Artificial Intelligence Review
https://doi.org/10.1007/s10462-023-10562-9

A survey of uncertainty in deep neural networks

Jakob Gawlikowski[1,2] · Cedrique Rovile Njieutcheu Tassi[3,4] · Mohsin Ali[2,5] · Jongseok Lee[6] · Matthias Humt[3,6] · Jianxiang Feng[3,6] · Anna Kruspe[2] · Rudolph Triebel[3,6] · Peter Jung[2,4,7] · Ribana Roscher[2,8] · Muhammad Shahzad[2] · Wen Yang[9] · Richard Bamler[10] · Xiao Xiang Zhu[2]

Abstract
Over the last decade, neural networks have reached almost every field of science and become a crucial part of various real world applications. Due to the increasing spread, confidence in neural network predictions has become more and more important. However, basic neural networks do not deliver certainty estimates or suffer from over- or under-confidence, i.e. are badly calibrated. To overcome this, many researchers have been working on understanding and quantifying uncertainty in a neural network's prediction. As a result, different types and sources of uncertainty have been identified and various approaches to measure and quantify uncertainty in neural networks have been proposed. This work gives a comprehensive overview of uncertainty estimation in neural networks, reviews recent advances in the field, highlights current challenges, and identifies potential research opportunities. It is intended to give anyone interested in uncertainty estimation in neural networks a broad overview and introduction, without presupposing prior knowledge in this field. For that, a comprehensive introduction to the most crucial sources of uncertainty is given and their separation into reducible model uncertainty and irreducible data uncertainty is presented. The modeling of these uncertainties based on deterministic neural networks, Bayesian neural networks (BNNs), ensemble of neural networks, and test-time data augmentation approaches is introduced and different branches of these fields as well as the latest developments are discussed. For a practical application, we discuss different measures of uncertainty, approaches for calibrating neural networks, and give an overview of existing baselines and available implementations. Different examples from the wide spectrum of challenges in the fields of medical image analysis, robotics, and earth observation give an idea of the needs and challenges regarding uncertainties in the practical applications of neural networks. Additionally, the practical limitations of uncertainty quantification methods in neural networks for mission- and safety-critical real world applications are discussed and an outlook on the next steps towards a broader usage of such methods is given.

Keywords Bayesian deep neural networks · Ensembles · Test-time augmentation · Calibration · Uncertainty

Extended author information available on the last page of the article

Published online: 29 July 2023

1 Introduction

Within the last decade enormous advances on deep neural networks (DNNs) have been realized, encouraging their adaptation in a variety of research fields, where complex systems have to be modeled or understood, such as earth observation, medical image analysis, or robotics. Although DNNs have become attractive in high-risk fields such as medical image analysis (Nair et al. 2020; Roy et al. 2019; Seebock et al. 2020; LaBonte et al. 2019; Reinhold et al. 2020; Eggenreich et al. 2020) or autonomous vehicle control (Feng et al. 2018; Choi et al. 2019; Amini et al. 2018; Loquercio et al. 2020), their deployment in mission- and safety-critical real world applications remains limited. The main factors responsible for this limitation are

- the lack of expressiveness and transparency of a deep neural network's inference model, which makes it difficult to trust their outcomes (Roy et al. 2019),
- the inability to distinguish between in-domain and out-of-domain samples (Lee et al. 2018a; Mitros and Mac Namee 2019) and the sensitivity to domain shifts (Ovadia et al. 2019),
- the inability to provide reliable uncertainty estimates for a deep neural network's decision (Ayhan and Berens 2018) and frequently occurring overconfident predictions (Guo et al. 2017; Wilson and Izmailov 2020), and
- the sensitivity to adversarial attacks that make deep neural networks vulnerable for sabotage (Rawat et al. 2017; Serban et al. 2018; Smith and Gal 2018).

These factors are mainly based on an uncertainty already included in the data (data uncertainty) or a lack of knowledge of the neural network (model uncertainty). To overcome these limitations, it is essential to provide uncertainty estimates, such that uncertain predictions can be ignored or passed to human experts (Gal and Ghahramani 2016). Providing uncertainty estimates is not only important for safe decision-making in high-risk fields, but it is also crucial in fields where the data sources are highly inhomogeneous and labeled data is rare, such as in remote sensing (Rußwurm et al. 2020; Gawlikowski et al. 2022). Also for fields where uncertainties form a crucial part of the learning techniques, such as for active learning (Gal et al. 2017b; Chitta et al. 2018; Zeng et al. 2018; Nguyen et al. 2019) or reinforcement learning (Gal and Ghahramani 2016; Huang et al. 2019a; Kahn et al. 2017; Lütjens et al. 2019), uncertainty estimates are highly important.

In recent years, researchers have shown an increased interest in estimating uncertainty in DNNs (Blundell et al. 2015; Gal and Ghahramani 2016; Lakshminarayanan et al. 2017; Malinin and Gales 2018; Sensoy et al. 2018; Wu et al. 2019; Van Amersfoort et al. 2020; Ramalho and Miranda 2020). The most common way to estimate the uncertainty of a prediction (the predictive uncertainty) is based on separately modelling the uncertainty caused by the model (epistemic or model uncertainty) and the uncertainty caused by the data (aleatoric or data uncertainty). While the first one is reducible by improving the model which is learned by the DNN, the latter one is not reducible. The most important approaches for modeling this separation are Bayesian inference (Blundell et al. 2015; Gal and Ghahramani 2016; Mobiny et al. 2021; Amini et al. 2018; Krueger et al. 2017), ensemble approaches (Lakshminarayanan et al. 2017; Valdenegro-Toro 2019; Wen et al. 2019), test-time augmentation approaches (Shorten and Khoshgoftaar 2019; Wen et al. 2021a), or single deterministic networks containing explicit components to represent the model and the data uncertainty (Malinin and Gales 2018;

Sensoy et al. 2018; Malinin and Gales 2019; Raghu et al. 2019). However, estimating the predictive uncertainty is not sufficient for safe decision-making. It is also crucial to assure that the uncertainty estimates are reliable. To this end, the calibration property (the degree of reliability) of DNNs has been investigated and re-calibration methods have been proposed (Guo et al. 2017; Wenger et al. 2020; Zhang et al. 2020) to obtain reliable (well-calibrated) uncertainty estimates.

There are several works that give an introduction and overview of uncertainty in statistical modelling. Ghanem et al. (2017) published a handbook about uncertainty quantification, which includes a detailed and broad description of different concepts of uncertainty quantification, but without explicitly focusing on the application of neural networks. The theses of Gal (1998) and Kendall (2019) contain a good overview of Bayesian neural networks (BNNs), especially with the focus on the Monte Carlo (MC) Dropout approach and its application in computer vision tasks. The thesis of Malinin (2019) also contains a very good introduction and additional insights into Prior networks. Wang and Yeung (2016, 2020) contributed two surveys on Bayesian deep learning. They introduced a general framework and the conceptual description of the BNNs, followed by an updated presentation of Bayesian approaches for uncertainty quantification in neural networks with a special focus on recommender systems, topic models, and control. In Ståhl et al. (2020), an evaluation of uncertainty quantification in deep learning is given by presenting and comparing the uncertainty quantification based on the softmax output, the ensemble of networks, BNNs, and autoencoders on the MNIST data set. Regarding the practicability of uncertainty quantification approaches for real-life mission- and safety-critical applications, (Gustafsson et al. 2020) introduced a framework to test the robustness required in real-world computer vision applications and delivered a comparison of two popular approaches, namely MC Dropout and Ensemble methods. Hüllermeier and Waegeman (2021) presented the concepts of aleatoric and epistemic uncertainty in neural networks and discussed different concepts to model and quantify them. In contrast to this, (Abdar et al. 2021) presented an overview on uncertainty quantification methodologies in neural networks and provide an extensive list of references for different application fields and a discussion of open challenges.

In this work, we present an extensive overview over all concepts that have to be taken into account when working with uncertainty in neural networks while keeping the applicability of real world applications in mind. Our goal is to provide the reader with a clear thread from the sources of uncertainty to applications, where uncertainty estimations are needed. Furthermore, we point out the limitations of current approaches and discuss further challenges to be tackled in the future. For that, we provide a broad introduction and comparison of different approaches and fundamental concepts. The survey is mainly designed for people already familiar with deep learning concepts and who are planning to incorporate uncertainty estimation into their predictions. But also for people already familiar with the topic, this review provides a useful overview of the whole concept of uncertainty in neural networks and their applications in different fields. In summary, we comprehensively discuss

- Sources and types of uncertainty (Sect. 2),
- Recent studies and approaches for estimating uncertainty in DNNs (Sect. 3),
- Uncertainty measures and methods for assessing the quality and impact of uncertainty estimates (Sect. 4),
- Recent studies and approaches for calibrating DNNs (Sect. 5),

- An overview over frequently used evaluation data sets, available benchmarks and implementations[1] (Sect. 6),
- An overview over real-world applications using uncertainty estimates (Sect. 7),
- A discussion on current challenges and further directions of research in the future (Sect. 8).

The basic descriptions of uncertainty representations in neural networks are not problem specific and many of the proposed methods (e.g., BNNs or ensemble of neural networks) can be applied to many different types of problems such as classification, regression, or segmentation. If not stated differently, the presented methods are not limited to a specific type of problem. In order to get a deeper dive into explicit applications of the methods, we refer to the section on applications and to further readings in the referenced literature.

2 Uncertainty in deep neural networks

A neural network is a non-linear function f_θ parameterized by model parameters θ (i.e. the network weights) that maps from a measurable input set $\mathbb{X}$ to a measurable output set $\mathbb{Y}$, i.e.

$$f_\theta : \mathbb{X} \rightarrow \mathbb{Y} \qquad f_\theta(x) = y. \tag{1}$$

For a supervised setting, we further have a finite set of training data $\mathcal{D} \subseteq \mathbb{D} = \mathbb{X} \times \mathbb{Y}$ containing N data samples and corresponding targets, i.e.

$$\mathcal{D} = (\mathcal{X}, \mathcal{Y}) = \{x_n, y_n\}_{n=1}^{N} \subseteq \mathbb{D}\,. \tag{2}$$

For a new data sample $x^* \in \mathbb{X}$, a neural network trained on $\mathcal{D}$ can be used to predict a corresponding target $f_\theta(x^*) = y^*$. We consider four different steps from the raw information in the environment to a prediction by a neural network with quantified uncertainties, namely

1. the *data acquisition* process: The occurrence of some information in the environment (e.g. a bird's singing) and a measured observation of this information (e.g. an audio record).
2. the *DNN building* process: The design and training of a neural network.
3. the *applied inference* model: The model is applied for inference (e.g. a BNN or an ensemble of neural networks).
4. the *prediction's uncertainty* model: The modelling of the uncertainties caused by the neural network and/or by the data.

In practice, these four steps contain several potential sources of uncertainty and errors, which again affect the final prediction of a neural network. The five factors that we think are the most vital for the cause of uncertainty in a DNN's predictions are

- the variability in real world situations,
- the errors inherent to the measurement systems,
- the errors in the architecture specification of the DNN,

[1] The list of available implementations can be found in Sect. 6 as well as within an additional GitHub repository under https://github.com/JakobCode/UncertaintyInNeuralNetworks_Resources.

- the errors in the training procedure of the DNN,
- the errors caused by unknown data.

In the following, the four steps leading from raw information to uncertainty quantification on a DNN's prediction are described in more detail. Within this, we highlight the sources of uncertainty that are related to the single steps and explain how the uncertainties are propagated through the process. Finally, we introduce a model for the uncertainty on a neural network's prediction and introduce the main types of uncertainty considered in neural networks.

The goal of this section is to give an accountable idea of the uncertainties in neural networks. Hence, for the sake of simplicity, we only describe and discuss the mathematical properties, which are relevant for understanding the approaches and applying the methodology in different fields.

2.1 Data acquisition

In the context of supervised learning, the data acquisition describes the process where measurements x and target variables y are generated in order to represent a (real world) situation ω from some space Ω. In the real world, a realization of ω could for example be a bird, x a picture of this bird, and y a label stating '*bird*'. During the measurement, random noise can occur and information may get lost. We model this randomness in x by

$$x|\omega \sim p_{x|\omega}. \tag{3}$$

Equivalently, the corresponding target variable y is derived, where the description is either based on another measurement or is the result of a labeling process.[2] For both cases, the description can be affected by noise and errors and we state it as

$$y|\omega \sim p_{y|\omega}. \tag{4}$$

A neural network is trained on a finite data set of realizations of $x|\omega_i$ and $y|\omega_i$ based on N real world situations $\omega_1, \dots, \omega_N$,

$$\mathcal{D} = \{x_i, y_i\}_{i=1}^N . \tag{5}$$

When collecting the training data, two factors can cause uncertainty in a neural network trained on this data. First, the sample space Ω should be sufficiently covered by the training data $x_1, \dots, x_N$ for $\omega_1, \dots, \omega_N$. For that, one has to take into account that for a new sample x^* it in general holds that $x^* \neq x_i$ for all training situations x_i. Following, the target has to be estimated based on the trained neural network model, which directly leads to the first factor of uncertainty:

[2] In many cases one can model the labeling process as a mapping from $\mathbb{X}$ to $\mathbb{Y}$, e.g. for speech recognition or various computer vision tasks. For other tasks, such as earth observation, this is not always the case. Data is often labeled based on high-resolution data while low-resolution data is utilized for the prediction task.

Factor I: Variability in real world situations

Most real world environments are highly variable and almost constantly affected by changes. These changes affect parameters such as temperature, illumination, clutter, and physical objects' size and shape. Changes in the environment can also affect the expression of objects, such as plants after rain look very different from plants after a drought. When real world situations change compared to the training set, this is called a distribution shift. Neural networks are sensitive to distribution shifts, which can lead to significant changes in the performance of a neural network.

The second case is based on the measurement system, which has a direct effect on the correlation between the samples and the corresponding targets. The measurement system generates information x_i and y_i that describe ω_i but might not contain enough information to learn a direct mapping from x_i to y_i. This means that there might be highly different real world information ω_i and ω_j (e.g. city and forest) resulting in very similar corresponding measurements x_i and x_j (e.g. temperature) or similar corresponding targets y_i and y_j (e.g. label noise that labels both samples as forest). This directly leads to our second factor of uncertainty:

Factor II: Error and noise in measurement systems

The measurements themselves can be a source of uncertainty on the neural network's prediction. This can be caused by limited information in the measurements, such as the image resolution. Moreover, it can be caused by noise, for example, sensor noise, by motion, or mechanical stress leading to imprecise measures. Furthermore, false labeling is also a source of uncertainty that can be seen as an error or noise in the measurement system. It is referenced as label noise and affects the model by reducing the confidence on the true class prediction during training. Depending on the intensity, this type of noise and errors can be used to regularize the training process and to improve robustness and generalization (Goodfellow et al. 2016; Peterson et al. 2019; Lukasik et al. 2020).

2.2 Deep neural network design and training

The design of a DNN covers the explicit modeling of the neural network and its stochastic training process. The assumptions on the problem structure induced by the design and training of the neural network are called inductive bias (Battaglia et al. 2018). We summarize all decisions of the modeler on the network's structure (e.g. the number of parameters, the layers, the activation functions, etc.) and training process (e.g. optimization algorithm, regularization, augmentation, etc.) in a structure configuration s. The defined network structure gives the third factor of uncertainty in a neural network's predictions:

Factor III: Errors in the model structure

The structure of a neural network has a direct effect on its performance and therefore also on the uncertainty of its prediction. For instance, the number of parameters affects the memorization capacity, which can lead to under- or over-fitting on the training data. Regarding uncertainty in neural networks, it is known that deeper networks tend to be overconfident in their softmax output, meaning that they predict too much probability on the class with the highest probability score (Guo et al. 2017).

For a given network structure s and a training data set $\mathcal{D}$, the training of a neural network is a stochastic process and therefore the resulting neural network f_θ is based on a random variable,

$$\theta | D, s \sim p_{\theta|D,s}. \tag{6}$$

The process is stochastic due to random decisions as the order of the data, random initialization, or random regularization as augmentation or dropout. The loss landscape of a neural network is highly non-linear and the randomness in the training process in general leads to different local optima θ^* resulting in different models (Lakshminarayanan et al. 2017). Also, parameters such as batch size, learning rate, and the number of training epochs affect the training and result in different models. Depending on the underlying task these models can significantly differ in their predictions for single samples, even leading to a difference in the overall model performance. This sensitivity to the training process directly leads to the fourth factor for uncertainties in neural network predictions:

Factor IV: Errors in the training procedure

The training process of a neural network includes many parameters that have to be defined (batch size, optimizer, learning rate, stopping criteria, regularization, etc.), and also stochastic decisions within the training process (batch generation and weight initialization) take place. All these decisions affect the local optima and it is therefore very unlikely that two training processes deliver the same model parameterization. A training data set that suffers from imbalance or low coverage of single regions in the data distribution also introduces uncertainties on the network's learned parameters, as already described in the data acquisition. This might be softened by applying augmentation to increase the variety or by balancing the impact of single classes or regions on the loss function.

Since the training process is based on the given training data set $\mathcal{D}$, errors in the data acquisition process (e.g. label noise) can result in errors in the training process.

2.3 Inference

The inference describes the prediction of an output y^* for a new data sample x^* by the neural network. At this time, the network is trained for a specific task. Thus, samples that are not inputs for this task cause errors and are therefore also a source of uncertainty:

Factor V: Errors caused by unknown data

Especially in classification tasks, a neural network that is trained on samples derived from a world $\mathcal{W}_1$ can also be capable of processing samples derived from a completely different world $\mathcal{W}_2$. This is for example the case when a network trained on images of cats and dogs receives a sample showing a bird. Here, the source of uncertainty does not lie in the data acquisition process, since we assume a world to contain only feasible inputs for a prediction task. Even though the practical result might be equal to too much noise on a sensor or complete failure of a sensor, the data considered here represents a valid sample, but for a different task or domain.

2.4 Predictive uncertainty model

As a modeller, one is mainly interested in the uncertainty that is propagated onto a prediction y^*, the so-called *predictive uncertainty*. Within the data acquisition model, the probability distribution for a prediction y^* based on some sample x^* is given by

$$p(y^*|x^*) = \int_\Omega p(y^*|\omega)p(\omega|x^*)d\omega \tag{7}$$

and a maximum a posteriori (MAP) estimation is given by

$$y^* = \arg\max_y p(y|x^*). \tag{8}$$

Since the modeling is based on the unavailable latent variable ω, one takes an approximative representation based on a sampled training data set $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ containing N samples and corresponding targets. The distribution and MAP estimator in (7) and (8) for a new sample x^* are then predicted based on the known examples by

$$p(y^*|x^*) = \int_D p(y^*|\mathcal{D}, x^*) \tag{9}$$

and

$$y^* = \arg\max_y p(y|\mathcal{D}, x^*). \tag{10}$$

In general, the distribution given in (9) is unknown and can only be estimated based on the given data in D. For this estimation, neural networks form a very powerful tool for many tasks and applications.

The prediction of a neural network is subject to both model-dependent and input data-dependent errors, and therefore the predictive uncertainty associated with y^* is in general separated into *data uncertainty* [also statistical or aleatoric uncertainty (Hüllermeier and Waegeman 2021)] and *model uncertainty* [also systemic or epistemic uncertainty (Hüllermeier and Waegeman 2021)]. Depending on the underlying approach, additional explicit modeling of *distributional uncertainty* (Malinin and Gales 2018) is used to model the uncertainty, which is caused by examples from a region not covered by the training data. Figure 1 illustrates the described types of uncertainty for regression and classification tasks.

2.4.1 Model- and data uncertainty

The model uncertainty covers the uncertainty that is caused by shortcomings in the model, either by errors in the training procedure, an insufficient model structure, or lack of knowledge due to unknown samples or a bad coverage of the training data set. In contrast to this, data uncertainty is related to uncertainty that directly stems from the data. Data uncertainty is caused by information loss when representing the real world within a data sample and represents the distribution stated in (7). For example, in regression tasks, noise in the input and target measurements causes data uncertainty that the network cannot learn to correct. In classification tasks, samples that do not contain enough information in order to identify one class with 100% certainty cause data uncertainty on the prediction. The information loss is a result of the measurement system, e.g. by representing real world information by image pixels with a specific resolution, or by errors in the labelling process.

For the five presented factors for uncertainties on a neural network's prediction, this means the following: Only factor 2 represents a source of aleatoric uncertainty since it causes insufficient data that make a certain prediction not possible. For all other factors, the source of uncertainty lies in the experimental setup and is related to epistemic uncertainty. The uncertainty induced by Factor I is a result of the insufficient coverage of the data distribution in the training data. Factor III and Factor IV clearly represent shortcomings in the training and the modelling of the network. Factor V is also related to epistemic uncertainty since the data itself might be fine but the unknown domain is

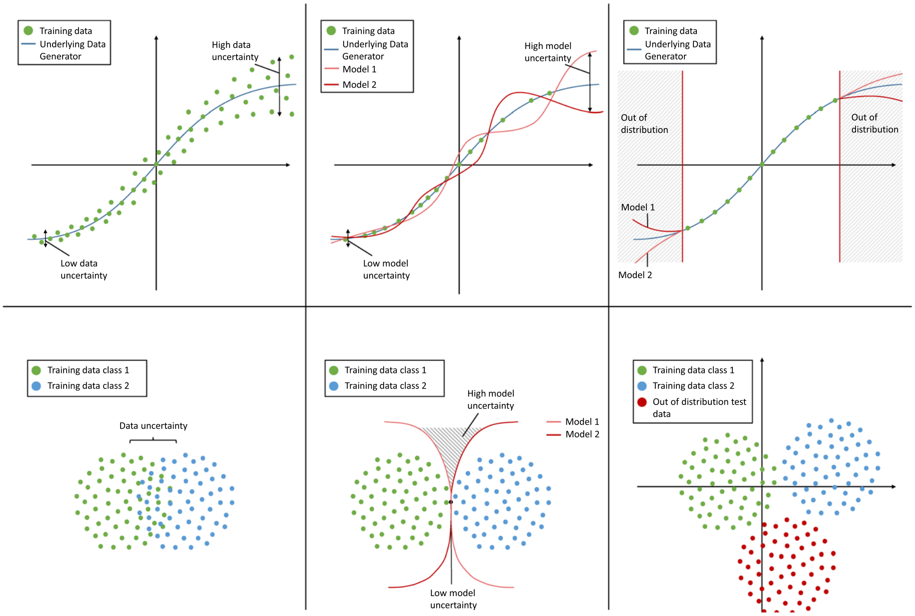


Fig. 1 Visualization of the data, the model, and the distributional uncertainty for classification and regression models

not included in the modelling and hence the model lacks knowledge of how to handle this data. Figure 2 illustrates the discussed stages of a neural network pipeline employed in a remote sensing classification task, along with the diverse sources of uncertainties that impact the resulting predictions.

