 1: MNIST (left) and CIFAR10G (right) with impulse noise parametrized by $\alpha$. Darker colors indicate a value closer to 0. For CIFAR10G the original RGB image is shown in the right-most column. The CIFAR10 classes in the examples are, from top to bottom: airplane, automobile, bird, cat, deer.

| Model  | Parameters | MNIST         | CIFAR10G      |
|--------|------------|---------------|---------------|
| Dense  | 118k       | 99.5% ± 0.07% | 45.8% ± 0.5%  |
| CNN    | 836k       | 99.9% ± 0.06% | 74.1% ± 0.95% |
| Swin   | 147k       | 97.3% ± 0.1%  | 66.2% ± 1.3%  |

Table 1: Summary of the three different neural-network architectures used for comparisons. The accuracy for MNIST and CIFAR10G is over the validation datasets with standard deviation calculated for 10 separate trainings. The dense model uses 3 layers (128, 128, 10) with ReLU activations, CNN is model A in [32] and the transformer is Swin [33].

activations. The convolutional neural network is identical to model A in [32], a simple fully convolutional model with 5 layers using max-pooling for spatial down-sampling and ReLU activations. Finally, we also use Swin-Transformer (swin) [33], a popular attention-based model for computer vision. See Table 1 for a summary of the model sizes and baseline accuracy on the target datasets.

6. Results

6.1. Model architecture

Figure 2 shows, for different neural-network architectures (columns) and posterior approximation methods (rows), the joint distribution of the total predictive uncertainty defined by equation (6) and epistemic uncertainty as defined by equation (8) with histograms in the ($H, I$) plane. The joint distributions in Figure 2 are evaluated on the test set of CIFAR10G shifted by impulse noise with noise parameter $\alpha = 0.09$, corresponding to 9% of pixels set to the maximum grayscale intensity. The joint distribution allows us to separate the origin of uncertainty: a sample with large value of $I$ on the vertical axis has higher epistemic uncertainty, whereas a sample on the
Looking at the posterior: on the origin of uncertainty in neural-network classification

horizontal axis has no epistemic uncertainty. The dashed red diagonal indicates the bound on epistemic uncertainty imposed by equation (10). In particular this means that a sample close to the epistemic bound is dominated by epistemic uncertainty. The mean and standard deviation of the joint distributions in Figure 2 are summarized in Table 4 together with prediction accuracies. For reference, the moments and accuracy on the unshifted test set is shown in Table 2.

In Figure 2, for the fully connected neural network (left column), convolutional neural network (middle column) and Swin model (right column), the joint distribution using ensembling (top) and dropout (bottom) show rough agreement. For CIFAR10G, the posterior of the convolutional neural network shows the most disagreement between the ensemble and dropout posterior, as can be verified numerically in table 4.

The center column of panels in Figure 2 shows the joint uncertainty distribution for the convolutional neural network. Here the perceived epistemic uncertainty is larger compared to the other models. The right-most column of panels in Figure 2 shows that the attention-based model exhibits similar epistemic uncertainty to the convolutional neural network in the middle column of panels. The attention-based model does however perceive a lower aleatoric uncertainty, although this is not accompanied by a significant increase in accuracy.

For MNIST, the joint distributions of predictive and epistemic uncertainty evaluated on the same distributional shift with $\alpha = 0.09$ can be found in Figure 3. Table 3 contains the moments of the joint distributions and the prediction accuracies. In Figure 3 and table 3 we find that for MNIST, the joint distributions for ensemble (top row) and dropout (bottom row) of the dense (left column) and convolutional (center column) neural networks show better agreement, whereas the Swin model shows a larger discrepancy between the different posterior approximation methods.

6.2. Dataset

To evaluate the dependence of the uncertainty quantification on the training dataset we calculate the induced shift in joint distribution of predictive and epistemic uncertainty when we apply the same data distributional shift to neural networks trained on MNIST and CIFAR10G. In Figure 4 the observed shift in joint distribution for the convolutional neural network is shown from $\alpha = 0$ (top) to $\alpha = 0.09$ (bottom) for both MNIST (left) and CIFAR10G (right). The shift in mean predictive and epistemic uncertainty is quantified in table 5. For the dense and Swin model we observe a significantly smaller shift in both $H$ and $I$ on CIFAR10G compared to MNIST, whereas the convolutional neural network perceives a larger shift on CIFAR10G compared to MNIST.