While model uncertainty can be (theoretically) reduced by improving the architecture, the learning process, or the training data set, the data uncertainties cannot be explained away (Kendall and Gal 2017). Therefore, DNNs that are capable of handling uncertain inputs and that are able to remove or quantify the model uncertainty and give a correct prediction of the data uncertainty are of paramount importance for a variety of real world mission- and safety-critical applications.

The Bayesian framework offers a practical tool to reason about uncertainty in deep learning (Gal and Ghahramani 2015). In Bayesian modeling, the model uncertainty is formalized as a probability distribution over the model parameters θ, while the data uncertainty is formalized as a probability distribution over the model outputs y^*, given a parameterized model f_θ. The distribution over a prediction y^*, the predictive distribution, is then given by

$$p(y^*|x^*, D) = \int \underbrace{p(y^*|x^*, \theta)}_{\text{Data}} \underbrace{p(\theta|D)}_{\text{Model}} \, d\theta \, . \tag{11}$$

The term $p(\theta|D)$ is referenced as posterior distribution on the model parameters and describes the uncertainty on the model parameters given a training data set D. The posterior distribution is in general not tractable. While ensemble approaches seek to approximate it by learning several different parameter settings and averaging over the resulting

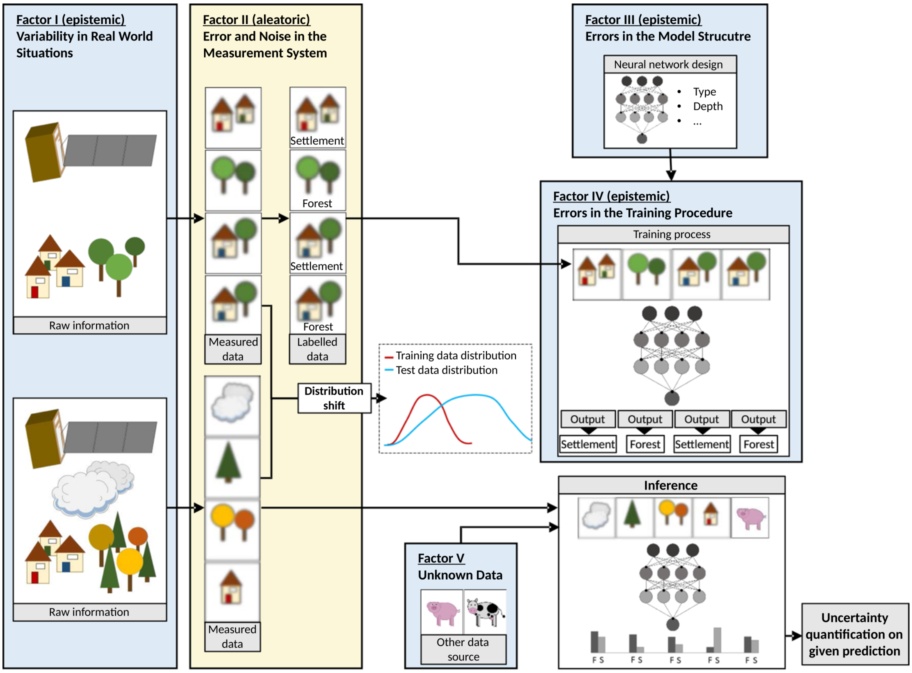


Fig. 2 The illustration shows the different steps of a neural network pipeline, based on the earth observation example of land cover classification (here settlement and forest) based on optical images. The different factors that affect the predictive uncertainty are highlighted in the boxes. Factor I is shown as changing environments by cloud-covered trees, and different types and colors of trees. Factor II is shown by insufficient measurements, that can not directly be used to separate between settlement and forest and by label noise. In practice, the resolution of such images can be low and which would also be part of Factor II. Factor III and Factor IV represent the uncertainties caused by the network structure and the stochastic training process, respectively. Factor V in contrast is represented by feeding the trained network with unknown types of images, namely cows and pigs

models (Lakshminarayanan et al. 2017), Bayesian inference reformulates it using Bayes Theorem (Bishop and Nasrabadi 2006)

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}. \tag{12}$$

The term $p(\theta)$ is called the prior distribution on the model parameters since it does not take any information but the general knowledge on θ into account. The term $p(D|\theta)$ represents the likelihood that the data in D is a realization of the distribution predicted by a model parameterized with θ. Many loss functions are motivated by or can be related to the likelihood function. Loss functions that seek to maximize the log-likelihood (for an assumed distribution) are for example the cross-entropy or the mean squared error (Ritter et al. 2018).

Even with the reformulation given in (12), the predictive distribution given in (11) is still intractable. To overcome this, several different ways to approximate the predictive distribution were proposed. A broad overview of the different concepts and some specific approaches is presented in Sect. 3.

2.4.2 Distributional uncertainty

Depending on the approaches that are used to quantify the uncertainty in y^*, the formulation of the predictive distribution might be further separated into data, distributional, and model parts (Malinin and Gales 2018):

$$p(y^*|x^*, D) = \int\int \underbrace{p(y|\mu)}_{\text{Data}} \underbrace{p(\mu|x^*, \theta)}_{\text{Distributional}} \underbrace{p(\theta|D)}_{\text{Model}} d\mu d\theta. \tag{13}$$

The distributional part in (13) represents the uncertainty on the actual network output, e.g. for classification tasks this might be a Dirichlet distribution, which is a distribution over the categorical distribution given by the softmax output. Modeled this way, distributional uncertainty refers to uncertainty that is caused by a change in the input-data distribution, while model uncertainty refers to uncertainty that is caused by the process of building and training the DNN. As modeled in (13), the model uncertainty affects the estimation of the distributional uncertainty, which affects the estimation of the data uncertainty.

While most methods presented in this paper only distinguish between model and data uncertainty, approaches specialized on out-of-distribution (OOD) detection often explicitly aim at representing the distributional uncertainty (Malinin and Gales 2018; Nandy et al. 2020). A more detailed presentation of different approaches for quantifying uncertainties in neural networks is given in Sect. 3. In Sect. 4, different measures for measuring the different types of uncertainty are presented.

2.5 Uncertainty classification

On the basis of the input data domain, the predictive uncertainty can also be classified into three main classes:

- *In-domain uncertainty* (Ashukha et al. 2019)

  In-domain uncertainty represents the uncertainty related to an input drawn from a data distribution assumed to be equal to the training data distribution. The in-domain uncertainty stems from the inability of the deep neural network to explain an in-domain sample due to a lack of in-domain knowledge. From a modeler's point of view, in-domain uncertainty is caused by design errors (model uncertainty) and the complexity of the problem at hand (data uncertainty). Depending on the source of the in-domain uncertainty, it might be reduced by increasing the quality of the training data (set) or the training process (Hüllermeier and Waegeman 2021).
- *Domain-shift uncertainty* (Ovadia et al. 2019)

  Domain-shift uncertainty denotes the uncertainty related to an input drawn from a shifted version of the training distribution. The distribution shift results from insufficient coverage by the training data and the variability inherent to real world situations. A domain-shift might increase the uncertainty due to the inability of the DNN to explain the domain-shift sample on the basis of the seen samples at training time. Some errors causing domain shift uncertainty can be modeled and can therefore be reduced. For example, occluded samples can be learned by the deep neural network to reduce domain shift uncertainty caused by occlusions (DeVries and Taylor 2017). However, it is difficult if not impossible to model all errors causing domain shift uncertainty, e.g., motion noise (Kendall and Gal 2017). From a modeler's point of view, domain-shift uncertainty is caused by external or environmental factors but can be reduced by covering the shifted domain in the training data set.

- *Out-of-domain uncertainty* (Hendrycks and Gimpel 2017; Liang et al. 2018b; Shafaei et al. 2019; Mundt et al. 2019)

  Out-of-domain uncertainty represents the uncertainty related to an input drawn from the subspace of unknown data. The distribution of unknown data is different and far from the training distribution. While a DNN can extract in-domain knowledge from domain-shift samples, it cannot extract in-domain knowledge from out-of-domain samples. For example, when domain-shift uncertainty describes phenomena like a blurred picture of a dog, out-of-domain uncertainty describes the case when a network that learned to classify cats and dogs is asked to predict a bird. The out-of-domain uncertainty stems from the inability of the DNN to explain an out-of-domain sample due to its lack of out-of-domain knowledge. From a modeler's point of view, out-of-domain uncertainty is caused by input samples, where the network is not meant to give a prediction for or by insufficient training data.

Since the model uncertainty captures what the DNN does not know due to a lack of in-domain or out-of-domain knowledge, it captures all, in-domain, domain-shift, and out-of-domain uncertainties. In contrast, the data uncertainty captures in-domain uncertainty that is caused by the nature of the data the network is trained on, for example, overlapping samples and systematic label noise.

3 Uncertainty estimation

As described in Sect. 2, several factors may cause model and data uncertainty and affect a DNN's prediction. This variety of sources of uncertainty makes the complete exclusion of uncertainties in a neural network impossible for almost all applications. Especially in practical applications employing real world data, the training data is only a subset of all possible input data, which means that a miss-match between the DNN domain and the unknown actual data domain is often unavoidable. However, an exact representation of the uncertainty of a DNN prediction is also not possible to compute, since the different uncertainties can in general not be modeled accurately and are most often even unknown. Therefore, methods for estimating uncertainty in a DNN prediction is a popular and vital field of research. The data uncertainty part is normally represented in the prediction, e.g. in the softmax output of a classification network or in the explicit prediction of a standard deviation in a regression network (Kendall and Gal 2017). In contrast to this, several different approaches which model the model uncertainty and seek to separate it from the data uncertainty in order to receive an accurate representation of the data uncertainty were introduced (Kendall and Gal 2017; Malinin and Gales 2018; Lakshminarayanan et al. 2017).

In general, the methods for estimating the uncertainty can be split into four different types based on the number (single or multiple) and the nature (deterministic or stochastic) of the used DNNs.

- *Single deterministic methods* give the prediction based on one single forward pass within a deterministic network. The uncertainty quantification is either derived by using additional (external) methods or is directly predicted by the network.

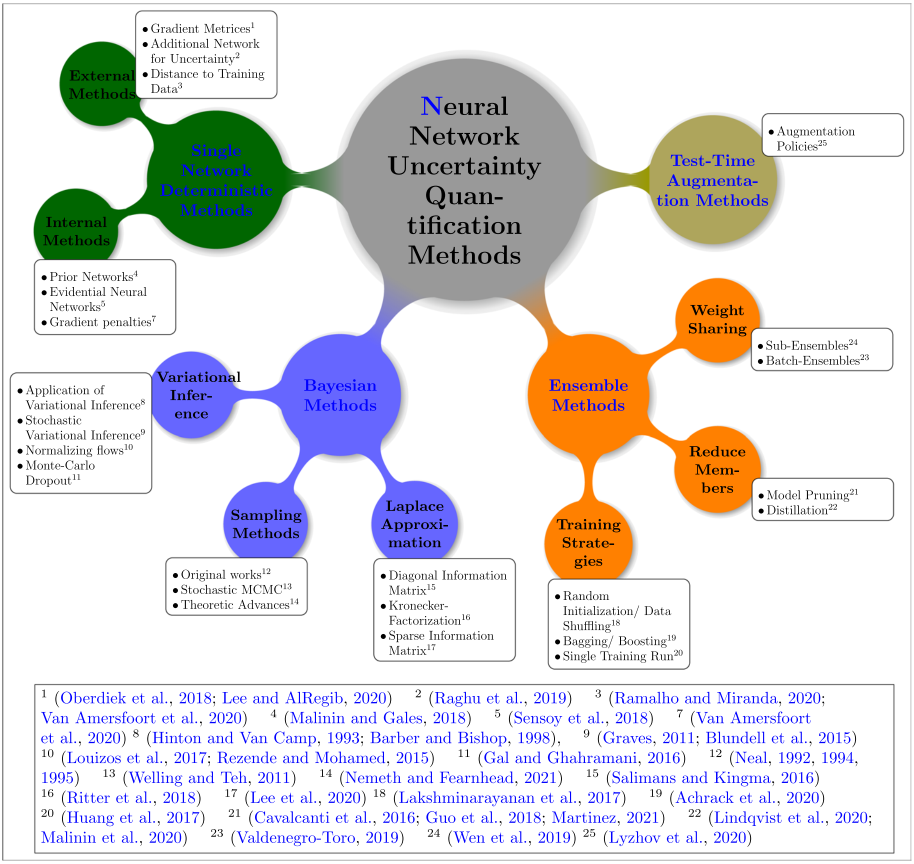

Neural Network Uncertainty Quantification Methods
Single Network Deterministic Methods
External Methods
• Gradient Metrices1
• Additional Network for Uncertainty2
• Distance to Training Data3
Internal Methods
• Prior Networks4
• Evidential Neural Networks5
• Gradient penalties7
Test-Time Augmentation Methods
• Augmentation Policies25
Bayesian Methods
Variational Inference
• Application of Variational Inference8
• Stochastic Variational Inference9
• Normalizing flows10
• Monte-Carlo Dropout11
Sampling Methods
• Original works12
• Stochastic MCMC13
• Theoretic Advances14
Laplace Approximation
• Diagonal Information Matrix15
• Kronecker-Factorization16
• Sparse Information Matrix17
Ensemble Methods
Weight Sharing
• Sub-Ensembles24
• Batch-Ensembles23
Reduce Members
• Model Pruning21
• Distillation22
Training Strategies
• Random Initialization/ Data Shuffling18
• Bagging/ Boosting19
• Single Training Run20
1 (Oberdiek et al., 2018; Lee and AlRegib, 2020) 2 (Raghu et al., 2019) 3 (Ramalho and Miranda, 2020; Van Amersfoort et al., 2020) 4 (Malinin and Gales, 2018) 5 (Sensoy et al., 2018) 7 (Van Amersfoort et al., 2020) 8 (Hinton and Van Camp, 1993; Barber and Bishop, 1998), 9 (Graves, 2011; Blundell et al., 2015) 10 (Louizos et al., 2017; Rezende and Mohamed, 2015) 11 (Gal and Ghahramani, 2016) 12 (Neal, 1992, 1994, 1995) 13 (Welling and Teh, 2011) 14 (Nemeth and Fearnhead, 2021) 15 (Salimans and Kingma, 2016) 16 (Ritter et al., 2018) 17 (Lee et al., 2020) 18 (Lakshminarayanan et al., 2017) 19 (Achrack et al., 2020) 20 (Huang et al., 2017) 21 (Cavalcanti et al., 2016; Guo et al., 2018; Martinez, 2021) 22 (Lindqvist et al., 2020; Malinin et al., 2020) 23 (Valdenegro-Toro, 2019) 24 (Wen et al., 2019) 25 (Lyzhov et al., 2020)

Fig. 3 Visualization of the four different types of uncertainty quantification methods presented in this paper

- *Bayesian methods* cover all kinds of stochastic DNNs, i.e. DNNs where two forward passes of the same sample generally lead to different results.
- *Ensemble methods* combine the predictions of several different deterministic networks at inference.
- *Test-time augmentation methods* give the prediction based on one single deterministic network but augment the input data at test-time in order to generate several predictions that are used to evaluate the certainty of the prediction.

In the following, the main ideas and further extensions of the four types are presented and their main properties are discussed. In Fig. 3, an overview of the different types and methods is given. In Fig. 4, the different underlying principles that are used to differentiate between the different types of methods are presented. Table 1 summarizes the main properties of the methods presented in this work, such as complexity, computational effort, memory consumption, flexibility, and others.

Table 1 An overview of the four general methods presented in this paper, namely Bayesian neural networks, ensembles, single deterministic neural networks, and test-time data augmentation

	Single deterministic networks	Bayesian methods	Ensemble methods	Test-time data augmentation
Description	Approaches that derive the prediction and predictive uncertainty based on a single deterministic forward pass	Model parameters are explicitly modeled as random variables. For a single forward pass the parameters are sampled from this distribution. Therefore, the prediction is stochastic and each prediction is based on different model weights	The prediction and uncertainty quantification at inference is based on the individual predictions received from multiple different networks. A variety among the single individual models' representations is crucial	The prediction and uncertainty quantification at inference is based on a single model and the combination of several predictions received from forward passes of different augmented versions of the original input sample
Description of Model Uncertainties	No	Yes	No	No
Need changes on pre-trained networks	Depends on the method	In general, yes, but also depends on the training strategy and the used approach	Yes (retrain several times)	No
Sensitivity to initialization and parameters of training process	High (in general)	Low (Usage of uninformative priors possible)	Low	Low
Number of networks trained	1	1	Several	1
Computational effort during training	Low	High	High	Low
Memory consumption at training	Low	Low	High	Low
Number of inputs per prediction	1	1	1	Several
Forward passes per prediction	1	Several	Several	Several
Evaluated modes	Single	Single	Multiple	Single
Computational effort during inference	Low (One forward pass, possibly some minor additional effort for uncertainty quantification)	High (sampling is either needed for explicit approach or for the approximation of intractable formulas)	High (Several models need to be evaluated)	High (Several augmentations and forward passes are performed)
Memory consumption at inference	Low	Low	High	Low

Table 1 (continued)

	Single deterministic networks	Bayesian methods	Ensemble methods	Test-time data augmentation
Application Scenarios	Check Table 2 for a more fine-grained presentation	Check Table 3 for a more fine-grained presentation	Medical: (Scalia et al. 2020; Dusenberry et al. 2020), Remote Sensing: (Ruzicka et al. 2020), Machinery Fault Detection: (Han and Li 2022)	Cell Segm.: (Moshkov et al. 2020), Brain Tumor Segm.: (Wang et al. 2018a), Remote Sensing: (Nalepa et al. 2019)

The labels high and low are given relative to the other approaches and based on the general idea

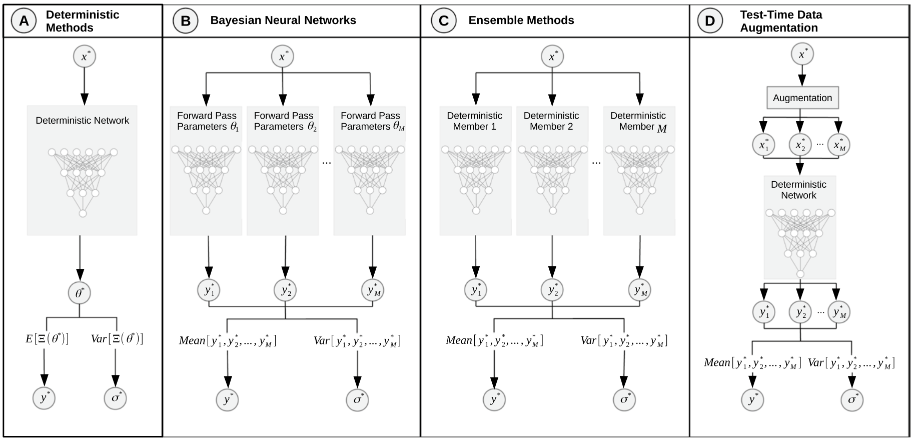


Fig. 4 A visualization of the basic principles of uncertainty modeling of the four presented general types of uncertainty prediction in neural networks. For a given input sample x^* each approach delivers a prediction y^*, a representation of model uncertainty σ_{model} and a value of data uncertainty σ_{data}. **A** Single deterministic model, **B** Bayesian neural network, **C** ensemble approach, and **D** test-time data augmentation. The mean and the standard deviation are only used to keep the visualization simple. In practice, other methods could be utilized. For the deterministic approaches the idea of predicting the parameters of a probability distribution Ξ is visualized, other approaches which base on tools additional to the prediction network are not visualized here

3.1 Single deterministic methods

For deterministic neural networks, the parameters are deterministic and each repetition of a forward pass delivers the same result. With single deterministic network methods for uncertainty quantification, we summarize all approaches where the uncertainty on a prediction y^* is computed based on one single forward pass within a deterministic network. In the literature, several such approaches can be found. They can be roughly categorized into approaches where one single network is explicitly modeled and trained in order to quantify uncertainties (Sensoy et al. 2018; Malinin and Gales 2018; Możejko et al. 2018; Nandy et al. 2020; Oala et al. 2020) and approaches that use additional components in order to give an uncertainty estimate on the prediction of a network (Raghu et al. 2019; Ramalho and Miranda 2020; Oberdiek et al. 2018; Lee and AlRegib 2020). While for the first type, the uncertainty quantification affects the training procedure and the predictions of the network, the latter type is in general applied to already trained networks. Since trained networks are not modified by such methods, they have no effect on the network's predictions. In the following, we call these two types *internal* and *external* uncertainty quantification approaches.

3.1.1 Internal uncertainty quantification approaches

Many of the internal uncertainty quantification approaches followed the idea of predicting the parameters of a distribution over the predictions instead of a direct pointwise maximum-a-posteriori estimation. Often, the loss function of such networks takes the expected divergence between the true distribution and the predicted distribution into account e.g., in Malinin and Gales (2018), Malinin and Gales (2019). The distribution over the outputs can be interpreted as a quantification of the model uncertainty (see

Table 2 Overview of the properties of internal and external deterministic single network methods

	Internal methods	External methods
Description	Estimate uncertainty using one evaluation of a single network without external components	Estimate uncertainty using one evaluation of the network while relying on additional external components
Implementation effort	Relatively low, but depends on the explicit approach, often only loss and network output have to be fixed	Relatively low, but depends on the explicit approach
Application on already trained networks possible	No	Yes
Separated prediction and uncertainty estimation	No	Yes
Application scenarios	Tumor Segm. (Monteiro et al. 2020), Molecular Property Prediction (Soleimany et al. 2021), Remote Sensing (Gawlikowski et al. 2022)	Eye Disease Detection (Raghu et al. 2019)

For a comparison of single deterministic network approaches with Bayesian, ensemble, and test-time augmentation methods, see Table 1

Sect. 2), trying to emulate the behavior of Bayesian modeling of the network parameters (Nandy et al. 2020). The prediction is then given as the expected value of the predicted distribution.