6.3. Accuracy

Figures 5 and 6 show the accuracy of the predictions of three neural networks conditioned on entropy regions and posterior approximations evaluated on the union of all noise
Looking at the posterior: on the origin of uncertainty in neural-network classification

H = 1.66 ± 0.39
I = 0.10 ± 0.04
acc = 41%
dense ensemble

H = 1.92 ± 0.22
I = 0.39 ± 0.11
acc = 19%
cnn ensemble

H = 1.12 ± 0.47
I = 0.27 ± 0.14
acc = 20%
winn ensemble

H, Predictive uncertainty
I, Epistemic uncertainty

Figure 2: Joint distribution \((H, I)\), corresponding to predictive uncertainty and epistemic uncertainty, for different neural networks trained on CIFAR10G and evaluated on noised data with noise parameter \(\alpha = 0.09\). For each model, the joint entropy distribution is shown for dropout (top) and ensemble (bottom). Inset in each frame is the noise level \((\alpha)\), accuracy of the model \((\text{acc})\) on the noised data together with mean and standard deviation of both axes. The dashed diagonal indicates the bound on epistemic uncertainty from equation (10).

| Architecture | Posterior | Acc. | H (nat) | I (nat) |
|--------------|-----------|------|---------|---------|
| dense        | ensemble  | 45%  | 1.58 ± 0.41 | 0.07 ± 0.04 |
| dense        | dropout   | 38%  | 1.85 ± 0.30 | 0.10 ± 0.04 |
| cnn          | ensemble  | 75%  | 1.07 ± 0.64 | 0.14 ± 0.10 |
| cnn          | dropout   | 64%  | 1.27 ± 0.61 | 0.11 ± 0.06 |
| swin         | ensemble  | 65%  | 0.90 ± 0.55 | 0.08 ± 0.06 |
| swin         | dropout   | 59%  | 1.01 ± 0.52 | 0.08 ± 0.06 |

Table 2: Mean and standard deviation for predictive and epistemic uncertainty and accuracy for fully connected (dense), convolutional (cnn) and attention-based (swin) models with posterior approximations using ensembles and dropout evaluated on CIFAR10G without data distributional shift.

levels, including the uncorrupted test set. We see that lower epistemic and aleatoric uncertainty does not always imply higher accuracy. The top middle panel of figure 6 shows that for the convolutional neural network, for a fixed moderate predictive uncertainty, the accuracy is not monotonically increasing with epistemic uncertainty. On the other hand, the attention-based Swin architecture in the top right panel shows a monotonically increasing accuracy with decreasing epistemic uncertainty.

The dropout posterior struggles to separate the high and low accuracy regions.
Looking at the posterior: on the origin of uncertainty in neural-network classification

H = 0.54 ± 0.47
I = 0.35 ± 0.32
acc = 65%
dense ensemble

H = 0.67 ± 0.54
I = 0.25 ± 0.22
acc = 73%
cnn ensemble

H = 0.69 ± 0.44
I = 0.44 ± 0.30
acc = 73%
cnn dropout

H = 0.57 ± 0.50
I = 0.29 ± 0.28
acc = 69%
dense dropout

H = 0.72 ± 0.54
I = 0.25 ± 0.21
acc = 73%
cnn dropout

H = 0.46 ± 0.38
I = 0.21 ± 0.19
swin dropout

Table 3: Mean and standard deviation for predictive and epistemic uncertainty for fully connected (dense), convolutional (cnn) and attention-based (swin) models with posterior approximations using ensembles and dropout for MNIST at fixed data distributional shift α = 0.09.