For classification tasks, the output in general represents class probabilities. These probabilities are a result of applying the softmax function

$$\begin{aligned} \text{softmax} &: \mathbb{R}^K \to \left\{ z \in \mathbb{R}^K | z_i \geq 0, \sum_{k=1}^{K} z_k = 1 \right\} \\ \text{softmax}(z)_j &= \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)} \end{aligned} \tag{14}$$

for multiclass settings and the sigmoid function

$$\begin{aligned} \text{sigmoid} &: \mathbb{R} \to [0, 1] \\ \text{sigmoid}(z) &= \frac{1}{1 + \exp(-z)} \end{aligned} \tag{15}$$

for binary classification tasks on the logits z. These probabilities can be already interpreted as a prediction of the data uncertainty. However, it is widely discussed that neural networks are often over-confident and the softmax output is often poorly calibrated, leading to inaccurate uncertainty estimates (Vasudevan et al. 2019; Hendrycks and Gimpel 2017; Sensoy et al. 2018; Możejko et al. 2018). Furthermore, the softmax output cannot be associated with model uncertainty. But without explicitly taking the model uncertainty into account, out-of-distribution samples could lead to outputs that certify a false confidence. For example, a network trained on cats and dogs will very likely not result in 50% dog and 50% cat when it is fed with the image of a bird. This is, because the network extracts features from the image and even though the features do not fit to the cat class, they might fit even less to the dog class. As a result, the network puts more probability on cat. Furthermore, it was shown that the combination of rectified linear unit (ReLu) networks and the softmax output leads to settings where the network becomes more and more confident as the distance between an out-of-distribution sample and the learned training set becomes larger (Hein et al. 2019). Figure 5 shows an example where the rotation of a digit from MNIST leads to false predictions with high softmax values.

This phenomenon is described and further investigated by Hein et al. (2019) who proposed a method to avoid this behaviour, based on enforcing a uniform predictive distribution far away from the training data.

Several other classification approaches (Sensoy et al. 2018; Malinin and Gales 2018, 2019; Nandy et al. 2020) followed a similar idea of taking the logit magnitude into account, but making use of the Dirichlet distribution. The Dirichlet distribution is the conjugate prior of the categorical distribution and hence can be interpreted as a distribution over categorical distributions. The density of the Dirichlet distribution is defined by

$$\text{Dir}(\mu|\alpha) = \frac{\Gamma(\alpha_0)}{\prod_{c=1}^{K} \Gamma(\alpha_c)} \prod_{c=1}^{K} \mu_c^{\alpha_c - 1}, \quad \alpha_c > 0,\ \alpha_0 = \sum_{c=1}^{K} \alpha_c \quad ,$$

where Γ is the gamma function, $\alpha_1, \ldots, \alpha_K$ are called the concentration parameters, and the scalar α_0 is the precision of the distribution. In practice, the concentrations $\alpha_1, \ldots, \alpha_K$ are derived by applying a strictly positive transformation, such as the exponential function,

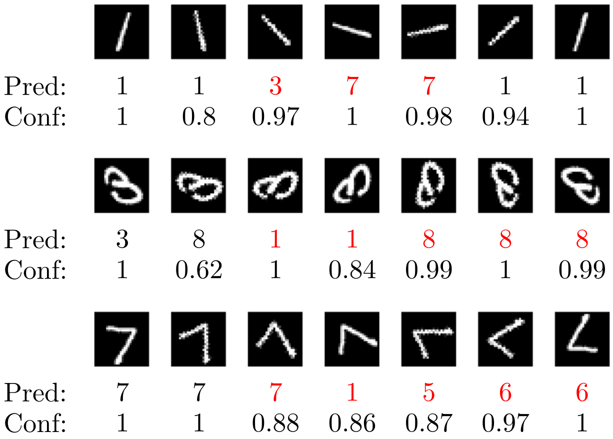
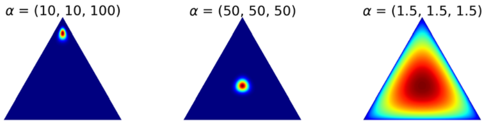


Fig. 5 Predictions received from a LeNet network trained on MNIST's handwritten digits from 0 to 9 and evaluated on different rotations of test samples. One can clearly see, that for some rotations the network gives high confidence on the false class due to confusion (e.g.: 3 is confused with 8) or representations not seen at training. These examples represent a simple case of how a basic classification network can lead to overconfident wrong predictions under data distribution shifts



(a) Low uncertainty (b) Data uncertainty (c) Out-of-distribution

Fig. 6 The desired behaviors of a Dirichlet distribution over categorical distributions. The visualizations show three Dirichlet distributions over three classes. Each node of the simplex represents one class. In **a** the sharp Dirichlet distribution with its expectation close to the upper node represents a certain prediction of a categorical distribution. In **b** the sharp Dirichlet distribution in the center of the simplex represents high data uncertainty but low distributional uncertainty. In **c** the flat Dirichlet distribution indicates high distributional uncertainty

to the logit values. As visualized in Fig. 6, a higher concentration value leads to a sharper Dirichlet distribution.

The set of all class probabilities of a categorical distribution over k classes is equivalent to a $k-1$-dimensional standard or probability simplex. Each node of this simplex represents a probability vector with the full probability mass on one class and each convex combination of the nodes represents a categorical distribution with the probability mass distributed over multiple classes. Malinin and Gales (2018) argued that a high model uncertainty should lead to a lower precision value and therefore to a flat distribution over the whole simplex since the network is not familiar with the data. In contrast to this, data uncertainty should be represented by a sharper but also centered distribution, since the network can handle the data, but cannot give a clear class preference. In Fig. 6 the different desired behaviors are shown.

The Dirichlet distribution is utilized in several approaches as *Dirichlet Prior Networks* (Malinin and Gales 2018; Tsiligkaridis 2021b) and *Evidential Neural Networks* (Sensoy et al. 2018). Both of these network types output the parameters of a Dirichlet distribution from which the categorical distribution describing the class probabilities can be derived. The general idea of prior networks (Malinin and Gales 2018) is already described above and is visualized in Fig. 6. Prior networks are trained in a multi-task way with the goal of minimizing the expected Kullback–Leibler (KL) divergence between the predictions of in-distribution data and a sharp Dirichlet distribution and between a flat Dirichlet distribution and the predictions of out-of-distribution data (Malinin and Gales 2018). Besides the main motivation of a better separation between in-distribution and OOD samples, these approaches also improve the separation between the confidence of correct and incorrect predictions, as was shown by Tsiligkaridis (2021a). As a follow-up, (Malinin and Gales 2019) discussed that for the case that the data uncertainty is high, the forward definition of the KL-divergence can lead to an undesirable multi-model target distribution. In order to avoid this, they reformulated the loss using the reverse KL divergence. The experiments showed improved results in the uncertainty estimation as well as for the adversarial robustness. Tsiligkaridis (2021b) extended the Dirichlet network approach by a new loss function that aims at minimizing an upper bound on the expected error based on the $\mathcal{L}_\infty$-norm, i.e. optimizing an expected worst-case upper bound. Wu et al. (2019) argued that using a mixture of Dirichlet distributions gives much more flexibility in approximating the posterior distribution. Therefore, an approach where the network predicts the parameters for a mixture of K Dirichlet distributions was suggested. For this, the network logits represent the parameters for M Dirichlet distributions and additionally M weights $\omega_i, i = 1, .., M$ with the constraint $\sum_{i=1}^{M} \omega_i = 1$ are optimized. Nandy et al. (2020) analytically showed that for in-domain samples with high data uncertainty, the Dirichlet distribution predicted for a false prediction is often flatter than for a correct prediction. They argued that this makes it harder to differentiate between in- and out-of-distribution predictions and suggested a regularization term for maximizing the gap between in- and out-of-distribution samples.

Evidential neural networks (Sensoy et al. 2018) also optimize the parameterization of a single Dirichlet network. The loss formulation is derived by using subjective logic and interpreting the logits as multinomial opinions or beliefs, as introduced in Evidence or Dempster-Shafer theory (Dempster 1968). Evidential neural networks set the total amount of evidence in relation to the number of classes and conclude a value of uncertainty from this, i.e. receiving an additional "I don't know class". The loss is formulated as the expected value of a basic loss, such as categorical cross entropy, with respect to a Dirichlet distribution parameterized by the logits. Additionally, a regularization term is added, encouraging the network predict the "I don't know state" if no evidence for an improvement in the data fit is found. Zhao et al. (2019) extended this idea by differentiating between vacuity and dissonance in the collected evidence in order to better separate in- and out-of-distribution samples. For that, two explicit data sets containing overlapping classes and out-of-distribution samples are needed to learn a regularization term. Amini et al. (2020) transferred the idea of evidential neural networks from classification tasks to regression tasks by learning the parameters of an evidential normal inverse gamma distribution over an underlying Normal distribution. Charpentier et al. (2020) avoided the need of OOD data for the training process by using normalizing flows to learn a distribution over a latent space for each class. A new input sample is projected onto this latent space and a Dirichlet distribution is parameterized based on the class-wise densities of the received latent point.

Besides the Dirichlet distribution based approaches described above, several other internal approaches exist. In Liang et al. (2018b), a relatively simple approach based on small perturbations on the training input data and the temperature scaling calibration is presented leading to efficient differentiation of in- and out-of-distribution samples. Możejko et al.

(2018) made use of the inhibited softmax function. It contains an artificial and constant logit that makes the absolute magnitude of the single logits more determined in the softmax output. Van Amersfoort et al. (2020) showed that Radial Basis Function (RBF) networks can be used to achieve competitive results in accuracy and very good results regarding uncertainty estimation. RBF networks learn a linear transformation on the logits and classify inputs based on the distance between the transformed logits and the learned class centroids. In Van Amersfoort et al. (2020), a scaled exponentiated L_2 distance was used. The data uncertainty can be directly derived from the distances between the centroids. By including penalties on the Jacobian matrix in the loss function, the network was trained to be more sensitive to changes in the input space. As a result, the method reached good performance on out-of-distribution detection. In several tests, the approach was compared to a five members deep ensemble (Lakshminarayanan et al. 2017) and it was shown that this single network approach performs at least equivalently well on detecting out-of-distribution samples and improves the true-positive rate.

For regression tasks, (Oala et al. 2020) introduced an uncertainty score based on the lower and upper bound output of an interval neural network. The interval neural network has the same structure as the underlying deterministic neural network and is initialized with the deterministic network's weights. In contrast to Gaussian representations of uncertainty given by a standard deviation, this approach can give non-symmetric values of uncertainty. Furthermore, the approach is found to be more robust in the presence of noise. Tagasovska and Lopez-Paz (2019) presented an approach to estimate data and model uncertainty. A simultaneous quantile regression loss function was introduced in order to generate well-calibrated prediction intervals for the data uncertainty. The model uncertainty is quantified based on a mapping from the training data to zero, based on so-called Orthonormal Certificates. The aim was that out-of-distribution samples, where the model is uncertain, are mapped to a non-zero value and thus can be recognized. Kawashima et al. (2021) introduced a method that computes virtual residuals in the training samples of a regression task based on a cross-validation like pre-training step. With original training data expanded by the information of these residuals, the actual predictor is trained to give a prediction and a value of certainty. The experiments indicated that the virtual residuals represent a promising tool in order to avoid overconfident network predictions.

3.1.2 External uncertainty quantification approaches

External uncertainty quantification approaches do not affect the models' predictions, since the evaluation of the uncertainty is separated from the underlying prediction task. Furthermore, several external approaches can be applied to already trained networks at the same time without affecting each other. Raghu et al. (2019) argued that when both tasks, the prediction, and the uncertainty quantification, are done by one single method, the uncertainty estimation is biased by the actual prediction task. Therefore, they recommended a "direct uncertainty prediction" and suggested training two neural networks, one for the actual prediction task and a second one for the prediction of the uncertainty on the first network's predictions. Similarly, Ramalho and Miranda (2020) introduced an additional neural network for uncertainty estimation. But in contrast to Raghu et al. (2019), the representation space of the training data is considered and the density around a given test sample is evaluated. The additional neural network uses this training data density in order to predict whether the main network's estimate is expected to be correct or false. Hsu et al. (2020) detected out-of-distribution examples in classification tasks at test-time by predicting total probabilities for each class, in addition to the categorical distribution given by the softmax output. The class-wise total probability is predicted by applying the sigmoid function to the network's logits. Based on these total probabilities, OOD examples can be identified as those with low class probabilities for all classes.

In contrast to this, (Oberdiek et al. 2018) took the sensitivity of the model, i.e. the model's slope, into account by using gradient metrics for the uncertainty quantification in classification tasks. Lee and AlRegib (2020) applied a similar idea but made use of back-propagated gradients. In their work, they presented state-of-the-art results on out-of-distribution and corrupted input detection.

3.1.3 Summing up single deterministic methods

Compared to many other principles, single deterministic methods are computationally efficient in training and evaluation. For training, only one network has to be trained and often the approaches can even be applied to pre-trained networks. Depending on the actual approach, only a single or at most two forward passes have to be fulfilled for evaluation. The underlying networks could contain more complex loss functions, which slows down the training process (Sensoy et al. 2018) or external components that have to be trained and evaluated additionally (Raghu et al. 2019). But in general, this is still more efficient than the number of predictions needed for ensembles based methods (Sect. 3.3), Bayesian methods (Sect. 3.2), and test-time data augmentation methods (Sect. 3.4). A drawback of single deterministic neural network approaches is the fact that they rely on a single opinion and can therefore become very sensitive to the underlying network architecture, training procedure, and training data.

3.2 Bayesian neural networks

Bayesian Neural Networks (BNNs) (Denker et al. 1987; Tishby et al. 1989; Buntine and Weigend 1991) have the ability to combine the scalability, expressiveness, and predictive performance of neural networks with the Bayesian learning as opposed to learning via the maximum likelihood principles. This is achieved by inferring the probability distribution over the network parameters $\theta = (w_1, \dots, w_K)$. More specifically, given a training input-target pair (x, y) the posterior distribution over the space of parameters $p(\theta|x, y)$ is modelled by assuming a prior distribution over the parameters $p(\theta)$ and applying Bayes theorem:

$$p(\theta|x, y) = \frac{p(y|x, \theta)p(\theta)}{p(y|x)} \propto p(y|x, \theta)p(\theta). \tag{16}$$

Here, the normalization constant in (16) is called the model evidence $p(y|x)$ which is defined as

$$p(y|x) = \int p(y|x, \theta)p(\theta)d\theta. \tag{17}$$

Once the posterior distribution over the weights has been estimated, the prediction of output y^* for a new input data x^* can be obtained by Bayesian Model Averaging or Full Bayesian Analysis that involves marginalizing the likelihood $p(y|x, \theta)$ with the posterior distribution:

$$p(y^*|x^*, x, y) = \int p(y^*|x^*, \theta)p(\theta|x, y)d\theta. \tag{18}$$

This Bayesian way of prediction is a direct application of the law of total probability and endows the ability to compute the principled predictive uncertainty. The integral of (18) is intractable for the most common prior posterior pairs and approximation techniques are therefore typically applied. The most widespread approximation, the *Monte Carlo Approximation*, follows the law of large numbers and approximates the expected value by the mean of N stochastic networks, $f_{\theta_1}, \dots, f_{\theta_N}$, parameterized by N samples, $\theta_1, \theta_2, \dots, \theta_N$, from the posterior distribution of the weights, i.e.

$$y^* \approx \frac{1}{N}\sum_{i=1}^{N} y_i^* = \frac{1}{N}\sum_{i=1}^{N} f_{\theta_i}(x^*). \tag{19}$$

Wilson and Izmailov (2020) argue that a key advantage of BNNs lies in this marginalization step, which particularly can improve both the accuracy and calibration of modern deep neural networks. We note that the use cases of BNNs are not limited to uncertainty estimation but open up the possibility to bridge the powerful Bayesian toolboxes within deep learning. Notable examples include Bayesian model selection (MacKay 1992a; Sato 2001; Corduneanu and Bishop 2001; Ghosh et al. 2019), model compression (Louizos et al. 2017; Federici et al. 2017; Achterhold et al. 2018), active learning (MacKay 1992b; Gal et al. 2017b; Kirsch et al. 2019), continual learning (Nguyen et al. 2018; Ebrahimi et al. 2020; Farquhar and Gal 2019; Li et al. 2020), theoretic advances in Bayesian learning (Khan et al. 2019) and beyond. While the formulation is rather simple, there exist several challenges. For example, no closed-form solution exists for the posterior inference as conjugate priors do not typically exist for complex models such as neural networks (Bishop and Nasrabadi 2006). Hence, approximate Bayesian inference techniques are often needed to compute the posterior probabilities. Yet, directly using approximate Bayesian inference techniques has been proven to be difficult as the size of the data and the number of parameters are too large for the use cases of deep neural networks. In other words, the integrals of the above equations are not computationally tractable as the size of the data and the number of parameters grows. Moreover, specifying a meaningful prior for deep neural networks is another challenge that is less understood.

In this survey, we classify the BNNs into three different types based on how the posterior distribution is inferred to approximate Bayesian inference:

- *Variational inference* (Hinton and Van Camp 1993; Barber and Bishop 1998)
  Variational inference approaches approximate the (in general intractable) posterior distribution by optimizing over a family of tractable distributions.
- *Sampling approaches* (Neal 1992)
  Sampling approaches deliver a representation of the target random variable from which realizations can be sampled. Such methods are based on Markov Chain Monte Carlo and further extensions.
- *Laplace approximation* (Denker and LeCun 1991; MacKay 1992c)
  Laplace approximation simplifies the target distribution by approximating the log-posterior distribution and then, based on this approximation, deriving a normal distribution over the network weights.

These three types differ in multiple criteria that are of interest for applicants. While variational inference and the Laplace approximation offer an analytical expression of the uncertainty and are derived in a deterministic manner, the sampling approaches generate samples and lack such an analytical expression and determinism. Here, it is important to note that the variational inference is deterministic, even though many approximations of it are based on stochastic sampling. On the other hand, the sampling approaches are not biased from the network's predictions and have the theoretical capability to combine multiple modes (i.e. multiple local solutions), where variational inference and the Laplace approximation only operate in the neighbourhood of a single mode. At the same time, a possible convergence to a solution is significantly harder to asses for the sampling approaches. Considering the computational costs, the Laplace approximation scales down to a normal neural network training, while the variational inference is slowed down by regularization and additional parameters that are needed for representing the uncertainty. The sampling approaches are most costly at training time since the training is already based on sampling. Further, the Laplace approximation has the advantage that it can be applied to pre-trained networks without any changes needed. At inference all the presented approaches are relatively costly since all are based on multiple forward passes in order to approximate the underlying probability distribution. An overview of the main differences in the three types can be found in Table 3.

While limiting our scope to these three categories, we also acknowledge several advances in related domains of BNN research. Some examples are (i) approximate inference techniques such as alpha divergence (Hernández-Lobato et al. 2016; Li and Gal 2017; Minka et al. 2005), expectation propagation (Minka 2001; Zhao et al. 2020), assumed density filtering (Hernández-Lobato and Adams 2015) etc, (ii) probabilistic programming to exploit modern Graphical Processing Units (GPUs) (Tran et al. 2016, 2017; Bingham et al. 2019; Cabañas et al. 2019), (iii) different types of priors (Ito et al. 2005; Sun et al. 2018), (iv) advancements in theoretical understandings of BNNs (Depeweg et al. 2017; Khan et al. 2019; Farquhar et al. 2020), (iv) uncertainty propagation techniques to speed up the marginalization procedures (Postels et al. 2019) and (v) computations of aleatoric uncertainty (Gast and Roth 2018; Depeweg et al. 2018).

3.2.1 Variational inference

The goal of variational inference is to infer the posterior probabilities $p(\theta|x, y)$ using a prespecified family of distributions $q(\theta)$. Here, this so-called variational family $q(\theta)$ is defined as a parametric distribution. An example is the Multivariate Normal distribution where its parameters are the mean and the covariance matrix. The main idea of variational inference is to find the settings of these parameters that make $q(\theta)$ to be close to the posterior of interest $p(\theta|x, y)$. This measure of closeness between the probability distributions is given by the Kullback–Leibler (KL) divergence

$$\mathrm{KL}(q||p) = \mathbb{E}_q\left[\log \frac{q(\theta)}{p(\theta|x, y)}\right]. \tag{20}$$

Due to the posterior $p(\theta|x, y)$ the KL-divergence in (20) can not be minimized directly. Instead, the evidence lower bound (ELBO), a function that is equal to the KL divergence up to a constant, is optimized. For a given prior distribution on the parameters $p(\theta)$, the ELBO is given by

Table 3 Overview over the properties of different types of Bayesian neural network approaches as also discussed in the introduction to Sect. 3.2. The properties are stated relatively among the approaches

	Variational inference	Sampling approaches	Laplace approximation
Description	Variational inference approaches approximate the (in general intractable) posterior distribution by optimizing over a family of tractable distributions achieved by minimizing the KL divergence	Representation of the target random variable from which realizations can be sampled. Such methods are based on Markov Chain Monte Carlo and further extensions	Simplifies the target distribution by approximating the log-posterior distribution and then, based on this approximation, deriving a normal distribution on the network weights
Analytic expression	Yes	No	Yes
Can be applied to pretrained networks	No	No	Yes
Deterministic?	Yes	No	Yes
Unbiased?	No	Yes	No
Optimum	Local optimum	Hard to mix between modes	Local optimum
Convergence	Easy to assess convergence	Hard to assess convergence	Easy to assess convergence
Computational effort at training time	Medium—convergence may be slowed down by regularization and additional parameters for uncertainty representation	High—M forward passes based on sampled parameters, otherwise intractable	Low—training of one deterministic model and Laplace approximation
Computational effort at inference	High—M forward passes based on sampled parameters, otherwise intractable	High—M forward passes based on sampled parameters, otherwise intractable	High—M forward passes based on sampled parameters, otherwise intractable
Application Scenarios	Medical: (Heo et al. 2018; Chen et al. 2021; Dusenberry et al. 2020; Nair et al. 2020; Roy et al. 2019; Scalia et al. 2020) Robotics: (Tchuiev and Indelman 2018; Feldman and Indelman 2018) Earth Observation: (Rußwurm et al. 2020)	Medical: (Liang et al. 2018a) Earth Observation: (Herrmann 2020) Plant Diseases: (Hernández and López 2020)	Robotics: (Humt et al. 2020; Yun and Liu 2023) Medical: (Niraula et al. 2022) Remote Sensing: (Rewicki 2021)

The properties should not be used to compare these approaches to other uncertainty methods such as ensembles, single deterministic models, and test-time augmentation methods. For a comparison of Bayesian methods to these methods see Table 1

$$L = \mathbb{E}_q\left[\log\frac{p(y|x,\theta)}{q(\theta)}\right] \tag{21}$$

and for the KL divergence

$$\mathrm{KL}(q||p) = -L + \log p(y|x) \tag{22}$$

holds.

Variational methods for BNNs have been pioneered by Hinton and Van Camp (Hinton and Van Camp 1993) where the authors derived a diagonal Gaussian approximation to the posterior distribution of neural networks (couched in information theory—a minimum description length). Another notable extension in the 1990s has been proposed by Barber and Bishop (1998), in which the full covariance matrix was chosen as the variational family, and the authors demonstrated how the ELBO can be optimized for neural networks. Several modern approaches can be viewed as extensions of these early works (Hinton and Van Camp 1993; Barber and Bishop 1998) with a focus on how to scale the variational inference to modern neural networks.

An evident direction with the current methods is the use of stochastic variational inference (or Monte-Carlo variational inference), where the optimization of ELBO is performed using a mini-batch of data. One of the first connections to stochastic variational inference has been proposed by Graves (2011) with Gaussian priors. In 2015, (Blundell et al. 2015) introduced Bayes By Backprop, a further extension of stochastic variational inference (Graves 2011) to non-Gaussian priors and demonstrated how the stochastic gradients can be made unbiased. Notable, (Kingma et al. 2015) introduced the local reparameterization trick to reduce the variance of the stochastic gradients. One of the key concepts is to reformulate the loss function of the neural network as the ELBO. As a result, the intractable posterior distribution is indirectly optimized and variational inference is compatible with back-propagation with certain modifications to the training procedure. These extensions widely focus on the fragility of stochastic variational inference that arises due to sensitivity to initialization, prior definition, and variance of the gradients. These limitations have been addressed recently by Wu et al. (2018), where a hierarchical prior was used and the moments of the variational distribution are approximated deterministically.

Above works commonly assumed mean-field approximations as the variational family, neglecting the correlations between the parameters. In order to make more expressive variational distributions feasible for deep neural networks, several works proposed to infer using the matrix normal distribution (Louizos and Welling 2016; Zhang et al. 2018a; Sun et al. 2017) or more expressive variants (Bae et al. 2018; Mishkin et al. 2018) where the covariance matrix is decomposed into the Kronecker products of smaller matrices or in a low-rank form plus a positive diagonal matrix. A notable contribution towards expressive posterior distributions has been the use of normalizing flows (Rezende and Mohamed 2015; Louizos and Welling 2017)—a hierarchical probability distribution where a sequence of invertible transformations are applied so that a simple initial density function is transformed into a more complex distribution. Interestingly, (Farquhar et al. 2020) argue that mean-field approximation is not a restrictive assumption, and the layer-wise weight correlations may not be as important as capturing the depth-wise correlations. While the claim of Farquhar et al. (2020) may remain to be an open question, the mean-field approximations have an advantage on smaller computational complexities (Farquhar et al. 2020). For example, (Osawa et al. 2019) demonstrated that variational inference can be scaled up to ImageNet size data sets and architectures using multiple GPUs and proposed practical tricks such as data augmentation, momentum initialization, and learning rate scheduling.

One of the successes in variational methods has been accomplished by casting existing stochastic elements of deep learning as variational inference. A widely known example is Monte Carlo Dropout (MC Dropout) where the dropout layers are formulated as Bernoulli distributed random variables, and training a neural network with dropout layers can be approximated as performing variational inference (Gal and Ghahramani 2015, 2016; Gal et al. 2017a). A main advantage of MC dropout is that the predictive uncertainty can be computed by activating dropout not only during training but also at test time. In this way, once the neural network is trained with dropout layers, the implementation efforts can be kept minimum and the practitioners do not need expert knowledge to reason about uncertainty—certain criteria that the authors are attributing to its success (Gal and Ghahramani 2016). The practical values of this method have been demonstrated also in several works (Eaton-Rosen et al. 2018; Loquercio et al. 2020; Rußwurm et al. 2020) and resulted in different extensions [evaluating the usage of different dropout masks for example for convolutional layers (Tassi and Rovile 2019) or by changing the representations of the predictive uncertainty into model and data uncertainties (Kendall and Gal 2017)]. Approaches that build upon a similar idea but randomly drop incoming activations of a node, instead of dropping an activation for all following nodes, were also proposed within the literature (Mobiny et al. 2021) and called drop connect. This was found to be more robust on the uncertainty representation, even though it was shown that a combination of both can lead to higher accuracy and robustness in the test predictions (McClure and Kriegeskorte 2016). Lastly, connections of variation inference to Adam (Khan et al. 2018), RMS Prop (Khan et al. 2017), and batch normalization (Atanov et al. 2019) have been further suggested in the literature.

3.2.2 Sampling methods

Sampling methods, also often called Monte Carlo methods, are another family of Bayesian inference algorithms that represent uncertainty without a parametric model. Specifically, sampling methods use a set of hypotheses (or samples) drawn from the distribution and offer the advantage that the representation itself is not restricted by the type of distribution (e.g. can be multi-modal or non-Gaussian)—hence probability distributions are obtained non-parametrically. Popular algorithms within this domain are particle filtering, rejection sampling, importance sampling, and Markov Chain Monte Carlo sampling (MCMC) (Bishop and Nasrabadi 2006). In the case of neural networks, MCMC is often used since alternatives such as rejection and importance sampling are known to be inefficient for such high-dimensional problems. The main idea of MCMC is to sample from arbitrary distributions by a transition in state space where this transition is governed by a record of the current state and the proposal distribution that aims to estimate the target distribution (e.g. the true posterior). To explain this, let us start defining the Markov Chain: a Markov Chain is a distribution over random variables $x_1, \cdots, x_T$ which follows the state transition rule:

$$p(x_1, \cdots, x_T) = p(x_1) \prod_{t=2}^{T} p(x_t | x_{t-1}), \tag{23}$$

i.e. the next state only depends on the current state and not on any other former state. In order to draw samples from the true posterior, MCMC sampling methods first generate samples in an iterative and the Markov Chain fashion. Then, at each iteration, the algorithm decides to either accept or reject the samples where the probability of acceptance is determined by certain rules. In this way, as more and more samples are produced, their values can approximate the desired distribution.

Hamiltonian Monte Carlo or Hybrid Monte Carlo (HMC) (Duane et al. 1987) is an important variant of MCMC sampling method (pioneered by Neal (1992, 1994, 1995); Neal et al. (2011) for neural networks), and is often known to be the gold standards of Bayesian inference (Neal et al. 2011; Dubey et al. 2016; Li and Gal 2017). The algorithm works as follows: (i) start by initializing a set of parameters θ (either randomly or in a user-specific manner). Then, for a given number of total iterations, (ii) instead of a random walk, a momentum vector—an auxiliary variable ρ is sampled, and the current value of parameters θ is updated via the Hamiltonian dynamics:

$$H(\rho, \theta) = -\text{log}p(\rho, \theta) = -\text{log}p(\rho|\theta) - \text{log}p(\theta). \tag{24}$$

Defining the potential energy ($V(\theta) = -logp(\theta)$ and the kinetic energy $T(\rho|\theta) = -\text{log}p(\rho|\theta)$, the update steps via Hamilton's equations are governed by,

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial \rho} = \frac{\partial T}{\partial \rho} \text{ and} \tag{25}$$

$$\frac{d\rho}{dt} = -\frac{\partial H}{\partial \theta} = -\frac{\partial T}{\partial \theta} - \frac{\partial V}{\partial \theta}. \tag{26}$$

The so-called leapfrog integrator is used as a solver (Leimkuhler and Reich 2004). (iii) For each step, a Metropolis acceptance criterion is applied to either reject or accept the samples (similar to MCMC). Unfortunately, HMC requires the processing of the entire data set per iteration, which is computationally too expensive when the data-set size grows to million to even billions. Hence, many modern algorithms focus on how to perform the computations in a mini-batch fashion stochastically. In this context, for the first time, (Welling and Teh 2011) proposed to combine Stochastic Gradient Descent (SGD) with Langevin dynamics [a form of MCMC (Rossky et al. 1978; Roberts and Stramer 2002; Neal et al. 2011)] in order to obtain a scalable approximation to MCMC algorithm based on mini-batch SGD (Kushner and Yin 2003; Goodfellow et al. 2016). The work demonstrated that performing Bayesian inference on Deep Neural Networks can be as simple as running a noisy SGD. This method does not include the momentum term of HMC via using the first-order Langevin dynamics and opened up a new research area on Stochastic Gradient Markov Chain Monte Carlo (SG-MCMC).

Consequently, several extensions are available which include the use of 2nd order information such as preconditioning and optimizing with the Fisher Information Matrix (FIM) (Ma et al. 2015; Marceau-Caron and Ollivier 2017; Nado et al. 2018), the Hessian (Simsekli et al. 2016; Zhang and Sutton 2011; Fu et al. 2016), adapting preconditioning diagonal matrix (Li et al. 2016a), generating samples from non-isotropic target densities using Fisher scoring (Ahn et al. 2012), and samplers in the Riemannian manifold (Patterson and Teh 2013) using the first order Langevin dynamics and Levy diffusion noise and momentum (Ye and Zhu 2018). Within these methods, the so-called parameter-dependent diffusion matrices are incorporated with the intention to offset the stochastic perturbation of the gradient. To do so, the "thermostat" ideas (Ding et al. 2014; Shang et al. 2015; Leimkuhler and Shang 2016) are proposed so that a prescribed constant temperature distribution is maintained with the parameter-dependent noise. Ahn et al. (2014) devised a distributed computing system for SG-MCMC to exploit the modern computing routines, while (Wang et al. 2018b) showed that Generative Adversarial Models (GANs) can be used to distill the samples for improved memory efficiency, instead of distillation for enhancing the run-time capabilities of computing predictive uncertainty (Balan et al. 2015). Lastly, other recent trends are techniques that reduce the variance (Dubey et al. 2016; Zou et al. 2018) and bias (Durmus et al. 2016; Durmus and Moulines 2019) arising from stochastic gradients.

Concurrently, there have been solid advances in the theory of SG-MCMC methods and their applications in practice. Sato and Nakagawa (Sato and Nakagawa 2014), for the first time, showed that the SGLD algorithm with constant step size weakly converges; (Chen et al. 2015) showed that faster convergence rates and more accurate invariant measures can be observed for SG-MCMCs with higher order integrators rather than a 1st order Euler integrator while (Teh et al. 2016) studied the consistency and fluctuation properties of the SGLD. As a result, verifiable conditions obeying a central limit theorem for which the algorithm is consistent, and how its asymptotic bias-variance decomposition depends on step-size sequences have been discovered. A more detailed review of the SG-MCMC with a focus on supporting theoretical results can be found in Nemeth and Fearnhead Nemeth and Fearnhead 2021. Practically, SG-MCMC techniques have been applied to shape classification and uncertainty quantification (Li et al. 2016b), empirically study and validate the effects of tempered posteriors (or called cold-posteriors) (Wenzel et al. 2020) and train a deep neural network in order to generalize and avoid over-fitting (Ye et al. 2017; Chandra et al. 2019).

3.2.3 Laplace approximation

The goal of the Laplace Approximation is to estimate the posterior distribution over the parameters of neural networks $p(\theta \mid x, y)$ around a local mode of the loss surface with a Multivariate Normal distribution. The Laplace Approximation to the posterior can be obtained by taking the second-order Taylor series expansion of the log posterior over the weights around the MAP estimate $\hat{\theta}$ given some data (x, y). If we assume a Gaussian prior with a scalar precision value $\tau > 0$, then this corresponds to the commonly used L_2-regularization, and the Taylor series expansion results in

$$\begin{aligned}\log p(\theta \mid x, y) &\approx \log p(\hat{\theta} \mid x, y)\\ &\quad + \frac{1}{2}(\theta - \hat{\theta})^T (H + \tau I)(\theta - \hat{\theta}),\end{aligned}$$

where the first-order term vanishes because the gradient of the log posterior $\delta\theta = \nabla \log p(\theta \mid x, y)$ is zero at the maximum $\hat{\theta}$. Taking the exponential on both sides and approximating integrals by reverse engineering densities, the weight posterior is approximately a Gaussian with the mean $\hat{\theta}$ and the covariance matrix $(H + \tau I)^{-1}$ where H is the Hessian of $\log p(\theta \mid x, y)$. This means that the model uncertainty is represented by the Hessian H resulting in a Multivariate Normal distribution:

$$p(\theta \mid x, y) \sim \mathcal{N}\big(\hat{\theta}, (H + \tau I)^{-1}\big). \tag{27}$$

In contrast to the two other methods described, the Laplace approximation can be applied on already trained networks and is generally applicable when using standard loss functions such as MSE or cross entropy and piece-wise linear activations (e.g. RELU). MacKay (1992c) and Denker and LeCun (1991) have pioneered the Laplace approximation for neural networks in the 1990s, and several modern methods provide an extension to deep neural networks (Botev et al. 2017; Martens and Grosse 2015; Ritter et al. 2018; Lee et al. 2020).

The core of the Laplace Approximation is the estimation of the Hessian. Unfortunately, due to the enormous number of parameters in modern neural networks, the Hessian matrices cannot be computed in a feasible way as opposed to relatively smaller networks in MacKay (1992c) and Denker and LeCun (1991). Consequently, several different ways for approximating H have been proposed in the literature. A brief review is as follows. Instead of diagonal approximations [e.g. Becker and LeCun (1989), Salimans and Kingma (2016)], several researchers have been focusing on including the off-diagonal elements [e.g. Liu and Nocedal (1989), Hennig (2013) and Le Roux and Fitzgibbon (2010)]. Amongst them, layer-wise Kronecker Factor approximation of Grosse and Martens (2016), Martens and Grosse (2015), Botev et al. (2017) and Chen et al. (2018) have demonstrated a notable scalability (Ba et al. 2016). A recent extension can be found in George et al. (2018) where the authors propose to re-scale the eigenvalues of the Kronecker factored matrices so that the diagonal variance in its eigenbasis is accurate. The work presents an interesting idea as one can prove that in terms of a Frobenius norm, the proposed approximation is more accurate than that of Martens and Grosse (2015). However, as this approximation is harmed by inaccurate estimates of eigenvectors, (Lee et al. 2020) proposed to further correct the diagonal elements in the parameter space.

Existing works obtain Laplace Approximation using various approximations of the Hessian in the line of fidelity-complexity trade-offs. For several works, an approximation using the diagonal of the Fisher information matrix or Gauss-Newton matrix, leading to independently distributed model weights, has been utilized in order to prune weights (LeCun et al. 1989) or perform continual learning in order to avoid catastrophic forgetting (Kirkpatrick et al. 2017). In Ritter et al. (2018), the Kronecker factorization of the approximate block-diagonal Hessian (Martens and Grosse 2015; Botev et al. 2017) have been applied to obtain scalable Laplace Approximation for neural networks. With this, the weights among different layers are still assumed to be independently distributed, but not the correlations within the same layer. Recently, building upon the current understanding of neural network's loss landscape that many eigenvalues of the Hessian tend to be zero, (Lee et al. 2020) developed a low-rank approximation that leads to sparse representations of the layers' co-variance matrices. Furthermore, (Lee et al. 2020) demonstrated that the Laplace Approximation can be scaled to ImageNet size data sets and architectures, and further showed that with the proposed sparsification technique, the memory complexity of modelling correlations can be made similar to the diagonal approximation. Lastly, (Kristiadi et al. 2020) proposed a simple procedure to compute the last-layer Gaussian approximation (neglecting the model uncertainty in all other layers of neural networks), and showed that even such a minimalist solution can mitigate overconfidence predictions of ReLU networks.

Recent efforts have extended the Laplace Approximation beyond the Hessian approximation. To tackle the widely known assumption that the Laplace Approximation is for the bell-shaped true posterior and thus resulting in under-fitting behavior (Ritter et al. 2018; Humt et al. 2020) proposed to use Bayesian Optimization and showed that hyperparameters of the Laplace Approximation can be efficiently optimized with increased calibration performance. Another work in this domain is by Kristiadi et al. (2021), who proposed uncertainty units—a new type of hidden units that changes the geometry of the loss landscape so that more accurate inference is possible. While (Shinde et al. 2020) demonstrated the practical effectiveness of the Laplace Approximation to the autonomous driving applications, (Feng et al. 2019) showed the possibility to (i) incorporate contextual information and (ii) domain adaptation in a semi-supervised manner within the context of image classification. This is achieved by designing unary potentials within a Conditional Random Field. Several real-time methods also exist that do not require multiple forwards passes to compute the predictive uncertainty. So-called linearized Laplace Approximation has been proposed in Foong et al. (2019), Immer et al. (2021) using the ideas of MacKay (1992b) and has been extended with Laplace bridge for classification (Hobbhahn et al. 2022). Within this framework, (Daxberger et al. 2020) proposed inferring the sub-networks to increase the expressivity of covariance propagation while remaining computationally tractable.

3.2.4 Sum up Bayesian methods

Bayesian methods for deep learning have emerged as a strong research domain by combining principled Bayesian learning for deep neural networks. A review of current BNNs has been provided with a focus on mostly, how the posterior $p(\theta|x, y)$ is inferred. As an observation, many of the recent breakthroughs have been achieved by performing approximate Bayesian inference in a mini-batch fashion (stochastically) or investigating relatively simple but scalable techniques such as MC-dropout or Laplace Approximation. As a result, several works demonstrated that the posterior inference in large-scale settings are now possible (Maddox et al. 2019; Osawa et al. 2019; Lee et al. 2020), and the field has several practical approximation tools to compute more expressive and accurate posteriors since the revival of BNNs beyond the pioneers (Hinton and Van Camp 1993; Barber and Bishop 1998; Neal 1992; Denker and LeCun 1991; MacKay 1992c). There are also emerging challenges on new frontiers beyond accurate inference techniques. Some examples are: (i) how to specify meaningful priors? (Ito et al. 2005; Sun et al. 2018), (ii) how to efficiently marginalize over the parameters for fast predictive uncertainty? (Balan et al. 2015; Postels et al. 2019; Hobbhahn et al. 2022; Lee et al. 2022) (iii) infrastructures such as new benchmarks, evaluation protocols and software tools (Mukhoti et al. 2018; Tran et al. 2017; Bingham et al. 2019; Filos et al. 2019), and (iv) towards better understandings on the current methodologies and their potential applications (Farquhar et al. 2020; Wenzel et al. 2020; Mukhoti and Gal 2018; Feng et al. 2019).

3.3 Ensemble methods

3.3.1 Principles of ensemble methods

Ensembles derive a prediction based on the predictions received from multiple so-called ensemble members. They target a better generalization by making use of synergy effects among the different models, arguing that a group of decision-makers tend to make better decisions than a single decision-maker (Sagi and Rokach 2018; Hansen and Salamon 1990). For an ensemble $f : X \rightarrow Y$ with members $f_i : X \rightarrow Y$ for $i \in 1, 2, \ldots, M$, this could be for example implemented by simply averaging over the members' predictions,

$$f(x) := \frac{1}{M}\sum_{i=1}^{M} f_i(x).$$

Based on this intuitive idea, several works applying ensemble methods to different kinds of practical tasks and approaches, for example bio-informatics (Cao et al. 2020; Nanni et al. 2020; Wei et al. 2017), remote sensing (Lv et al. 2017; Dai et al. 2019; Marushko and

Doudkin 2020), or reinforcement learning (Kurutach et al. 2018; Rajeswaran et al. 2017) can be found in the literature. Besides the improvement in accuracy, ensembles give an intuitive way of representing the model uncertainty on a prediction by evaluating the variety among the member's predictions.

Compared to Bayesian and single deterministic network approaches, ensemble methods have two major differences. First, the general idea behind ensembles is relatively clear and there are not many groundbreaking differences in the application of different types of ensemble methods and their application in different fields. Hence, this section focuses on different strategies to train an ensemble and some variations that target making ensemble methods more efficient. Second, ensemble methods were originally not introduced to explicitly handle and quantify uncertainties in neural networks. Although the derivation of uncertainty from ensemble predictions is obvious, since they actually aim at reducing the model uncertainty, ensembles were first introduced and discussed in order to improve the accuracy on a prediction (Hansen and Salamon 1990). Therefore, many works on ensemble methods do not explicitly take the uncertainty into account. Notwithstanding this, ensembles have been found to be well suited for uncertainty estimations in neural networks (Lakshminarayanan et al. 2017).

3.3.2 Single- and multi-mode evaluation

One main point where ensemble methods differ from the other methods presented in this paper is the number of local optima that are considered, i.e. the differentiation into *single-mode* and *multi-mode* evaluation.

In order to create synergies and marginalise false predictions of single members, the members of an ensemble have to behave differently in case of an uncertain outcome. The mapping defined by a neural network is highly non-linear and hence the optimized loss function contains many local optima to which a training algorithm could converge. Deterministic neural networks converge to one single local optimum in the solution space (Fort et al. 2019). Other approaches, e.g. BNNs, still converge to one single optimum, but additionally, take the uncertainty on this local optimum into account (Fort et al. 2019). This means, that neighbouring points within a certain region around the solution also affect the loss and also influence the prediction of a test sample. Since these methods focus on single regions, the evaluation is called *single-mode* evaluation. In contrast to this, ensemble methods consist of several networks, which should converge to different local optima. This leads to a so-called multi-mode evaluation (Fort et al. 2019).