Comparing e.g. the two rightmost panels in figure 5, the ensemble posterior clearly separates a region of high accuracy whereas the dropout posterior mixes up high- and low accuracy samples. In the same figure the dropout posterior for the convolutional neural network also shows a degraded separation of high accuracy regions. For the simple fully connected neural network there is a more pronounced correlation between the predictive uncertainty and model accuracy.
Looking at the posterior: on the origin of uncertainty in neural-network classification

| Architecture | Posterior | Acc. | H (nat)        | I (nat)        |
|--------------|-----------|------|----------------|----------------|
| dense        | ensemble  | 41%  | 1.66 ± 0.39    | 0.10 ± 0.04    |
| dense        | dropout   | 36%  | 1.91 ± 0.27    | 0.10 ± 0.04    |
| cnn          | ensemble  | 20%  | 1.92 ± 0.22    | 0.39 ± 0.11    |
| cnn          | dropout   | 23%  | 1.65 ± 0.39    | 0.20 ± 0.07    |
| swin         | ensemble  | 20%  | 1.12 ± 0.47    | 0.27 ± 0.14    |
| swin         | dropout   | 24%  | 1.10 ± 0.42    | 0.18 ± 0.12    |

Table 4: Mean and standard deviation for predictive and epistemic uncertainty for fully connected (dense), convolutional (cnn) and attention-based (swin) models with posterior approximations using ensembles and dropout evaluated on CIFAR10G at fixed data distributional shift $\alpha = 0.09$.

Figure 4: Change in joint distribution $(H, I)$ from $\alpha = 0$ (top) to $\alpha = 0.09$ (bottom) on CIFAR10G (left) and MNIST (right) for the convolutional model.

7. Discussion

7.1. Model architecture

There are three points we want to make regarding how uncertainty quantification depends on model architecture.

First, the origin of uncertainty is not objective. Table 2 shows that there are significant differences in the perceived origin of uncertainty between the different model architectures for in-domain data. For CIFAR10G, the fully connected neural network perceives a higher degree of aleatoric uncertainty, and low epistemic uncertainty
Looking at the posterior: on the origin of uncertainty in neural-network classification

| Architecture | Dataset | $\alpha$ | $\Delta H$ (nat) | $\Delta I$ (nat) |
|--------------|---------|---------|----------------|----------------|
| dense        | mnist   | 0.09    | 0.49           | 0.33           |
| dense        | cifar   | 0.09    | 0.08           | 0.03           |
| cnn          | mnist   | 0.09    | 0.65           | 0.24           |
| cnn          | cifar   | 0.09    | 0.85           | 0.25           |
| swin         | mnist   | 0.09    | 0.63           | 0.41           |
| swin         | cifar   | 0.09    | 0.22           | 0.20           |

Table 5: Change in mean predictive and epistemic uncertainty, $\Delta H$ and $\Delta I$, for fully connected (dense), convolutional (cnn) and attention-based (swin) models from $\alpha = 0$ to $\alpha = 0.09$. Posterior approximation by ensembling.

Figure 5: Accuracy distribution in the $(H, I)$-plane for a union over noise levels $-0.27 \leq \alpha \leq 0.27$ evaluated on CIFAR10G. Prediction accuracy conditioned on the joint distribution of posterior predictive entropy $H$ and epistemic uncertainty $I$ is shown for the ensemble posterior (top) and dropout posterior (bottom).

compared to the convolutional and attention-based model. This shows that the origin of uncertainty depends on the model architecture: the fully connected neural network struggles to express the relationship between inputs and classes of CIFAR10G, and thus perceives a higher degree of aleatoric uncertainty. Even though the model has
Figure 6: Accuracy distribution in the \((H, I)\)-plane for a union over noise levels \(-0.27 \leq \alpha \leq 0.27\) evaluated on MNIST. Prediction accuracy conditioned on the joint distribution of posterior predictive entropy \(H\) and epistemic uncertainty \(I\) is shown for the ensemble posterior (top) and dropout posterior (bottom).

For CIFAR10G, Figures 2 and Table 4 show that for a moderate data distributional shift, the fully connected neural network perceives a higher degree of aleatoric uncertainty and low epistemic uncertainty compared to the convolutional and attention-based models.

Second, the different model posteriors agree about the origin of uncertainty when the dataset complexity is low. In the case of MNIST, Figures 3 and Table 3 show that all neural-network architectures considered here show rough agreement between the joint distributions of predictive and epistemic uncertainty. For the considerably less complex dataset of MNIST, as evident by the higher accuracies in Table 1, all three architectures can easily capture the features necessary. Hence, they also agree about the origin uncertainty. The simplicity of the dataset is also reflected in that the different posterior approximations show more agreement compared with CIFAR10G, with the exception of the attention-based model.