In Fig. 7, the considered parameters of a single-mode deterministic, single-mode probabilistic (Bayesian), and multi-mode ensemble approach are visualized. The goal of multi-mode evaluation is that different local optima could lead to models with different strengths and weaknesses in the predictions such that a combination of several such models brings synergy effects improving the overall performance.

3.3.3 Bringing variety into ensembles

One of the most crucial points when applying ensemble methods is to maximize the variety in the behaviour among the single networks (Renda et al. 2019; Lakshminarayanan et al. 2017). In order to increase the variety, several different approaches can be applied:

- Random initialization and data shuffle

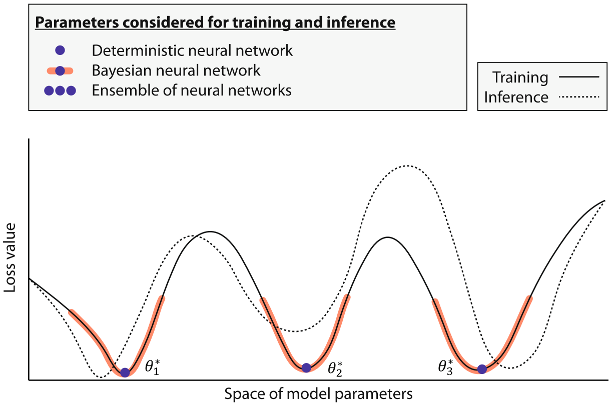

Parameters considered for training and inference
Deterministic neural network
Bayesian neural network
Ensemble of neural networks
Training
Inference
Loss value
θ_1^*
θ_2^*
θ_3^*
Space of model parameters

Fig. 7 A visualization of the different evaluation behaviours of deterministic neural networks, Bayesian neural networks, and the ensemble of deterministic neural networks. The x-axis indicates the network parameters θ and the y-axis represents the loss value. While the deterministic network learns the parameters based on a pointwise estimation, the Bayesian neural network also takes the surrounding of the single point into account. The ensemble of deterministic methods optimizes pointwise but learns several different parameter settings

Due to the very non-linear loss landscape, different initializations of a neural network lead in general to different training results. Since the training is realized on mini-batches, the order of the training data points also affects the final result.

- Bagging and boosting

  Bagging (Bootstrap aggregating) and Boosting are two strategies that vary the distribution of the used training data sets by sampling new sets of training samples from the original set. Bagging is sampling from the training data uniformly and with replacement (Bishop and Nasrabadi 2006). Thanks to the replacement process, ensemble members can see single samples several times in the training set while missing some other training samples. For boosting, the members are trained one after another, and the probability of sampling a sample for the next training set is based on the performance of the already trained ensemble (Bishop and Nasrabadi 2006).
- Data augmentation

  Augmenting the input data randomly for each ensemble member leads to models trained on different data points and therefore in general to a larger variety among the different members.
- Ensemble of different network architecture

  The combination of different network architectures leads to different loss landscapes and can therefore also increase the diversity in the resulting predictions (Herron et al. 2020).

In several works, it has been shown that the variety induced by random initialization works sufficiently and that bagging could even lead to a weaker performance (Lee et al. 2015; Lakshminarayanan et al. 2017). Livieris et al. (2021) evaluated different bagging and boosting strategies for ensembles of weight-constrained neural networks. Interestingly, it is found that bagging performs better for a small number of ensemble members while boosting performs better for a large number. Nanni et al. (2019) evaluated ensembles based on different types of image augmentation for bioimage classification tasks and compared those to each other. Guo and Gould (2015) used augmentation methods within in an ensemble approach for object detection. Both works stated that the ensemble approach using augmentations improves the resulting accuracy. In contrast to this, (Rahaman et al. 2021; Wen et al. 2021b) stated with respect to uncertainty quantification that image augmentation can harm the calibration of an ensemble, and post-processing calibration methods have to be slightly adapted when using ensemble methods. Other ways of inducing variety for specific tasks have been also introduced. For instance, in Kim et al. (2018), the members are trained with different attention masks in order to focus on different parts of the input data. Other approaches focused on the training process and introduced learning rate schedulers that are designed to discover several local optima within one training process (Huang et al. 2017; Yang and Wang 2020). Following, an ensemble can be built based on local optima found within one single training run. It is important to note that if not explicitly stated, the works and approaches presented so far targeted improvements in predictive accuracy and did not explicitly consider uncertainty quantification.

3.3.4 Ensemble methods and uncertainty quantification

Besides the improvement in the accuracy, ensembles are widely used for modelling uncertainty on predictions of complex models, such as in climate prediction (Leutbecher and Palmer 2008; Parker 2013). Accordingly, ensembles are also used for quantifying the uncertainty on a deep neural network's prediction, and over the last years they became more and more popular for such tasks (Lakshminarayanan et al. 2017; Renda et al. 2019). Lakshminarayanan et al. (2017) are often referenced as a base work on uncertainty estimations derived from ensembles of neural networks and as a reference for the competitiveness of deep ensembles. They introduced an ensemble training pipeline to quantify predictive uncertainty within DNNs. In order to handle data and model uncertainty, the member networks are designed with two heads, representing the prediction and a predicted value of data uncertainty on the prediction. The approach is evaluated with respect to accuracy, calibration, and out-of-distribution detection for classification and regression tasks. In all tests, the method performs at least equally well as the BNN approaches used for comparison, namely Monte Carlo Dropout and Probabilistic Backpropagation. Lakshminarayanan et al. (2017) also showed that shuffling the training data and a random initialization of the training process induces a sufficient variety in the models in order to predict the uncertainty for the given architectures and data sets. Furthermore, bagging is even found to worsen the predictive uncertainty estimation, extending the findings of Lee et al. (2015), who found bagging to worsen the predictive accuracy of ensemble methods on the investigated tasks. Gustafsson et al. (2020) introduced a framework for the comparison of uncertainty quantification methods with a specific focus on real life applications. Based on this framework, they compared ensembles and Monte Carlo dropouts and found ensembles to be more reliable and applicable to real life applications. These findings endorse the results reported by Beluch et al. (2018) who found ensemble methods to deliver more accurate and better calibrated predictions on active learning tasks than Monte Carlo Dropout. Ovadia et al. (2019) evaluated different uncertainty quantification methods based on test sets affected by distribution shifts. The excessive evaluation contains a variety of model types and data modalities. As a takeaway, the authors stated that already for a relatively small ensemble size of five, deep ensembles seem to perform best and are more robust to data set shifts than the compared methods. Vyas et al. (2018) presented an ensemble method for the improved detection of out-of-distribution samples. For each member, a subset of the training data is considered as out-of-distribution. For the training process, a loss, seeking a minimum margin greater than zero between the average entropy of the in-domain and the out-of-distribution subsets is introduced and leads to a significant improvement in the out-of-distribution detection.

3.3.5 Making ensemble methods more efficient

Compared to single model methods, ensemble methods come along with a significantly increased computational effort and memory consumption (Sagi and Rokach 2018; Malinin et al. 2020). When deploying an ensemble for a real life application the available memory and computational power are often limited. Such limitations could easily become a bottleneck (Kocić et al. 2019) and could become critical for applications with limited reaction time. Reducing the number of models leads to less memory and computational power consumption. *Pruning approaches* reduce the complexity of ensembles by pruning over the members and reducing the redundancy among them. For that, several approaches based on different diversity measures are developed to remove single members without strongly affecting the performance (Guo et al. 2018; Cavalcanti et al. 2016; Martínez-Muñoz et al. 2008).

Distillation is another approach where the number of networks is reduced to one single model. It is the procedure of teaching a single network to represent the knowledge of a group of neural networks (Buciluǎ et al. 2006). First works on the distillation of neural networks were motivated by restrictions when deploying large-scale classification problems (Buciluǎ et al. 2006). The original classification problem is separated into several sub-problems focusing on single blocks of classes that are difficult to differentiate. Several smaller trainer networks are trained on the sub-problems and then teach one student network to separate all classes at the same time. In contrast to this, *Ensemble distillation approaches* capture the behaviour of an ensemble by one single network. The first works on ensemble distillation used the average of the softmax outputs of the ensemble members in order to teach a student network the derived predictive uncertainty (Hinton et al. 2015). Englesson and Azizpour (2019) justify the resulting predictive distributions of this approach and additionally cover the handling of out-of-distribution samples. When averaging over the members' outputs, the model uncertainty, which is represented in the variety of ensemble outputs, gets lost. To overcome this drawback, researchers applied the idea of learning higher order distributions, i.e. distributions over a distribution, instead of directly predicting the output (Lindqvist et al. 2020; Malinin et al. 2020). The members are then distillated based on the divergence from the average distribution. The idea is closely related to the prior networks (Malinin and Gales 2018) and the evidential neural networks (Sensoy et al. 2018), which are described in Sect. 3.1. Malinin et al. (2020) modelled ensemble members and the distilled network as prior networks predicting the parameters of a Dirichlet distribution. The distillation then seeks to minimize the KL divergence between the averaged Dirichlet distributions of the ensemble members and the output of the distilled network. Lindqvist et al. (2020) generalized this idea to any other parameterizable distribution. With that, the method is also applicable to regression problems, for example by predicting a mean and standard deviation to describe a normal distribution. Within several tests, the distillation models generated by these approaches are able to distinguish between data uncertainty and model uncertainty. Although distillation methods cannot completely capture the behaviour of an underlying ensemble, it has been shown that they are capable of delivering good and for some experiments even comparable results (Lindqvist et al. 2020; Malinin et al. 2020; Reich et al. 2020).

Other approaches, as *sub-ensembles* (Valdenegro-Toro 2019) and *batch-ensembles* (Wen et al. 2019) seek to reduce the computation effort and memory consumption by sharing parts among the single members. It is important to note that the possibility of using different model architectures for the ensemble members could get lost when parts of the ensemble are shared. Also, the training of the models cannot be run in a completely independent manner. Therefore, the actual time needed for training does not necessarily decrease in the same way as the computational effort does.

Sub-ensembles (Valdenegro-Toro 2019) divide a neural network architecture into two sub-networks. The trunk network is for the extraction of general information from the input data, and the task network uses this information to fulfill the actual task. In order to train a sub-ensemble, first, the weights of each member's trunk network are fixed based on the resulting parameters of one single model's training process. Following, the parameters of each ensemble member's task network are trained independently from the other members. As a result, the members are built with a common trunk and an individual task sub-network. Since the training and the evaluation of the trunk network have to be done only once, the number of computations needed for training and testing decreases by the factor $\frac{M\cdot N_{\text{task}}+N_{\text{trunk}}}{M\cdot N}$, where N_{task}, N_{trunk}, and N stand for the number of variables in the task networks, the trunk network, and the complete network. Valdenegro-Toro (2019) further underlined the usage of a shared trunk network by arguing that the trunk network is in general computational more costly than the task network. In contrast to this, batch-ensembles (Wen et al. 2019) connect the member networks with each other at every layer. The ensemble members' weights are described as a Hadamard product of one shared weight matrix $W \in \mathbb{R}^{n\times m}$ and M individual rank one matrix $F_i \in \mathbb{R}^{n\times m}$, each linked with one of the M ensemble members. The rank one matrices can be written as a multiplication $F_i = r_i s_i^{\mathrm{T}}$ of two vectors $s \in \mathbb{R}^n$ and $r \in \mathbb{R}^m$ and hence the matrix F_i can be described by $n+m$ parameters. With this approach, each additional ensemble member increases the number of parameters only by the factor $\frac{n+m}{M\cdot(n+m)+n\cdot m}+1$ instead of $\frac{M+1}{M} = 1 + \frac{1}{M}$. On the one hand, with this approach, the members are not independent anymore such that all the members have to be trained in parallel. On the other hand, the authors also showed that parallelization can be realized similarly to the optimization on mini-batches and on a single unit.

3.3.6 Sum up ensemble methods

Ensemble methods are very easy to apply since no complex implementation or major modification of the standard deterministic model have to be realized. Furthermore, ensemble members are trained independently from each other, which makes the training easily parallelizable. Also, trained ensembles can be extended easily, but the needed memory and computational effort increase linearly with the number of members for training and evaluation. The main challenge when working with ensemble methods is the need of introducing diversity among the ensemble members. For accuracy, uncertainty quantification, and out-of-distribution detection, random initialization, data shuffling, and augmentations have been found to be sufficient for many applications and tasks (Lakshminarayanan et al. 2017; Nanni et al. 2019). Since these methods may be applied anyway, they do not need much additional effort. The independence of the single ensemble members leads to a linear increase in the required memory and computation power with each additional member. This holds for the training as well as for testing. This limits the deployment of ensemble methods in many practical applications where the computation power or memory is limited, the application is time-critical, or very large networks with high inference time are included (Malinin et al. 2020).

Many aspects of ensemble approaches are only investigated with respect to the performance on the predictive accuracy but do not take predictive uncertainty into account. This also holds for the comparison of different training strategies for a broad range of problems and data sets. Especially since the overconfidence from single members can be transferred to the whole ensemble, strategies that encourage the members to deliver different false predictions instead of all delivering the same false prediction should be further investigated. For a better understanding of ensemble behavior, further evaluations of the loss landscape, as done by Fort et al. (2019), could offer interesting insights.

3.4 Test-time augmentation

Inspired by ensemble methods and adversarial examples (Ayhan and Berens 2018), the test-time augmentation is one of the simpler predictive uncertainty estimation techniques. The basic method is to create multiple test samples from each test sample by applying data augmentation techniques on it and then test all those samples to compute a predictive distribution in order to measure uncertainty. The idea behind this method is that the augmented test samples allow the exploration of different views and is therefore capable of capturing the uncertainty. In general, test-time augmentation can use the same augmentation techniques that can be used for regularization during training and has been shown to improve calibration to in-distribution data and out-of-distribution data detection (Ashukha et al. 2019; Lyzhov et al. 2020). Mostly, this technique of test-time augmentations has been used in medical image processing (Wang et al. 2018a, 2019; Ayhan and Berens 2018; Moshkov et al. 2020). One of the reasons for this is that the field of medical image processing already makes heavy use of data augmentations while using deep learning (Ronneberger et al. 2015), so it is quite easy to just apply those same augmentations during test time to calculate the uncertainties. Another reason is that collecting medical images is costly, thus forcing practitioners to rely on data augmentation techniques. Moshkov et al. (2020) used the test-time augmentation technique for cell segmentation tasks. For that, they created multiple variations of the test data before feeding it to a trained UNet or Mask R-CNN architecture. Following this, they used majority voting to create the final output segmentation mask and discuss the policies of applying different augmentation techniques and how they affect the final predictive results of the deep networks.

Overall, test-time augmentation is an easy method for estimating uncertainties because it keeps the underlying model unchanged, requires no additional data, and is simple to put into practice with off-the-shelf libraries. Nonetheless, it needs to be kept in mind that during applying this technique, one should only apply valid augmentations to the data, meaning that the augmentations should not generate data from outside the target distribution. According to Shanmugam et al. (2020), test-time augmentation can change many correct predictions into incorrect predictions (and vice versa) due to many factors such as the nature of the problem at hand, the size of training data, the deep neural network architecture, and the type of augmentation. To limit the impact of these factors, (Shanmugam et al. 2020) proposed a learning-based method for test-time augmentation that takes these factors into consideration. In particular, the proposed method learns a function that aggregates the predictions from each augmentation of a test sample. Similar to Shanmugam et al. (2020), Lyzhov et al. (2020) proposed a method, named “greedy Policy Search”, for constructing a test-time augmentation policy by choosing augmentations to be included in a fixed-length policy. Similarly, (Kim et al. 2020) proposed a method for learning a loss predictor from the training data for instance-aware test-time augmentation selection. The predictor selects test-time augmentations with the lowest predicted loss for a given sample.

Although learnable test-time augmentation techniques (Shanmugam et al. 2020; Lyzhov et al. 2020; Kim et al. 2020) help to select valid augmentations, one of the major open questions is to find out the effect on uncertainty due to different kinds of augmentations. It can for example happen that a simple augmentation-like reflection is not able to capture much of the uncertainty while some domain-specialized stretching and shearing captures more uncertainty. It is also important to find out how many augmentations are needed to correctly quantify uncertainties in a given task. This is particularly important in applications like earth observation, where inference might be needed on a global scale with limited resources.

3.5 Neural network uncertainty quantification approaches for real life applications

In order to use the presented methods on real life tasks, several different considerations have to be taken into account. The memory and computational power are often restricted while many real world tasks may be time-critical (Kocić et al. 2019). An overview of the main properties is given in Table 1.

The presented applications all come along with advantages and disadvantages, depending on the properties a user is interested in. While ensemble methods and test-time augmentation methods are relatively easy to apply, Bayesian approaches deliver a clear description of the uncertainty on the models parameters and also deliver a deeper theoretical basis. The computational effort and memory consumption is a common restriction on real life applications, where single deterministic network approaches perform best, but the distillation of ensembles or efficient Bayesian methods can also be taken into consideration. Within the different types of Bayesian approaches, the performance, the computational effort, and the implementation effort still vary strongly. Laplace approximations are relatively easy to apply and compared to sampling approaches much less computational effort is needed. Furthermore, there often already exist pretrained networks for an application. In this case, Laplace Approximation and external deterministic single network approaches can in general be applied to already trained networks.

Another important aspect that has to be taken into account for uncertainty quantification in real life applications is the source and type of uncertainty. For real life applications, out-of-distribution detection forms the maybe most important challenge in order to avoid unexpected decisions of the network and to be aware of adversarial attacks. Especially since many motivations of uncertainty quantification are given by risk minimization, methods that deliver risk-averse predictions are an important field to evaluate. Many works already demonstrated the capability of detecting out-of-distribution samples on several tasks and built a strong fundamental tool set for the deployment in real life applications (Yu and Aizawa 2019; Vyas et al. 2018; Ren et al. 2019; Gustafsson et al. 2020). However, in real life, the tasks are much more difficult than finding out-of-distribution samples among data sets (e.g., MNIST or CIFAR data sets) and the main challenge lies in comparing such approaches on several real-world data sets against each other. The work of Gustafsson et al. (2020) forms a first important step towards an evaluation of methods that better suits the demands in real life applications. Interestingly, they show for their tests ensembles to outperform the considered Bayesian approaches. This indicates, that the multi-mode evaluation given by ensembles is a powerful property for real life applications. Nevertheless, Bayesian approaches have delivered strong results as well and furthermore come along with a strong theoretical foundation (Lee et al. 2020; Hobbhahn et al. 2022; Eggenreich et al. 2020; Gal et al. 2017b). As a way to go, the combination of efficient ensemble strategies and Bayesian approaches could combine the variability in the model parameters while still considering several modes for a prediction. Also, single deterministic approaches as the prior networks (Malinin and Gales 2018; Nandy et al. 2020; Sensoy et al. 2018; Zhao et al. 2019) deliver comparable results while consuming significantly less computation power. However, this efficiency often comes along with the problem that separated sets of in- and out-of-distribution samples have to be available for the training process (Zhao et al. 2019; Nandy et al. 2020). In general, the development of new problem and loss formulations such as given in Nandy et al. (2020) leads to a better understanding and description of the underlying problem and forms an important field of research.

4 Uncertainty measures and quality

In Sect. 3, we presented different methods for modeling and predicting different types of uncertainty in neural networks. In order to evaluate these approaches, measures have to be applied to the derived uncertainties. In the following, we present different measures for quantifying the different predicted types of uncertainty. In general, the correctness and trustworthiness of these uncertainties are not automatically given. In fact, there are several reasons why evaluating the quality of the uncertainty estimates is a challenging task.

- First, the quality of the uncertainty estimation depends on the underlying method for estimating uncertainty. This is exemplified in the work undertaken by Yao et al. (2019), which shows that different approximates of Bayesian inference (e.g. Gaussian and Laplace approximates) result in different qualities of uncertainty estimates.
- Second, there is a lack of ground truth uncertainty estimates (Lakshminarayanan et al. 2017), and defining ground truth uncertainty estimates is challenging. For instance, if we define the ground truth uncertainty as the uncertainty across human subjects, we still have to answer questions as “How many subjects do we need?” or “How to choose the subjects?”.
- Third, there is a lack of a unified quantitative evaluation metric (Huang et al. 2019b). To be more specific, uncertainty is defined differently in different machine learning tasks such as classification, segmentation, and regression. For instance, prediction intervals or standard deviations are used to represent uncertainty in regression tasks, while entropy (and other related measures) are used to capture uncertainty in classification and segmentation tasks.

4.1 Evaluating uncertainty in classification tasks

For classification tasks, the network’s softmax output already represents a measure of confidence. But since the raw softmax output is neither very reliable (Hendrycks and Gimpel 2017) nor can it represent all sources of uncertainty (Smith and Gal 2018), further approaches and corresponding measures were developed.

4.1.1 Measuring data uncertainty in classification tasks

Consider a classification task with K different classes and a probability vector network output $p(x)$ for some input sample x. In the following p is used for simplification and p_k stands for the k-th entry in the vector. In general, the given prediction p represents a categorical distribution, i.e. it assigns a probability to each class to be the correct prediction. Since the prediction is not given as an explicit class but as a probability distribution, (un)certainty estimates can be directly derived from the prediction. In general, this pointwise prediction can be seen as estimated data uncertainty (Kendall and Gal 2017). However, as discussed in Sect. 2, the model's estimation of the data uncertainty is affected by model uncertainty, which has to be taken into account separately. In order to evaluate the amount of predicted data uncertainty, one can for example apply the maximal class probability or the entropy measures:

$$\text{Maximal probability:} \qquad p_{\max} = \max \left\{p_k\right\}_{k=1}^{K} \tag{28}$$

$$\text{Entropy:} \quad \mathrm{H}(p) = -\sum_{k=1}^{K} p_k \log_2(p_k) \tag{29}$$

The maximal probability represents a direct representation of certainty, while entropy describes the average level of information in a random variable. Even though a softmax output should represent the data uncertainty, one cannot tell from a single prediction how large the amount of model uncertainty is that affects this specific prediction as well.