Third, the accuracy of posterior approximations depends on architecture. The two posterior approximations, ensemble and dropout, show particularly good agreement...
Looking at the posterior: on the origin of uncertainty in neural-network classification

regarding the epistemic uncertainty of the fully connected model as shown in Table 4. On CIFAR10G, for the convolutional and attention-based neural networks there are larger differences between the approximations, and the posteriors for the convolutional model show the least agreement. Thus, a posterior approximation method that is robust for one architecture, might not perform as well for another.

In summary, besides looking for performant architectures, we should also search for architectures and posterior approximations that provide robust uncertainty quantification for a given data domain.

7.2. Dataset

We discuss two points concerning how the posteriors depend on the dataset.

First, the relative change in perceived uncertainty under a fixed data distributional shift depends on the dataset, and varies with model architecture. By comparing the change in the joint distribution under a fixed data distributional shift in Figure 4 we observe that a given model architecture perceive the same data distributional shift in different ways, depending on the underlying training dataset. We find an asymmetry in shift sensitivity: only the convolutional model shows a stronger sensitivity of the predictive and epistemic uncertainty when evaluated on CIFAR10G compared to MNIST, as evident by Table 5 and Figure 4. Furthermore, the fully connected model perceives the impulse noise on CIFAR10G very differently from the impulse noise on MNIST. The distributional shift for MNIST induces a large change in both epistemic and predictive uncertainty, whereas for CIFAR10C the induced shift in uncertainty is significantly smaller. Since CIFAR10C contains realistic digital images, one might expect impulse noise to be more in-domain than for MNIST. The converse is true for the convolutional model, where the induced shift is larger for CIFAR10G. Thus, the sensitivity to a particular data distributional shift depends on the data domain and on neural-network architecture. Even though this difference in perceived relative change of uncertainty is present for widely different data, it also has implications for different domains in the same training dataset, something that would be interesting to quantify in more detail.

Secondly, robustness of prediction accuracy under data distributional shifts for a given model depends on the dataset. For CIFAR10G, the higher accuracy of the fully connected neural network on noised data in Table 4 shows that this architecture is more robust against this particular distributional shift, even though the model is less accurate close to the training domain as seen in Table 2. For MNIST, the convolutional model is instead more robust than both the dense and attention-based architectures as seen in Table 3.

7.3. Accuracy

In this section we discuss three points regarding the connection between the joint distribution of epistemic and total predictive uncertainty, and prediction accuracy.
First, looking at the joint distribution helps to identify samples where a given model is more accurate. One important goal of uncertainty quantification is to assess when the predictions of a model can be trusted. It is natural to expect that a prediction is more accurate when the total predictive uncertainty is low. We show that the joint distribution of epistemic and total uncertainty can identify accurate predictions where predictive uncertainty or epistemic uncertainty alone cannot. This can be seen in Figures 5 and 6, where the joint distribution clearly resolves where the models are more accurate. Figures 5 and 6 show that the projection of the joint distribution on either the axis of predictive uncertainty or the axis of epistemic uncertainty mixes samples with high and low accuracy. As a consequence, selecting for predictive uncertainty or epistemic uncertainty alone, is not as effective in identifying where a given model is more accurate. This confirms observations made in e.g. Refs. [18, 34], regarding the clustering of incorrect predictions at high uncertainty.

Second, when dropout posteriors differ from ensemble posteriors, they tend to be worse at identifying samples where the model is accurate. In particular, Figure 5 illustrates that approximate posteriors obtained with untuned dropout rates are significantly worse at separating high and low accuracy regions.

Third, for the attention-based model it is difficult to find a threshold of the projected distribution on either predictive uncertainty or epistemic uncertainty that results in high accuracy. The top-right panels in Figures 5 and 6 show a distinct structure for the accuracy conditioned on the joint distribution where the marginalized distribution on either axis mixes samples of high and low accuracy. This has implications for active learning. In the present example, choosing samples based on e.g. posterior predictive uncertainty alone is inefficient, because it results in training on data regions where the model is already accurate.

8. Conclusions

Posterior predictive entropy and mutual information are used extensively as measures of total predictive uncertainty and epistemic uncertainty to assess the uncertainty and performance of neural networks and their predictions [2]. We introduced the joint distribution of predictive uncertainty and epistemic uncertainty and used it to show the variability of uncertainty measures for different neural-network architectures, data distributional shifts and posterior approximation methods. Our main conclusion is that the origin of uncertainty is not objective. Different neural-network architectures disagree about the epistemic and aleatoric uncertainty of the same data. For a given model, the sensitivity of the uncertainty quantification to a specific type of data distributional shift depends on the underlying training dataset. Our results imply that the joint distribution contains information about where a model is more accurate, and can be used to improve the separation of high accuracy regions compared to epistemic and predictive uncertainty alone.

An open question is whether it is possible to explain how uncertainty quantification
Looking at the posterior: on the origin of uncertainty in neural-network classification

depends on architecture from the mathematical theory of neural networks, and how to use this to build architectures that target robust uncertainty quantification. Do the attention-based models provide a more challenging architecture in general for posterior approximation? Recently there have been efforts towards understanding attention-based models from the Bayesian perspective better [35]. In practical application, uncertainty quantification using the Bayesian posterior depends on accurate posterior approximations. Thus it continues to be of utmost importance to find computationally efficient methods to obtain accurate posterior approximations.

Predictive entropy and mutual information are two measures of uncertainty derived from the high-dimensional model posterior and the posterior predictive distribution. If there are other, complementary or more informative, derived quantities that can capture the uncertainty of artificial neural networks better also remains an interesting question.

Acknowledgments

BM and HL were supported in part by Vetenskapsrådet (grant numbers 2017-3865 and 2021-4452). OB was supported by Vetenskapsrådet (grant number 2017-05162). HL would like to thank Erik Werner for enlightening discussions.
Appendix A. Toy model posterior

The relation between mutual information of the parameter posterior and mutual information of the posterior predictive in equation (8) provides a way of calculating the expected change in entropy of the high-dimensional parameter posterior \( p(\theta|D) \) in terms of the typically lower-dimensional posterior predictive distribution \( p(y|x,D) \) and the likelihood \( p(y|x,\theta) \). To illuminate this relation, and evaluate the involved quantities in closed-form, we present a detailed verification for a simple toy model. This also serves as an illustration of the computational complexity involved in computing the Bayesian posterior directly.

The toy problem consists of classifying points on the real line into two classes \( c_1, c_2 \) and we use a simple two-parameter linear model

\[
\begin{align*}
    p(c_1|x, \theta_1, \theta_2) &= \begin{cases} 
        1 & x - \theta_1 < -\theta_2, \\
        \frac{(x-\theta_1+\theta_2)}{2\theta_2} & |x - \theta_1| < \theta_2, \\
        0 & x - \theta_1 > \theta_2,
    \end{cases} \\
    p(c_2|x, \theta_1, \theta_2) &= 1 - p(c_1|x, \theta_1, \theta_2).
\end{align*}
\]

By construction, this model has a strong prior for samples of class 1 being located to the left of a decision boundary at \( \theta_1 \) and class 2 to the right. See figure A1 for the resulting probability distributions for a particular choice of the model parameters \( \theta_1 \) and \( \theta_2 \).

Let \( \theta_1 \) be uniformly distributed on \([-1, 1]\) and \( \theta_2 \) on \([\frac{1}{2}, 2]\). With this prior on the parameters, we can calculate the prior predictive distribution over the two classes by

\[
p(c|x) = \int_{\theta_1} \int_{\theta_2} p(c|x, \theta_1, \theta_2)p(\theta_1, \theta_2)d\theta_1d\theta_2,
\]

visualized in figure A2, where we see that the prior parameter distribution results in a smooth prior predictive distribution.

Figure A1: Example of the likelihood \( p(c|\theta_1, \theta_2, x) \) in equations A.1 and A.2 over the two classes \( c \in \{c_1, c_2\} \) for different values of \( x \) with model parameters \( \theta_1 = 5 \) and \( \theta_2 = 1 \) corresponding to the decision boundary.
Looking at the posterior: on the origin of uncertainty in neural-network classification

Figure A2: Prior class distribution given the uniform prior $p(\theta_1, \theta_2)$ on the model parameters.

Figure A3: Posterior predictive distribution given one observation of class $c_1$ at $x_1 = 2$ compared to the prior predictive distribution. The observation of class 1 to the right shifts the posterior in this direction.