4.1.2 Measuring model uncertainty in classification tasks

As already discussed in Sect. 3, a single softmax prediction is not a very reliable way for uncertainty quantification since it is often badly calibrated (Smith and Gal 2018) and does not have any information about the certainty of the model itself has on this specific output (Smith and Gal 2018). An (approximated) posterior distribution $p(\theta|D)$ on the learned model parameters can help to receive better uncertainty estimates. With such a posterior distribution, the softmax output itself becomes a random variable and one can evaluate its variation, i.e. uncertainty. For simplicity, we denote $p(y|\theta, x)$ also as p and it will be clear from context whether p depends on θ or not. The most common measures for this are mutual information (MI), the expected Kullback–Leibler Divergence (EKL), and the predictive variance. Basically, all these measures compute the expected divergence between the (stochastic) softmax output and the expected softmax output

$$\hat{p} = \mathbb{E}_{\theta\sim p(\theta|D)}\left[p(y|x,\theta\right] . \tag{30}$$

The MI uses entropy to measure the mutual dependence between two variables. In the described case, the difference between the information given in the expected softmax output and the expected information in the softmax output is compared, i.e.

$$\mathrm{MI}(\theta, y|x, D) = \mathrm{H}\left[\hat{p}\right] - \mathbb{E}_{\theta\sim p(\theta|D)}\mathrm{H}\left[p(y|x,\theta)\right] . \tag{31}$$

Smith and Gal (2018) pointed out that the MI is minimal when the knowledge about model parameters does not increase the information in the final prediction. Therefore, the MI can be interpreted as a measure of model uncertainty.

The Kullback–Leibler divergence measures the divergence between two given probability distributions. The EKL can be used to measure the (expected) divergence among the possible softmax outputs,

$$\mathbb{E}_{\theta\sim p(\theta|D)}\left[KL(\hat{p}\,\|\,p)\right] = \mathbb{E}_{\theta\sim p(\theta|D)}\left[\sum_{i=1}^{K}\hat{p}_i \log\left(\frac{\hat{p}_i}{p_i}\right)\right], \tag{32}$$

which can also be interpreted as a measure of uncertainty on the model's output and therefore represents the model uncertainty.

The predictive variance evaluates the variance on the (random) softmax outputs, i.e.

$$\sigma(p) = \mathbb{E}_{\theta\sim p(\theta|D)}\left[(p-\hat{p})^2\right]. \tag{33}$$

As described in Sect. 3, an analytically described posterior distribution $p(\theta|D)$ is only given for a subset of the Bayesian methods. And even for an analytically described distribution, the propagation of the parameter uncertainty into the prediction is in almost all cases intractable and has to be approximated for example with Monte Carlo approximation. Similarly, ensemble methods collect predictions from M neural networks, and test-time data augmentation approaches receive M predictions from M different augmentations applied to the original input sample. For all these cases, we receive a set of M samples, $\left\{p^i\right\}_{i=1}^{M}$, which can be used to approximate the intractable or even undefined underlying distribution. With these approximations, the measures defined in (31), (32), and (33) can be applied straight forward and only the expectation has to be replaced by average sums. For example, the expected softmax output becomes

$$\hat{p} \approx \frac{1}{M}\sum_{i=1}^{M} p^i\,.$$

For the expectations given in (31), (32), and (33), the expectation is approximated similarly.

4.1.3 Measuring distributional uncertainty in classification tasks

Although these uncertainty measures are widely used to capture the variability among several predictions derived from BNNs (Kendall and Gal 2017), ensemble methods (Lakshminarayanan et al. 2017), or test-time data augmentation methods (Ayhan and Berens 2018), they cannot capture distributional shifts in the input data or OOD examples, which could lead to a biased inference process and a falsely stated confidence. If all predictors attribute a high probability mass to the same (false) class label, this induces a low variability among the estimates. Hence, the network seems to be certain about its prediction, while the uncertainty in the prediction itself (given by the softmax probabilities) is also evaluated to be low. To tackle this issue, several approaches described in Sect. 3 take the magnitude of the logits into account since a larger logit indicates larger evidence for the corresponding class (Sensoy et al. 2018). Thus, the methods either interpret the total sum of the (exponentials of) the logits as the precision value of a Dirichlet distribution (see description of Dirichlet Priors in Sect. 3.1) (Malinin and Gales 2018, 2019; Nandy et al. 2020), or as a collection of evidence that is compared to a defined constant (Sensoy et al. 2018; Możejko et al. 2018).

One can also derive a total class probability for each class individually by applying the sigmoid function to each logit (Hsu et al. 2020). Based on the class-wise total probabilities, OOD samples might easier be detected, since all classes can have low probability at the same time. Other methods deliver an explicit measure of how well new data samples suit the training data distribution. Based on this, they also give a measure that a sample will be predicted correctly (Ramalho and Miranda 2020).

4.1.4 Performance measure on complete data set

While the measures described above measure the performance of individual predictions, others evaluate the usage of these measures on a set of samples. Measures of uncertainty can be used to separate between correctly and falsely classified samples or between in-domain and OOD samples (Hendrycks and Gimpel 2017). For that, the samples are split into two sets, for example, in-domain and OOD or correctly classified and falsely classified. The two most common approaches are the *Receiver Operating Characteristic* (ROC) curve and the *Precision-Recall* (PR) curve. Both methods generate curves based on different thresholds of the underlying measure. For each considered threshold, the ROC curve plots the true positive rate against the false positive rate,[3] and the PR curve plots the precision against the recall.[4] While the ROC and PR curves give a visual idea of how well the underlying measures are suited to separate the two considered test cases, they do not give a qualitative measure. To reach this, the area under the curve (AUC) can be evaluated. Roughly speaking, the AUC gives a probability value that a randomly chosen positive sample leads to a higher measure than a randomly chosen negative example. For example, the maximum softmax values measure ranks of correctly classified examples higher than falsely classified examples. Hendrycks and Gimpel (2017) showed for several application fields that correct predictions have in general a higher predicted certainty in the softmax value than false predictions. Especially for the evaluation of in-domain and OOD examples, the *Area Under Receiver Operating Curve* (AUROC) and the *Area Under Precision-Recall Curce* (AUPRC) are commonly used (Nandy et al. 2020; Malinin and Gales 2018, 2019). The clear weakness of these evaluations is the fact that the performance is evaluated and the optimal threshold is computed based on a given test data set. A distribution shift from the test set distribution can ruin the whole performance and make the derived thresholds impractical.

4.2 Evaluating uncertainty in regression tasks

4.2.1 Measuring data uncertainty in regression predictions

In contrast to classification tasks, where the network typically outputs a probability distribution over the possible classes, regression tasks only predict a pointwise estimation without any hint of data uncertainty. As already described in Sect. 3, a common approach to overcome this is to let the network predict the parameters of a probability distribution, for

[3] The true positive rate is the number of samples, which are correctly predicted as positive divided by the total number of true samples. The false positive rate is the number of samples falsely predicted as positive divided by the total number of negative samples [see also (Davis and Goadrich 2006)].

[4] The precision is equal to the number of samples that are correctly classified as positive, divided by the total number of positive samples. The recall is equal to the number of samples correctly predicted as positive divided by the total number of positive samples [see also (Davis and Goadrich 2006)].

example, a mean vector and a standard deviation for a normally distributed uncertainty (Lakshminarayanan et al. 2017; Kendall and Gal 2017). In doing so, a measure of data uncertainty is directly given. The prediction of the standard deviation allows an analytical description that the (unknown) true value is within a specific region. The interval that covers the true value with a probability of α (under the assumption that the predicted distribution is correct) is given by

$$\left[\hat{y} - \frac{1}{2}\Phi^{-1}(\alpha)\cdot\sigma;\quad \hat{y} + \frac{1}{2}\Phi^{-1}(\alpha)\cdot\sigma\right] \tag{34}$$

where Φ^{-1} is the quantile function, the inverse of the cumulative probability function. For a given probability value α the quantile function gives a boundary, such that $100\cdot\alpha\%$ of a standard normal distribution's probability mass is on values smaller than $\Phi^{-1}(\alpha)$. Quantiles assume some probability distribution and interpret the given prediction as the expected value of the distribution.

In contrast to this, other approaches (Pearce et al. 2018; Su et al. 2018) directly predict a so-called prediction interval (PI)

$$PI(x) = \left[B_l, B_u\right] \tag{35}$$

in which the prediction is assumed to lay. Such intervals induce uncertainty as a uniform distribution without giving a concrete prediction. The certainty of such approaches can, as the name indicates, be directly measured by the size of the predicted interval. The *Mean Prediction Interval Width* (MPIW) can be used to evaluate the average certainty of the model (Pearce et al. 2018; Su et al. 2018). In order to evaluate the correctness of the predicted intervals the *Prediction Interval Coverage Probability* (PICP) can be applied (Pearce et al. 2018; Su et al. 2018). The PCIP represents the percentage of test predictions that fall into a prediction interval and is defined as

$$\text{PICP} = \frac{c}{n}, \tag{36}$$

where n is the total number of predictions and c is the number of ground truth values that are actually captured by the predicted intervals.

4.2.2 Measuring model uncertainty in regression predictions

In Sect. 2, it is described, that model uncertainty is mainly caused by the model's architecture, the training process, and underrepresented areas in the training data. Hence, there is no real difference in the causes and effects of model uncertainty between regression and classification tasks such that model uncertainty in regression tasks can be measured equivalently as already described for classification tasks, i.e. in most cases by approximating an average prediction and measuring the divergence among the single predictions (Kendall and Gal 2017).

4.3 Evaluating uncertainty in segmentation tasks

The evaluation of uncertainties in segmentation tasks is very similar to the evaluation of classification problems. The uncertainty is estimated in segmentation tasks using approximates of Bayesian inference (Nair et al. 2020; Roy et al. 2019; LaBonte et al. 2019;

Eaton-Rosen et al. 2018; McClure et al. 2019; Soleimany et al. 2019; Soberanis-Mukul et al. 2020; Seebock et al. 2020) or test-time data augmentation techniques (Wang et al. 2019). In the context of segmentation, the uncertainty in pixel-wise segmentation is measured using confidence intervals (LaBonte et al. 2019; Eaton-Rosen et al. 2018), the predictive variance (Soleimany et al. 2019; Seebock et al. 2020), the predictive entropy (Roy et al. 2019; Wang et al. 2019; McClure et al. 2019; Soberanis-Mukul et al. 2020) or the mutual information (Nair et al. 2020). The uncertainty in structure (volume) estimation is obtained by averaging over all pixel-wise uncertainty estimates (Seebock et al. 2020; McClure et al. 2019). The quality of volume uncertainties is assessed by evaluating the coefficient of variation, the average Dice score, or the intersection over union (Roy et al. 2019; Wang et al. 2019). These metrics measure the agreement in area overlap between multiple estimates in a pair-wise fashion. Ideally, a false segmentation should result in an increase in pixel-wise and structure uncertainty. To evaluate whether this is the case, (Nair et al. 2020) evaluated the pixel-level true positive rate and false detection rate as well as the ROC curves for the retained pixels at different uncertainty thresholds. Similar to Nair et al. (2020), McClure et al. (2019) also analyzed the area under the ROC curve.

5 Calibration

A predictor is called well-calibrated if the derived predictive confidence represents a good approximation of the actual probability of correctness (Guo et al. 2017). Therefore, in order to make use of uncertainty quantification methods, one has to be sure that the network is well calibrated. Formally, for classification tasks a neural network f_θ is calibrated (Kuleshov et al. 2018) if it holds that

$$\forall p \in [0,1]: \quad \sum_{i=1}^{N}\sum_{k=1}^{K} \frac{y_{i,k} \cdot \mathbb{1}\{f_\theta(x_i)_k = p\}}{\mathbb{1}\{f_\theta(x_i)_k = p\}} \xrightarrow{N\to\infty} p\,. \tag{37}$$

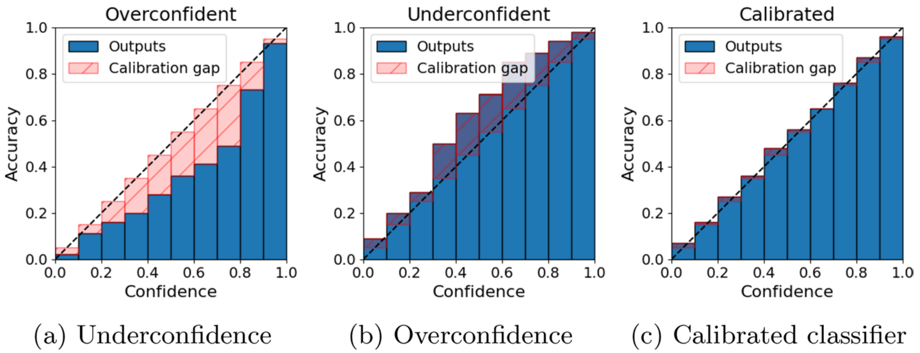
here $\mathbb{1}\{\cdot\}$ is the indicator function that is either 1 if the condition is true or 0 if it is false, and $y_{i,k}$ is the k-th entry in the one-hot encoded ground truth vector of a training sample (x_i, y_i). This formulation means that for example 30% of all predictions with a predictive confidence of 70% should actually be false. For regression tasks the calibration can be defined such that predicted confidence intervals should match the confidence intervals empirically computed from the data set (Kuleshov et al. 2018), i.e.

$$\forall p \in [0,1]: \quad \sum_{i=1}^{N} \frac{\mathbb{1}\left\{y_i \in \mathrm{conf}_p(f_\theta(x_i))\right\}}{N} \xrightarrow{N\to\infty} p, \tag{38}$$

where conf_p is the confidence interval that covers p percent of a distribution.

A DNN is called under-confident if the left-hand side of (37) and (38) are larger than p. Equivalently, it is under-confident if the terms are smaller than p. The calibration property of a DNN can be visualized using a *reliability diagram*, as shown in Fig. 8.

In general, calibration errors are caused by factors related to model uncertainty (Guo et al. 2017). This is intuitively clear, since as discussed in Sect. 2, data uncertainty represents the underlying uncertainty that an input x and a target y represent the same real world information. Following, correctly predicted data uncertainty would lead to a perfectly calibrated neural network. In practice, several works pointed out that deeper networks tend to



(a) Underconfidence (b) Overconfidence (c) Calibrated classifier

Fig. 8 **a** Reliability diagram showing an overconfident classifier: The bin-wise accuracy is smaller than the corresponding confidence. **b** Reliability diagram of an underconfident classifier: The bin-wise accuracy is larger than the corresponding confidence. **c** Reliability diagram of a well-calibrated classifier: The confidence fits the actual accuracy for the single bins

be more overconfident than shallower ones (Guo et al. 2017; Seo et al. 2019; Li and Hoiem 2020).

Several methods for uncertainty estimation presented in Sect. 3 also improve the network's calibration (Lakshminarayanan et al. 2017; Gal and Ghahramani 2016). This is clear since these methods quantify model and data uncertainty separately and aim at reducing the model uncertainty on the predictions. Besides the methods that improve calibration by reducing the model uncertainty, a large and growing body of literature has investigated methods for explicitly reducing calibration errors. These methods are presented in the following, followed by measures to quantify the calibration error. It is important to note that these methods do not reduce the model uncertainty, but propagate the model uncertainty onto the representation of the data uncertainty. For example, if a binary classifier is overfitted and predicts all samples of a test set as class A with probability 1, while half of the test samples are actually class B, the recalibration methods might map the network output to 0.5 in order to have reliable confidence. This probability of 0.5 is not equivalent to the data uncertainty but represents the model uncertainty propagated onto the predicted data uncertainty.

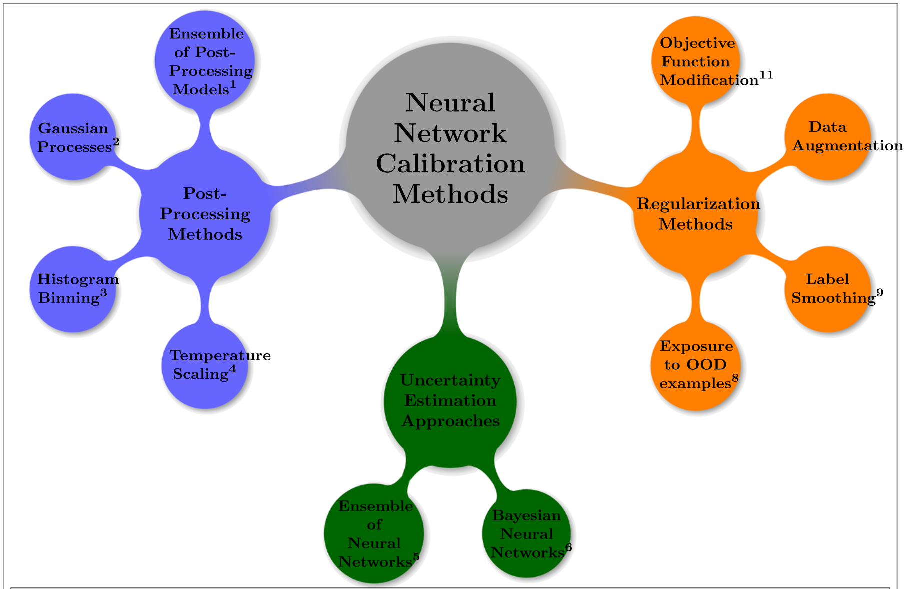
5.1 Calibration methods

Calibration methods can be classified into three main groups according to the step when they are applied:

- *Regularization methods applied during the training phase* (Szegedy et al. 2016; Pereyra et al. 2017; Lee et al. 2018a; Müller et al. 2019; Venkatesh and Thiagarajan 2019)

  These methods modify the objective, optimization, and/or regularization procedure in order to build DNNs that are inherently calibrated.
- *Post-processing methods applied after the training process of the DNN* (Guo et al. 2017; Wenger et al. 2020)

  These methods require a held-out calibration data set to adjust the prediction scores for recalibration. They only work under the assumption that the distribution of the left-


Neural Network Calibration Methods
Ensemble of Post-Processing Models[1]
Gaussian Processes[2]
Post-Processing Methods
Histogram Binning[3]
Temperature Scaling[4]
Uncertainty Estimation Approaches
Ensemble of Neural Networks[5]
Bayesian Neural Networks[6]
Objective Function Modification[11]
Data Augmentation[10]
Regularization Methods
Label Smoothing[9]
Exposure to OOD examples[8]

[1] (Zhang et al., 2020) [2] (Wenger et al., 2020) [3] (Zadrozny and Elkan, 2001) [4](Guo et al., 2017; Liang et al., 2018b; Laves et al., 2019) [5] (Lakshminarayanan et al., 2017; Mehrtash et al., 2020) [6] (Izmailov et al., 2020; Foong et al., 2019; Zhang et al., 2019; Laves et al., 2019; Wilson and Izmailov, 2020) [8] (Hendrycks et al., 2019) [9] (Szegedy et al., 2016; Müller et al., 2019) [10] (Thulasidasan et al., 2019; Maroñas et al., 2020; Rahaman et al., 2021; Patel et al., 2021) [11] (Pereyra et al., 2017; Seo et al., 2019; Lee et al., 2018a; Venkatesh and Thiagarajan, 2019; Maroñas et al., 2020)

Fig. 9 Visualization of the different types of uncertainty calibration methods presented in this paper

out validation set is equivalent to the distribution, on which inference is done. Hence, also the size of the validation data set can influence the calibration result.

- *Neural network uncertainty estimation methods*

  Approaches, as presented in Sect. 3, that reduce the amount of model uncertainty on a neural network's confidence prediction, also lead to a better calibrated predictor. This is because the remaining predicted data uncertainty better represents the actual uncertainty on the prediction. Such methods are based for example on Bayesian methods (Izmailov et al. 2020; Foong et al. 2019; Zhang et al. 2019; Laves et al. 2019; Wilson and Izmailov 2020) or deep ensembles (Lakshminarayanan et al. 2017; Mehrtash et al. 2020).

In the following, we present the three types of calibration methods in more detail (Fig. 9).

5.1.1 Regularization methods

Regularization methods for calibrating confidences manipulate the training of DNNs by modifying the objective function or by augmenting the training data set. The goal and idea of regularization methods are very similar to the methods presented in Sect. 3.1 where the methods mainly quantify model and data uncertainty separately within a single forward pass. However, the methods in Sect. 3.1 quantify the model and data uncertainty, while these calibration methods are regularized in order to minimize the model uncertainty. Following, at inference, the model uncertainty cannot be obtained anymore. This is the main motivation for us to separate the approaches presented below from the approaches presented in Sect. 3.1.

One popular regularization based calibration method is label smoothing (Szegedy et al. 2016). For label smoothing, the labels of the training examples are modified by taking a small portion α of the true class' probability mass and assigning it uniformly to the false classes. For hard, non-smoothed labels, the optimum cannot be reached in practice, as the gradient of the output with respect to the logit vector z,

$$\begin{aligned}\nabla_z \mathrm{CE}(y, \hat{y}(z)) &= \mathrm{softmax}(z) - y \\ &= \frac{\exp(z)}{\sum_{i=1}^{K} \exp(z_i)} - y\,,\end{aligned} \tag{39}$$

can only converge to zero with increasing distance between the true and false classes' logits. As a result, the logits of the correct class are much larger than the logits for the incorrect classes and the logits of the incorrect classes can be very different from each other. Label-smoothing avoids this and while it generally leads to a higher training loss, the calibration error decreases and the accuracy often increases as well (Müller et al. 2019).

Seo et al. (2019) extended the idea of label smoothing and directly aimed at reducing the model uncertainty. For this, they sampled T forward passes of a stochastic neural network already at training time. Based on the T forward passes of a training sample (x_i, y_i), a normalized model variance α_i is derived as the mean of the Bhattacharyya coefficients (Comaniciu et al. 2000) between the T individual predictions $\hat{y}_1, \dots, \hat{y}_T$ and the average prediction $\bar{y} = \frac{1}{T}\sum_{t=1}^{T} \hat{y}_t$,

$$\begin{aligned}\alpha_i &= \frac{1}{T}\sum_{t=1}^{T} BC(\bar{y}_i, \hat{y}_{i,t}) \\ &= \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K}\sqrt{\bar{y}_{i,k} \cdot \hat{y}_{i,t,k}}\,.\end{aligned} \tag{40}$$

Based on this α_i, (Seo et al. 2019) introduced the variance-weighted confidence-integrated loss function that is a convex combination of two contradictive loss functions,

$$L^{\mathrm{VWCI}}(\theta) = -\sum_{i=1}^{N}(1-\alpha_i)L_{\mathrm{GT}}^{(i)}(\theta) + \alpha_i L_{\mathrm{U}}^{(i)}(\theta)\,, \tag{41}$$

where $L_{\mathrm{GT}}^{(i)}$ is the mean cross-entropy computed for the training sample x_i with given ground-truth y_i. L_{U} represents the mean KL-divergence between a uniform target probability vector and the computed prediction. The adaptive smoothing parameter α_i pushes predictions of training samples with high model uncertainty (given by high variances) towards a uniform distribution while increasing the prediction scores of samples with low model uncertainty. As a result, variances in the predictions of a single sample are reduced and the network can then be applied with a single forward pass at inference.