Suppose we observe class $c_1$ at $x_1 = 2$, we can then calculate a posterior distribution for the model parameters

$$p(\theta_1, \theta_2 \{x_1, c_1\}) = \frac{p(c_1|x_1, \theta_1, \theta_2)p(\theta_1, \theta_2)}{p(c_1|x_1)}$$ \hspace{1cm} (A.4)

where we have used Bayes theorem to express the parameter posterior in terms of conditional probabilities that can be calculated explicitly.

With this posterior we calculate the posterior predictive distribution in equation (1) of the introduction, resulting in a slightly shifted distribution for class 1 in figure A3, compared to the class prior in figure A2.

Using the toy model we can now explicitly verify the relation between (7) and (8). The expected entropy difference in (7) is given by

$$I(\theta_1, \theta_2|x) = \int_{\theta_1} \int_{\theta_2} p(\theta_1, \theta_2) \log(p(\theta_1, \theta_2)) d\theta_1 d\theta_2$$

$$- \sum_{i=1,2} p(c_i|x) \int_{\theta_1} \int_{\theta_2} p(\theta_1, \theta_2 \{x_1, c_i\}) \log (p(\theta_1, \theta_2 \{x_1, c_i\})) d\theta_1 d\theta_2 \hspace{1cm} (A.5)$$
Looking at the posterior: on the origin of uncertainty in neural-network classification

Figure A4: Entropy difference $I(\theta_1, \theta_2| x)$ and $I(c|x)$ for the toy model quantifying the epistemic uncertainty for different $x$ using only the prior.

and the posterior predictive entropy difference in (8) is given by

$$I(c|x) = \sum_{i=1,2} p(c_i|x) \log(p(c_i|x))$$

$$- \int_{\theta_1} \int_{\theta_2} \left( \sum_{i=1,2} p(c_i|x, \theta_1, \theta_2) \log(p(c_i|x, \theta_1, \theta_2)) \right) p(\theta_1, \theta_2) \, d\theta_1 \, d\theta_2 \quad (A.6)$$

Note that in this case, where we compute the entropy difference when adding a single observation equation (A.6) only uses the prior distribution.

Numerically evaluating these expressions gives figure A4 where the two curves are indistinguishable, as expected.

Figure A4 shows that the epistemic uncertainty is largest close to the decision boundary of the prior. This can be understood intuitively by the fact that the model and parameter priors are such that adding observations of class 1 far to the left (or class 2 far to the right) does not add new information.

Continuing, we can perform the same calculation but instead add a new observation on top of the first one. Assuming independent observations the posterior now becomes

$$p(\theta_1, \theta_2 | (x_1, c_1), (x_2, c_2)) = \frac{p(c_1|x_1, \theta_1, \theta_2) \, p(c_2|x_2, \theta_1, \theta_2) \, p(\theta_1, \theta_2)}{p((x_1, c_1), (x_2, c_2))} \quad (A.7)$$

and carefully calculating the entropy differences now instead results in the epistemic uncertainty in figure A5. Note first that the two expressions are still in excellent agreement. The observation of class 1 at $x = 2$ is in tension with the prior which can be seen by the bi-modal epistemic uncertainty.

The epistemic uncertainty can be compared to the entropy of the posterior predictive distribution in figure A6 which peaks in the region between the prior decision boundary and the observed class 1 sample.
Figure A5: Entropy difference $I(\theta_1, \theta_2|x, \{x_1 = 2, c = 1\})$ and $I(c|x, \{x_1 = 2, c = 1\})$ for the toy model quantifying the epistemic uncertainty for different $x$ after a single observation $\{x = 2, c = 1\}$.

Figure A6: Entropy of the posterior predictive distribution using the prior $H(c|x)$ and after one observation $H(c|x, \{x_1 = 2, c = 1\})$ for the toy model quantifying aleatoric and epistemic uncertainty for different $x$.

References

[1] Jakob Gawlikowski, Cedrique Rovile Njieutchen Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. A survey of uncertainty in deep neural networks, 2021.

[2] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information Fusion, 76:243–297, 2021. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2021.05.008. URL https://www.sciencedirect.com/science/article/pii/S1566253521001081.