Pereyra et al. (2017) combated the overconfidence issue by adding the negative entropy to the standard loss function and therefore a penalty that increases with the network's predicted confidence. This results in the entropy-based objective function L^H, which is defined as

$$L^H(\theta) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log \hat{y}_i - \alpha_i H(\hat{y}_i), \tag{42}$$

where $H(\hat{y}_i)$ is the entropy of the output and α_i is a parameter that controls the strength of the entropy-based confidence penalty. The parameter α_i is computed equivalently for the VWCI loss.

Instead of regularizing the training process by modifying the objective function, (Thulasidasan et al. 2019) regularized it by using a data-agnostic data augmentation technique named mixup (Zhang et al. 2018b). In mixup training, the network is not only trained on the training data but also on virtual training samples $(\tilde{x}, \tilde{y})$ generated by a convex combination of two random training pairs (x_i, y_i) and (x_j, y_j), i.e.

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j \tag{43}$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j. \tag{44}$$

According to Thulasidasan et al. (2019), the label smoothing resulting from mixup training can be viewed as a form of entropy-based regularization resulting in the inherent calibration of networks trained with mixup. Maroñas et al. (2020) see mixup training among the most popular data augmentation regularization techniques due to its ability to improve the calibration as well as the accuracy. However, they argued that in mixup training the data uncertainty in mixed inputs affects the calibration and therefore mixup does not necessarily improve the calibration. They also underlined this claim empirically. Similarly, Rahaman et al. (2021) experimentally showed that the distributional-shift induced by data augmentation techniques such as mixup training can negatively affect the confidence calibration. Based on this observation, (Maroñas et al. 2020) proposed a new objective function that explicitly takes the calibration performance on the unmixed input samples into account. Inspired by the expected calibration error (ECE, see Sect. 5.2) (Naeini et al. 2015) measured the calibration performance on the unmixed samples for each batch b by the differentiable squared differences between the batch accuracy and the mean confidence on the batch samples. The total loss is given as a weighted combination of the original loss on mixed and unmixed samples and the calibration measure is evaluated only on the unmixed samples:

$$L^{ECE}(\theta) = \frac{1}{B}\sum_{b \in B} L^b(\theta) + \beta ECE_b, \tag{45}$$

where $L^b(\theta)$ is the original unregularized loss using training and mixed samples included in batch b and β is a hyperparameter controlling the relative importance given to the batchwise expected calibration error ECE_b. By adding the batchwise calibration error for each batch $b \in B$ to the standard loss function, the miscalibration induced by mixup training is regularized.

In the context of data augmentation, (Patel et al. 2021) improved the calibration of uncertainty estimates by using on-manifold data augmentation. While mixup training combines training samples, on-manifold adversarial training generates out-of-domain samples using adversarial attack. They experimentally showed that on-manifold adversarial training outperforms mixup training in improving the calibration. Similar to Patel et al. (2021), Hendrycks et al. (2019) showed that exposing classifiers to OOD examples at training can help to improve the calibration.

5.1.2 Post-processing methods

Post-processing (or post-hoc) methods are applied after the training process and aim at learning a re-calibration function. For this, a subset of the training data is held-out during the training process and used as a calibration set. The re-calibration function is applied to the network's outputs (e.g. the logit vector) and yields an improved calibration learned on the left-out calibration set. Zhang et al. (2020) discussed three requirements that should be satisfied by post-hoc calibration methods. They should

1. preserve the accuracy, i.e. should not affect the predictor's performance.
2. be data efficient, i.e. only a small fraction of the training data set should be left out for the calibration.
3. be able to approximate the correct re-calibration map as long as there is enough data available for calibration.

Furthermore, they pointed out that none of the existing approaches fulfills all three requirements.

For classification tasks, the most basic but still very efficient way of post-hoc calibration is temperature scaling (Guo et al. 2017). For temperature scaling, the temperature $T > 0$ of the softmax function

$$\text{softmax}(z_i) = \frac{\exp^{z_i/T}}{\sum_{j=1}^{K} \exp^{z_j/T}}, \tag{46}$$

is optimized. For $T = 1$ the function remains the regular softmax function. For $T > 1$ the output changes such that its entropy increases, i.e. the predicted confidence decreases. For $T \in (0, 1)$ the entropy decreases and following, the predicted confidence increases. As already mentioned above, a perfectly calibrated neural network outputs MAP estimates. Since the learned transformation can only affect the uncertainty, the log-likelihood based losses as cross-entropy do not have to be replaced by a special calibration loss. While the data efficiency and the preservation of the accuracy are given, the expressiveness of basic temperature scaling is limited (Zhang et al. 2020). To overcome this, (Zhang et al. 2020) investigated an ensemble of several temperature scaling models. Doing so, they achieved better calibrated predictions, while preserving the classification accuracy and improving the data efficiency and the expressive power. Kull et al. (2019) were motivated by non-neural network calibration methods, where the calibration is performed class-wise as a one-vs-all binary calibration. They showed that this approach can be interpreted as learning a linear transformation of the predicted log-likelihoods followed by a softmax function. This again is equivalent to training a dense layer on the log-probabilities and hence the method is also very easy to implement and apply. Obviously, the original predictions are not guaranteed to be preserved.

Analogous to temperature scaling for classification networks, Levi et al. (2022) introduced standard deviation scaling (std-scaling) for regression networks. As the name indicates, the method is trained to rescale the predicted standard deviations of a given network. Equivalently to the motivation of optimizing temperature scaling with the cross-entropy loss, std-scaling can be trained using the Gaussian log-likelihood function as loss, which is in general also used for the training of regression networks, which also gives a prediction for the data uncertainty.

Wenger et al. (2020) proposed a Gaussian process (GP) based method, which can be used to calibrate any multi-class classifier that outputs confidence values and presented their methodology by calibrating neural networks. The main idea behind their work is to learn the calibration map by a Gaussian process that is trained on the networks' confidence predictions and the corresponding ground-truths in the left-out calibration set. For this approach, the preservation of the predictions is also not assured.

5.1.3 Calibration with uncertainty estimation approaches

As already discussed above, removing the model uncertainty and receiving an accurate estimation of the data uncertainty leads to a well-calibrated predictor. Following several works based on deep ensembles (Lakshminarayanan et al. 2017; Mehrtash et al. 2020) and BNNs, (Izmailov et al. 2020; Foong et al. 2019; Kristiadi et al. 2020) also compared their performance to other methods based on the resulting calibration. Lakshminarayanan et al. (2017) and Mehrtash et al. (2020) reported an improved calibration by applying deep ensembles compared to single networks. However, Rahaman et al. (2021) showed that for specific configurations as the usage of mixup-regularization, deep ensembles can even increase the calibration error. On the other side, they showed that applying temperature scaling on the averaged predictions can give a significant improvement on the calibration.

For the Bayesian approaches, (Kristiadi et al. 2020) showed that restricting the Bayesian approximation to the weights of the last fully connected layer of a DNN is already enough to improve the calibration significantly. Zhang et al. (2019) and Laves et al. (2019) showed that confidence estimates computed with MC dropout can be poorly calibrated. To overcome this, (Zhang et al. 2019) proposed structured dropout, which consists of dropping channels, blocks, or layers, to promote model diversity and reduce calibration errors.

5.2 Evaluating calibration quality

Evaluating calibration consists of measuring the statistical consistency between the predictive distributions and the observations (Vaicenavicius et al. 2019). For classification tasks, several calibration measures are based on binning. For that, the predictions are ordered by the predicted confidence $\hat{p}_i$ and grouped into M bins $b_1, \dots, b_M$. Following, the calibration of the single bins is evaluated by setting the average bin confidence

$$\mathrm{conf}(b_m) = \frac{1}{|b_m|} \sum_{s \in b_m} \hat{p}_s \tag{47}$$

in relation to the average bin accuracy

$$\mathrm{acc}(b_m) = \frac{1}{|b_m|} \sum_{s \in b_m} \mathbb{1}(\hat{y}_s = y_s), \tag{48}$$

where $\hat{y}_s$, y_s, and $\hat{p}_s$ refer to the predicted and true class label of a sample s. As noted in Guo et al. (2017), confidences are well-calibrated when for each bin $\mathrm{acc}(b_m) = \mathrm{conf}(b_m)$. For a visual evaluation of a model's calibration, the reliability diagram introduced by DeGroot and Fienberg (1983) is widely used. For a reliability diagram, the $\mathrm{conf}(b_m)$ is plotted against $\mathrm{acc}(b_m)$. For a well-calibrated model, the plot should be close to the diagonal, as visualized in Fig. 8. The basic reliability diagram visualization does not distinguish between different classes. In order to do so and hence to improve the interpretability of the calibration error, (Vaicenavicius et al. 2019) used an alternative visualization named multi-dimensional reliability diagram.

For a quantitative evaluation of a model's calibration, different calibration measures can be considered. The *Expected Calibration Error* (ECE) is a widely used binning-based calibration measure (Naeini et al. 2015; Guo et al. 2017; Laves et al. 2019; Mehrtash et al. 2020; Thulasidasan et al. 2019; Wenger et al. 2020). For the ECE, M equally-spaced bins $b_1, \dots, b_M$ are considered, where b_m denotes the set of indices of samples whose confidences fall into the interval $I_m = [\frac{m-1}{M}, \frac{m}{M}]$. The ECE is then computed as the weighted average of the bin-wise calibration errors, i.e.

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|b_m|}{N} |\mathrm{acc}(b_m) - \mathrm{conf}(b_m)|. \tag{49}$$

For the ECE, only the predicted confidence score (top-label) is considered. In contrast to this, the *Static Calibration Error* (SCE) (Nixon et al. 2019; Ghandeharioun et al. 2019) considers the predictions of all classes (all-labels). For each class, the SCE computes the calibration error within the bins and then averages across all the bins, i.e.

$$\mathrm{SCE} = \frac{1}{K} \sum_{k=1}^{K} \sum_{m=1}^{M} \frac{|b_{m_k}|}{N} |\mathrm{conf}(b_{m_k}) - \mathrm{acc}(b_{m_k})|. \tag{50}$$

here $conf(b_{m_k})$ and $acc(b_{m_k})$ are the confidence and accuracy of bin b_m for class label k, respectively. Nixon et al. (2019) empirically showed that all-label calibration measures such as the SCE are more effective in assessing the calibration error than the top-label calibration measures as the ECE.

In contrast to the ECE and SCE, which group predictions into M equally-spaced bins (what in general leads to different numbers of evaluation samples per bin), the adaptive calibration error (Nixon et al. 2019; Ghandeharioun et al. 2019) adaptively groups predictions into R bins with different widths but the equal number of predictions. With this adaptive bin size, the *adaptive Expected Calibration Error* (aECE)

$$\mathrm{aECE} = \frac{1}{R} \sum_{r=1}^{R} |\mathrm{conf}(b_r) - \mathrm{acc}(b_r)|, \tag{51}$$

and the *adaptive Static Calibration Error* (aSCE)

$$\mathrm{aSCE} = \frac{1}{KR} \sum_{k=1}^{K} \sum_{r=1}^{R} |\mathrm{conf}(b_{r_k}) - \mathrm{acc}(b_{r_k})| \tag{52}$$

are defined as extensions of the ECE and the SCE. As has been empirically shown in Patel et al. (2021) and Nixon et al. (2019), the adaptive binning calibration measures aECE and aSCE are more robust to the number of bins than the corresponding equal-width binning calibration measures ECE and SCE.

It is important to make clear that in a multi-class setting, the calibration measures can suffer from imbalance in the test data. Even when then calibration is computed classwise, the computed errors are weighted by the number of samples in the classes. Following, larger classes can shadow the bad calibration on small classes, comparable to accuracy values in classification tasks (Pulgar et al. 2017).

6 Data sets and baselines

In this section, we collect commonly used tasks and data sets for evaluating uncertainty estimation among existing works. Besides, a variety of baseline approaches commonly used as a comparison against the methods proposed by the researchers are also presented. By providing a review on the relevant information of these experiments, we hope that both researchers and practitioners can benefit from it. While the former can gain a basic understanding of recent benchmarks tasks, data sets, and baselines so that they can design appropriate experiments to validate their ideas more efficiently, the latter might use the provided information to select more relevant approaches to start based on a concise overview on the tasks and data sets on which the approach has been validated.

In the following, we will introduce the data sets and baselines summarized in Table 4 according to the taxonomy used throughout this review.

The structure of the table is designed to organize the main contents of this section concisely, hoping to provide a clear overview of the relevant works. We group the approaches of each category into one of four blocks and extract the most commonly used tasks, data sets, and provided baselines for each column respectively. The corresponding literature is listed at the bottom of each block to facilitate lookup. Note that we focus on methodological comparison here, but not the choice of architecture for different methods which has an impact on performance as well. Due to the space limitation and visual density, we only show the most important elements (task, data set, baselines) ranked according to the frequency of use in the literature we have researched.

The main results are as follows. One of the most frequent tasks for evaluating uncertainty estimation methods is the regression task, where samples close and far away from the training distribution are studied. Furthermore, the calibration of uncertainty estimates in the case of classification problems is very often investigated. Further noteworthy tasks are OOD detection and robustness against adversarial attacks. In the medical domain, calibration of semantic segmentation results is the predominant use case.

The choice of data sets is mostly consistent among all reviewed works. For regression, toy data sets are employed for visualization of uncertainty intervals while the UCI (Dua and Graff 2017) data sets are studied in light of (negative) log-likelihood comparison. The most common data sets for calibration and OOD detection are MNIST (LeCun et al. 1998; Deng 2012), CIFAR10 and CAFIAR100 (Krizhevsky 2009) as well as SVHN (Netzer et al. 2011) while ImageNet (Deng et al. 2009) and its tiny variant are also studied frequently. These form distinct pairs when OOD detection is studied where models trained on CIFAR variants are evaluated on SVHN and visa versa while MNIST is paired with variants of itself like notMNIST and FashionMNIST (Xiao et al. 2017). Classification data sets are also commonly distorted and corrupted to study the effects on calibration, blurring the line between OOD detection and adversarial attacks.

Finally, the most commonly used baselines by far are Monte Carlo (MC) Dropout and deep ensembles while the softmax output of deterministic models is almost always employed as a kind of surrogate baseline. It is interesting to note that inside each approach–BNNs, Ensembles, Single Deterministic Models, and Input Augmentation–some baselines are preferred over others. BNNs are most frequently compared against variational inference methods like Bayes' by Backprop (BBB) or Probabilistic Backpropagation (PBP) while for Single Deterministic Models it is more common to

Table 4 Overview of frequently compared benchmark approaches, tasks and their data sets among existing works organized according to the taxonomy of this paper. From left to right, the columns indicate the approach considered, the tasks evaluated with the corresponding approaches, the data sets used for this evaluation, and the baselines commonly used in these works

	Tasks	Task index: data sets	Baselines
Bayesian neural networks[1–17]	1. Regression[1,2,3,4,5,6,7,8,9,10] 2. Calibration[11,12,13,14] 3. OOD Detection[3,5,7,11,12,13] 4. Adversarial Attacks[3,5,7] 5. Active Learning[4,7,14] 6. Continual Learning[12] 7. Reinforcement Learning (Intrinsic Motivation, Contextual Bandits)[1,14,8]	1. UCI 2., 3., 4. (not)MNIST, CIFAR10/100, SHVN, ImageNet 5. UCI 6. Permuted MNIST	Softmax[42], MCdropout[1], DeepEnsemble[17], NormalizingFlow[3], BBB[2], PBP[4], SWAG[11], KFAC[5], DVI[6], HMC[15], VOGN[12], INF[7]
Ensembles[18–30]	1. Regression[18,19] 2. Calibration[18,19,20,21,22,23,24,25,26,27,28] 3. OOD Detection[18,19,21,24,25,26,28,29,30] 4. Active Learning[27]	1. Toy dataset[4], UCI 2., 3. Toy dataset[4], (not) MNIST, SVHN, LSUN, CIFAR10/100, (Tiny)ImageNet, Diabetic Retinopathy 4. MNIST	Softmax[43], MCdropout[1]. DeepEnsemble[18], NormalizingFlow[3], BBB[2], PBP[4], SGLD[16], MFVI[17], TemperatureScaling[39, 53]
Single deterministic models[32–54]	1. Regression[32,33,34] 2. Calibration[32,33,34,35,36,37,38] 3. OOD Detection[33,39,38,39,40,41,42,43,44,45,46,47,48,49,50,51] 4. Adversarial Attacks[32,37,47]	1. Toy dataset[4], UCI, NYU Depth 2., 3. (E/Fashion/not)MNIST, Toy dataset, CIFAR10/100, SVHN, LSUN, (Tiny) ImageNet, IMDB, Diabetic Retinopathy, Omniglot 4. MNIST, CIFAR10, NYU Depth, Omniglot	Softmax[43], MCdropout[1], DeepEnsemble[18,31], NormalizingFlow[3], BBB[2], DPN[23], Dirichlet[41], Mahalanobis[52], TemperatureScaling[39,53]
Test-time data augmen-tation[54–56]	1. Semantic Segmentation[54,55] 2. Calibration[56] 3. OOD Detection[54,55,56]	1., 2., 3. Medical data, Diabetic Retinopathy	Softmax[43], MCdropout[1]

[1]Gal and Ghahramani (2016), [2]Blundell et al. (2015), [3]Louizos and Welling (2017), [4]Hernández-Lobato and Adams (2015), [5]Ritter et al. (2018), [6]Wu et al. (2018), [7]Lee et al. (2020), [8]Sun et al. (2018), [9]Sun et al. (2017), [10]Izmailov et al. (2020), [11]Maddox et al. (2019), [12]Osawa et al. (2019), [13]Wen et al. (2021b), [14]Zhang et al. (2018a), [15]Neal (1995), [16]Welling and Teh (2011), [17]Graves (2011), [18]Lakshminarayanan et al. (2017), [19]Lindqvist et al. (2020), [20]Rahaman et al. (2021), [21]Achrack et al. (2020), [22]Huang et al. (2017), [23]Malinin et al. (2020), [24]Valdenegro-Toro (2019), [25]Wen et al. (2021b), [26]Wen et al. (2019), [27]Beluch et al. (2018), [28]Ovadia et al. (2019), [29]Vyas et al. (2018), [30]Englesson and Azizpour (2019), [31]Pearce et al. (2018), [32]Amini et al. (2020), [33]Tagasovska and Lopez-Paz (2019), [34]Kawashima et al. (2021), [35]Wu et al. (2018), [36]Tsiligkaridis (2021a), [37]Tsiligkaridis (2021b), [38]Vasudevan et al. (2019), [39]Liang et al. (2018b), [40]Malinin and Gales (2018), [41]Malinin and Gales (2019), [42]Hein et al. (2019), [43]Hendrycks and Gimpel (2017), [44]Nandy et al. (2020), [45]Możejko et al. (2018), [46]Lee and AlRegib (2020), [47]Sensoy et al. (2018), [48]Van Amersfoort et al. (2020), [49]Ramalho and Miranda (2020), [50]Raghu et al. (2019), [51]Oberdiek et al. (2018), [52]Lee et al. (2018b), [53]Guo et al. (2017), [54]Wang et al. (2018a), [55]Wang et al. (2019), [56]Ayhan and Berens (2018)

compare them against distance-based methods in the case of OOD detection. Overall, BNN methods show a more diverse set of tasks considered while being less frequently evaluated on large data sets like ImageNet.

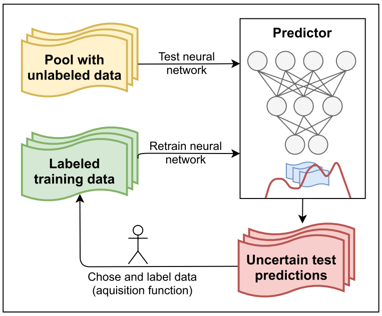


Fig. 10 The active learning framework: The acquisition function evaluates the uncertainties on the network's test predictions in order to select unlabelled data. The selected data are labelled and added to the pool of labelled data, which is used to train and improve the performance of the predictor

7 Applications of uncertainty estimates

From a practical point of view, the main motivation for quantifying uncertainties in DNNs is to be able to classify the received predictions and to make more confident decisions. This section gives a brief overview and examples of the aforementioned motivations. In the first part, we discuss how uncertainty is used within active learning and reinforcement learning. Subsequently, we discuss the interest of the communities working on domain fields like medical image analysis, robotics, and earth observation. These application fields are used representatively for the large number of domains where uncertainty quantification plays an important role. The challenges and concepts could (and should) be transferred to any application domain of interest.

7.1 Active learning

The process of collecting labeled data for supervised training of a DNN can be laborious, time-consuming, and costly. To reduce the annotation effort, the active learning framework shown in Fig. 10 trains the DNN sequentially on differently labelled data sets increasing in size over time (Iuzzolino et al. 2020). In particular, given a small labelled data set and a large unlabeled data set, a deep neural network trained in the setting of active learning learns from the small labeled data set and decides based on the acquisition function, which samples to select from the pool of unlabeled data. The selected data are added to the training data set and a new DNN is trained on the updated training data set. This process is then repeated with the training set increasing in size over time. Uncertainty sampling is one most popular criteria used in acquisition functions (Settles 2009) where predictive uncertainty determines which training samples have the highest uncertainty and should be labelled next. Uncertainty-based active learning strategies for deep learning applications have been successfully used in several works (Gal et al. 2017b; Chitta et al. 2018; Pop and Fulop 2018; Zeng et al. 2018; Nguyen et al. 2019; Feng et al. 2021).