[3] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable Bayesian deep learning methods for robust computer vision. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pages 318–319, 2020.
REFERENCES

[4] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012. 2, 5

[5] David J. C. MacKay. Information-Based Objective Functions for Active Data Selection. *Neural Computation*, 4(4):590–604, 07 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.4.590. URL https://doi.org/10.1162/neco.1992.4.4.590. 2, 7

[6] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112, 2009. ISSN 0167-4730. doi: https://doi.org/10.1016/j.strusafe.2008.06.020. URL https://www.sciencedirect.com/science/article/pii/S0167473008000556. Risk Acceptance and Risk Communication. 2

[7] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *J. Artif. Int. Res.*, 4(1):129–145, mar 1996. ISSN 1076-9757. 2

[8] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are Bayesian neural network posteriors really like? In *International conference on machine learning*, pages 4629–4640. PMLR, 2021. 3, 5

[9] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017. 3, 5, 6

[10] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 3

[11] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 3, 5, 6

[12] Lara Hoffmann and Clemens Elster. Deep ensembles from a bayesian perspective. arXiv preprint arXiv:2105.13283, 2021. 3

[13] Ali Awad. Denoising images corrupted with impulse, Gaussian, or a mixture of impulse and Gaussian noise. *Engineering Science and Technology, an International Journal*, 22(3):746–753, 2019. ISSN 2215-0986. doi: https://doi.org/10.1016/j.jestch.2019.01.012. URL https://www.sciencedirect.com/science/article/pii/S2215098618321050. 3, 8

[14] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 3, 8

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3

[16] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings*
REFERENCES

of Machine Learning Research, pages 1184–1193. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/depeweg18a.html.

[17] Biraja Ghoshal and Allan Tucker. Estimating uncertainty and interpretability in deep learning for coronavirus (covid-19) detection, 2020. URL https://arxiv.org/abs/2003.10769.

[18] Biraja Ghoshal, Allan Tucker, Bal Sanghera, and Wai Lup Wong. Estimating uncertainty in deep learning for reporting confidence to clinicians in medical image segmentation and diseases detection. Computational Intelligence, 37(2):701–734, 2021. doi: https://doi.org/10.1111/coin.12411. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/coin.12411.

[19] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can You Trust Your Model’s Uncertainty? Evaluating Predictive Uncertainty under Dataset Shift. Curran Associates Inc., Red Hook, NY, USA, 2019.

[20] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019. URL https://arxiv.org/abs/1903.12261.

[21] Bernhard Mehlig. Machine learning with neural networks. Cambridge University Press, Cambridge, England, October 2021.

[22] Devinderjit Sivia and John Skilling. Data analysis: a Bayesian tutorial. OUP Oxford, 2006.

[23] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res., 15(1):1929–1958, jan 2014. ISSN 1532-4435.

[24] Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In Proceedings of the Twenty-First International Conference on Machine Learning, ICML ’04, page 78, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015435. URL https://doi.org/10.1145/1015330.1015435.

[25] Loic Le Folgoc, Vasileios Baltatzis, Sujal Desai, Anand Devaraj, Sam Ellis, Octavio E. Martinez Manzanera, Arjun Nair, Huaqi Qiu, Julia Schnabel, and Ben Glocker. Is MC dropout Bayesian?, 2021. URL https://arxiv.org/abs/2110.04286.

[26] Stanislav Fort and Stanislaw Jastrzebski. Large scale structure of neural network loss landscapes. Advances in Neural Information Processing Systems, 32, 2019.

[27] C. E. Shannon. A mathematical theory of communication. The Bell System Technical Journal, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

[28] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máthé Lengyel. Bayesian active learning for classification and preference learning. arXiv preprint arXiv:1112.5745, 2011.
[29] Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection, 2018. 7

[30] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep Bayesian active learning. Advances in neural information processing systems, 32:7026–7037, 2019. 8

[31] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 8

[32] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net, 2014. URL https://arxiv.org/abs/1412.6806. 9

[33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 10012–10022, October 2021. 9

[34] Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. Advances in Neural Information Processing Systems, 33:18237–18248, 2020. 17

[35] Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers Can Do Bayesian Inference. arXiv e-prints, art. arXiv:2112.10510, December 2021. 18