7.2 Reinforcement learning

The general framework of deep reinforcement learning is shown in Fig. 11. In the context of reinforcement learning, uncertainty estimates can be used to solve the exploration-exploitation dilemma. It says that uncertainty estimates can be used to effectively balance

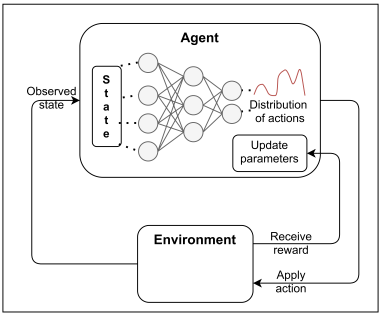


Fig. 11 The reinforcement learning framework: The agent interacts with the environment by executing a specific action influencing the next state of the agent. The agent observes a reward representing the cost associated with the executed action. The agent chooses actions based on a policy learned by a deep neural network. However, the predicted uncertainty associated with the action predicted by the deep neural network can help the agent to decide whether to execute the predicted action or not

the exploration of unknown environments with the exploitation of existing knowledge extracted from known environments. For example, if a robot interacts with an unknown environment, the robot can safely avoid catastrophic failures by reasoning about its uncertainty. To estimate the uncertainty in this framework, (Huang et al. 2019a) used an ensemble of bootstrapped models (models trained on different data sets sampled with replacement from the original data set), while Gal and Ghahramani (2016) approximated Bayesian inference via dropout sampling. Inspired by Gal and Ghahramani (2016) and Huang et al. (2019a), Kahn et al. (2017) and Lütjens et al. (2019) used a mixture of deep Bayesian networks performing dropout sampling on an ensemble of bootstrapped models. For further reading, (Ghavamzadeh et al. 2015) presented a survey of Bayesian reinforcement learning.

7.3 Uncertainty in real-world applications

With the increasing usage of deep learning approaches within many different fields, quantifying and handling uncertainties has become more and more important. On one hand, uncertainty quantification plays an important role in risk minimization, which is needed in many application fields. On the other hand, many fields offer only challenging data sources, which are hard to control and verify. This makes the generation of trust-worthy ground truth a very challenging task. In the following, three different fields where uncertainty plays an important role are presented, namely robotics including autonomous driving, medical image analysis, and earth observation.

7.3.1 Medical analysis

As there is a significant amount of uncertainty in medical data, it is essential for machine learning approaches in this field to provide reliable predictions and confidence values in order to gain trust from physicians and patients (Abdullah et al. 2022; Begoli et al. 2019; Loftus et al. 2022). For medical applications involving clinical background data, distribution shifts are quite common. This is because medical information is highly confidential, and machine learning models are often trained on data from different sources (e.g., varying patient backgrounds or measurement systems) than those they are applied to later (Koh et al. 2021). This becomes even more relevant when the data samples are highly individualized, such as in the case of patients' electronic health records (EHR). Gianfrancesco et al. (2018) points out that EHRs can be potentially biased due to missing data, small sample sizes for specific subgroups, and misclassification or measurement errors in the data. Consequently, quantifying epistemic uncertainty is of critical importance in this field. Heo et al. (2018) and Chen et al. (2021) both employ variational inference to represent epistemic uncertainty, demonstrating its effectiveness on various risk prediction tasks based on EHR. Qiu et al. (2019) also utilize Bayesian neural networks to model epistemic uncertainty, confirming that an increased level of noise adversely affects predictive performance but simultaneously increases the predicted uncertainty. Dusenberry et al. (2020) examine the role of epistemic uncertainty in patient mortality prediction and disease classification based on electronic health records. They reveal that Bayesian neural network designs can capture epistemic uncertainty more efficiently than ensembles for these tasks.

Another crucial area of application for uncertainty quantification in medical deep learning is the detection of diseases across various types of image data (Yang and Fevens 2021). Since the size, shape, and location of many diseases vary largely across patients, the estimation of the predictive uncertainty is crucial in analyzing medical images in applications such as lesion detection (Nair et al. 2020; Seebock et al. 2020), lung node segmentation (Hu et al. 2019), brain tumor segmentation (Eaton-Rosen et al. 2018; Wang et al. 2018a, 2019; Roy et al. 2019; McClure et al. 2019), parasite segmentation in images of liver stage malaria (Soleimany et al. 2019), recognition of abnormalities on chest radiographs (Ghesu et al. 2019), and age estimation (Eggenreich et al. 2020). Here, uncertainty estimates in particular improve the interpretability of decisions of DNNs (Ayhan et al. 2020). They are essential to understand the reliability of segmentation results, to detect false segmented areas, and to guide human experts in the task of refinement (Wang et al. 2019). Well-calibrated and reliable uncertainty estimates allow clinical experts to properly judge whether an automated diagnosis can be trusted (Ayhan et al. 2020). Uncertainty was estimated in medical image segmentation based on Monte Carlo dropout (Eaton-Rosen et al. 2018; Hu et al. 2019; Nair et al. 2020; Roy et al. 2019; Seebock et al. 2020; Soberanis-Mukul et al. 2020; LaBonte et al. 2019; Reinhold et al. 2020; Eggenreich et al. 2020), spike-and-slab dropout (McClure et al. 2019), and spatial dropout (Soleimany et al. 2019). (Wang et al. 2018a, 2019) used test-time augmentation to estimate the data-dependent uncertainty in medical image segmentation.

The tasks discussed thus far have primarily focused on clinical applications; however, approaches with a medical background can also be found in areas such as drug design and evaluation. For instance, the classification of molecular properties and drug discovery are relevant applications. Ghoshal et al. (2021) incorporate MC dropout into B-cell epitope prediction, which could potentially be applied in the evaluation of vaccine candidates. Several studies in this field employ Monte Carlo dropout (Kim et al. 2021; Semenova et al. 2020), and they have already demonstrated that incorporating epistemic uncertainty in predictions can lead to more reliable outcomes compared to deterministic approaches. Scalia et al. (2020) compares different uncertainty quantification methods in molecular property prediction tasks, showing that deep ensembles outperform MC dropout-based approaches. Liang et al. (2018a) proposes a Markov-Chain Monte Carlo-based approach for identifying genes associated with anti-cancer drug sensitivities.

7.3.2 Robotics

Robots are active agents that perceive, decide, plan, and act in the real world – all based on their incomplete knowledge about the world. As a result, mistakes of the robots not only cause failures of their own mission but can endanger human lives, e.g. in the case of surgical robotics, self-driving cars, space robotics, etc. Hence, the robotics application of deep learning poses unique research challenges that significantly differ from those often addressed in computer vision and other off-line settings (Sünderhauf et al. 2018). For example, the assumption that the testing condition comes from the same distribution as training is often invalid in many settings of robotics, resulting in deterioration of the performance of DNNs in uncontrolled and detrimental conditions. This raises the question how we can quantify the uncertainty in a DNN's predictions in order to avoid catastrophic failures. Answering such questions are important in robotics, as it might be a lofty goal to expect data-driven approaches (in many aspects from control to perception) to always be accurate. Instead, reasoning about uncertainty can help in leveraging the recent advances in deep learning for robotics.

Reasoning about uncertainties and the use of probabilistic representations, as opposed to relying on a single, most-likely estimate, have been central to many domains of robotics research, even before the advent of deep learning (Thrun 2002). In robot perception, several uncertainty-aware methods have been proposed in the past, starting from localization methods (Fox 1998; Fox et al. 2000; Thrun et al. 2001) to simultaneous localization and mapping (SLAM) frameworks (Durrant-Whyte and Bailey 2006; Bailey and Durrant-Whyte 2006; Montemerlo et al. 2002; Kaess et al. 2010). As a result, many probabilistic methods such as factor graphs (Dellaert et al. 2017; Loeliger 2004) are now the work-horse of advanced consumer products such as robotic vacuum cleaners and unmanned aerial vehicles. In the case of planning and control, estimation problems are widely treated as Bayesian sequential learning problems, and sequential decision-making frameworks such as POMDPs (Silver and Veness 2010; Ross et al. 2008) assume a probabilistic treatment of the underlying planning problems. With probabilistic representations, many reinforcement learning algorithms are backed up by stability guarantees for safe interactions in the real world (Richards et al. 2018; Berkenkamp et al. 2016, 2017). Lastly, there have been also several advances starting from reasoning (semantics (Grimmett et al. 2016) to joint reasoning with geometry), embodiment [e.g. active perception (Bajcsy 1988)] to learning [e.g. active learning (Triebel et al. 2016; Narr et al. 2016; Cohn et al. 1996) and identifying unknown objects (Nguyen et al. 2015; Wong et al. 2020; Boerdijk et al. 2021)].

Similarly, with the advent of deep learning, many researchers proposed new methods to quantify the uncertainty in deep learning as well as on how to further exploit such information. As opposed to many generic approaches, we summarize task-specific methods and their application in practice as followings. Notably, (Richter and Roy 2017) proposed to perform novelty detection using auto-encoders, where the reconstructed outputs of auto-encoders were used to decide how much one can trust the network's predictions. Peretroukhin et al. (2020) developed a SO(3) representation and uncertainty estimation framework for the problem of rotational learning problems with uncertainty. Lütjens et al. (2019), Kahn et al. (2017), Kahn et al. (2018), Stulp et al. (2011) demonstrated uncertainty-aware, real world application of a reinforcement learning algorithm for robotics, while (Tchuiev and Indelman 2018; Feldman and Indelman 2018) proposed to leverage spatial information, on top of MC-dropout. Shinde et al. (2020), Yang et al. (2020), Wang et al. (2017) developed deep learning based localization systems along with uncertainty estimates.

Other approaches also learn from the robots' past experiences of failures or detect inconsistencies of the predictors (Gurău et al. 2016; Daftry et al. 2016). In summary, the robotics community has been both, the users and the developers of the uncertainty estimation frameworks targeted to the specific problems.

Yet, robotics poses several unique challenges to uncertainty estimation methods for DNNs. These are for example, (i) how to limit the computational burden and build real-time capable methods that can be executed on the robots with limited computational capacities (e.g. aerial, space robots, etc); (ii) how to leverage spatial and temporal information, as robots sense sequentially instead of having a batch of training data for uncertainty estimates; (iii) whether robots can select the most uncertainty samples and update its learner online; (iv) Whether robots can purposefully manipulate the scene when uncertain. Most of these challenges arise due to the properties of robots that they are physically situated systems.

7.3.3 Earth observation (EO)

Earth Observation (EO) systems are increasingly used to make critical decisions related to urban planning (Netzband et al. 2007), resource management (Giardino et al. 2010), disaster response (Van Westen 2000), and many more. Right now, there are hundreds of EO satellites in space, owned by different space agencies and private companies. Like in many other domains, deep learning has shown great initial success in the field of EO over the past few years (Zhu et al. 2017). These early successes consisted of taking the latest developments of deep learning in computer vision and applying them to small curated earth observation data sets (Zhu et al. 2017). At the same time, the underlying data is very challenging. Even though the amount of data is huge, so is the variability in the data. This variability is caused by different sensor types, spatial changes (e.g. different regions and resolutions), and temporal changes (e.g. changing light conditions, weather conditions, seasons). Besides the challenge of efficient uncertainty quantification methods for such large amounts of data, several other challenges that can be tackled with uncertainty quantification exist in the field of EO. All in all, the sensitivity of many EO applications together with the nature of EO systems and the challenging EO data make the quantification of uncertainties very important in this field. Despite hundreds of publications in the last years on DL for EO, the range of literature on measuring the uncertainties of these systems is relatively small.

Furthermore, due to the large variation in the data, a data sample received at test time is often not covered by the training data distribution. For example, while preparing training data for a local climate zone classification, the human experts might be presented only with images where there is no obstruction and structures are clearly visible. When a model which is trained on this data set is deployed in the real world, it might see images with clouds obstructing the structures or snow giving them a completely different look. Also, the classes in EO data can have a very wide distribution. For example, there are millions of types of houses in the world and no training data can contain the examples for all of them. The question is where the OOD detector will draw the line and declare the following houses as OOD. Hence, OOD detection is important in earth observation and uncertainty measurements play an important part in this (Gawlikowski et al. 2022).

Another common task in EO, where uncertainties can play an important role, is data fusion. Optical images normally contain only a few channels like RGB. In contrast to this, EO data can contain optical images with up to hundreds of channels, and a variety of different sensors with different spatial, temporal, and semantic properties. Fusing the information from these different sources and channels propagates the uncertainties from different sources into the prediction. The challenge lies in developing methods that do not only quantify uncertainties but also the amount of contribution from different channels individually and which learn to focus on the trustworthy data source for a given sample (Schmitt and Zhu 2016).

Unlike normal computer vision scenarios where the image acquisition equipment is quite near to the subject, the EO satellites are hundreds of kilometers away from the subject. The sensitivity of sensors, atmospheric absorption properties, and surface reflectance properties all contribute to uncertainties in the acquired data. Integrating the knowledge of physical EO systems, which also contain information about uncertainty models in those systems, is another major open issue. However, for several applications in EO, measuring uncertainties is not only something good to have but rather an important requirement of the field. E.g., the geo-variables derived from EO data may be assimilated into process models (ocean, hydrological, weather, climate, etc) and the assimilation requires the probability distribution of the estimated variables.

8 Conclusion and outlook

8.1 Conclusion—how well do the current uncertainty quantification methods work for real world applications?

Even though many advances on uncertainty quantification in neural networks have been made over the last years, their adoption in practical mission- and safety-critical applications is still limited. There are several reasons for this, which are discussed one by one as follows:

- Missing validation of existing methods over real-world problems

  Although DNNs have become the de facto standard in solving numerous computer vision and medical image processing tasks, the majority of existing models are not able to appropriately quantify the uncertainty that is inherent to their inferences, particularly in real world applications. This is primarily because the baseline models are mostly developed using standard data sets such as Cifar10/100, ImageNet, or well-known regression data sets that are specific to a particular use case and are therefore not readily applicable to complex real-world environments, such as low-resolution satellite data or other data sources affected by noise. Although many researchers from other fields apply uncertainty quantification in their field (Rußwurm et al. 2020; Loquercio et al. 2020; Choi et al. 2019), a broad and structured evaluation of existing methods based on different real world applications is not available yet. Works like (Gustafsson et al. 2020) already built the first step towards a real life evaluation.
- Lack of standardized evaluation protocol

  Existing methods for evaluating the estimated uncertainty are better suited to compare uncertainty quantification methods based on measurable quantities such as the calibration (Nado et al. 2021) or the performance on OOD detection (Malinin and Gales 2018). As described in Sect. 6, these tests are performed on standardized sets within the machine learning community. Furthermore, the details of these experiments might differ in the experimental setting from paper to paper (Mukhoti et al.

2018). However, a clear standardized protocol of tests that should be performed on uncertainty quantification methods is still not available. For researchers from other domains, it is difficult to directly find state-of-the-art methods for the field they are interested in, not to speak of the hard decision on which sub-field of uncertainty quantification to focus. This makes the direct comparison of the latest approaches difficult and also limits the acceptance and adoption of currently existing methods for uncertainty quantification.

- Inability to evaluate uncertainty associated to a single decision

  Existing measures for evaluating the estimated uncertainty (e.g., the expected calibration error) are based on the whole testing data set. This means, that equivalent to classification tasks on unbalanced data sets, the uncertainty associated with single samples or small groups of samples may potentially get biased towards the performance on the rest of the data set. But for practical applications, assessing the reliability of predicted confidence would give much more possibilities than an aggregated reliability based on some testing data, which are independent of the current situation (Kull and Flach 2014). Especially for mission- and safety-critical applications, pointwise evaluation measures could be of paramount importance and hence such evaluation approaches are very desirable.
- Lack of ground truth uncertainties

  Current methods are empirically evaluated and the performance is underlined by reasonable and explainable values of uncertainty. A ground truth uncertainty that could be used for validation is in general not available. Additionally, even though existing methods are calibrated on given data sets, one cannot simply transfer these results to any other data set since one has to be aware of shifts in the data distribution and that many fields can only cover a tiny portion of the actual data environment. In application fields such as EO, the preparation of a huge amount of training data is hard and expensive, and hence synthetic data can be used to train a model. For this artificial data, artificial uncertainties in labels and data should be taken into account to receive a better understanding of the uncertainty quantification performance. The gap between the real and synthetic data, or estimated and real uncertainty further limits the adoption of currently existing methods for uncertainty quantification.
- Explainability issue

  Existing methods of neural network uncertainty quantification deliver predictions of certainty without any clue about what causes possible uncertainties. Even though those certainty values often look *reasonable* to a human observer, one does not know whether the uncertainties are actually predicted based on the same observations the human observer made. But without being sure about the reasons and motivations of single uncertainty estimations, a proper transfer from one data set to another, and even only a domain shift, are much harder to realize with guaranteed performance. Regarding safety-critical real-life applications, the lack of explainability makes the application of the available methods significantly harder. Besides the explainability of neural network decisions, existing methods for uncertainty quantification are not well understood on a higher level. For instance, explaining the behavior of single deterministic approaches, ensembles or Bayesian methods is a current direction of research and remains difficult to grasp in every detail (Fort et al. 2019). It is, however, crucial to understand how those methods operate and capture uncertainty to identify pathways for refinement, and detect and characterize uncertainty, failures, and important shortcomings (Fort et al. 2019).

8.2 Outlook

- Generic evaluation framework

  As already discussed above, there are still problems regarding the evaluation of uncertainty methods, such as the lack of 'ground truth' uncertainties, the inability to test on single instances, and standardized benchmarking protocols. To cope with such issues, the provision of an evaluation protocol containing various concrete baseline data sets and evaluation metrics that cover all types of uncertainty would undoubtedly help to boost research in uncertainty quantification. Also, the evaluation with regard to risk-averse and worst case scenarios should be considered there. This means, that uncertainty predictions with a very high predicted uncertainty should never fail, such as for a prediction of a red or green traffic light. Such a general protocol would enable researchers to easily compare different types of methods against an established benchmark as well as on real world data sets. The adoption of such a standard evaluation protocol should be encouraged by conferences and journals.
- Expert & systematic comparison of baselines

  A broad and structured comparison of existing methods for uncertainty estimation on real world applications is not available yet. An evaluation of real world data is even not standard in current machine learning research papers. As a result, given a specific application, it remains unclear which method for uncertainty estimation performs best and whether the latest methods outperform older methods also on real world examples. This is also partly caused by the fact, that researchers from other domains that use uncertainty quantification methods, in general, present successful applications of single approaches on a specific problem or a data set by hand. Considering this, there are several points that could be adopted for a better comparison within the different research domains. For instance, domain experts should also compare different approaches against each other and present the weaknesses of single approaches in this domain. Similarly, for a better comparison among several domains, a collection of all the works in the different real world domains could be collected and exchanged on a central platform. Such a platform might also help machine learning researchers in providing an additional source of challenges in the real world and would pave the way to broadly highlight weaknesses in the current state-of-the-art approaches. Google's repository on baselines in uncertainties in neural networks (Nado et al. 2021)[5] could be such a platform and a step towards achieving this goal.
- Uncertainty ground truths

  It remains difficult to validate existing methods due to the lack of uncertain ground truths. An actual uncertainty ground truth on which methods can be compared in an ImageNet like manner would make the evaluation of predictions on single samples possible. To reach this, the evaluation of the data generation process and occurring sources of uncertainty, such as the labeling process, might be investigated in more detail.
- Explainability and physical models

  Knowing the actual reasons for a false high certainty or a low certainty makes it much easier to engineer the methods for real life applications, which again increases the trust of people into such methods. Recently, (Antorán et al. 2020) claimed to have published the first work on explainable uncertainty estimation. Uncertainty estimations, in

[5] https://github.com/google/uncertainty-baselines.

general, form an important step towards explainable artificial intelligence. Explainable uncertainty estimations would give an even deeper understanding of the decision process of a neural network, which, in the practical deployment of DNNs, shall incorporate the desired ability to be risk averse while staying applicable in real world (especially safety-critical applications). Also, the possibility of improving explainability with physically based arguments offers great potential. While DNNs are very flexible and efficient, they do not directly embed the domain-specific expert knowledge that is mostly available and can often be described by mathematical or physical models, such as earth system science problems (Reichstein et al. 2019). Such physic-guided models offer a variety of possibilities to include explicit knowledge as well as practical uncertainty representations into a deep learning framework (Willard et al. 2020; De Bézenac et al. 2019).

Funding Open Access funding enabled and organized by Projekt DEAL. This work is in part supported by the German Federal Ministry of Education and Research (BMBF) in the framework of the international future AI lab “AI4EO—Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond” (Grant Number: 01DD20001).

Data availability No data is used in this work. The only part of this paper where concrete data sets play a role is Sect. 6, which lists commonly used data sets for different problem types. All of these data sets are publicly available and free for non-commercial research. A more detailed description of each data set can be found in the referenced works.

References

Abdar M, Pourpanah F, Hussain S, Rezazadegan D, Liu L, Ghavamzadeh M, Fieguth P, Cao X, Khosravi A, Acharya UR et al (2021) A review of uncertainty quantification in deep learning: techniques, applications and challenges. Inf Fusion 76:243–297

Abdullah AA, Hassan MM, Mustafa YT (2022) A review on Bayesian deep learning in healthcare: applications and challenges. IEEE Access 10:36538–36562

Achrack O, Kellerman R, Barzilay O (2020) Multi-loss sub-ensembles for accurate classification with uncertainty estimation. arXiv preprint arXiv:2010.01917

Achterhold J, Koehler JM, Schmeink A, Genewein T (2018) Variational network quantization. In: International conference on learning representations

Ahn S, Balan AK, Welling M (2012) Bayesian posterior sampling via stochastic gradient fisher scoring. In: International conference on machine learning

Ahn S, Shahbaba B, Welling M (2014) Distributed stochastic gradient MCMC. In: International conference on machine learning, PMLR, pp 1044–1052

Amini A, Soleimany A, Karaman S, Rus D (2018) Spatial uncertainty sampling for end-to-end control. arXiv preprint arXiv:1805.04829

Amini A, Schwarting W, Soleimany A, Rus D (2020) Deep evidential regression. In: Advances in neural information processing systems 33

Antorán J, Bhatt U, Adel T, Weller A, Hernández-Lobato JM (2020) Getting a clue: a method for explaining uncertainty estimates. In: International conference on learning representations

Ashukha A, Lyzhov A, Molchanov D, Vetrov D (2019) Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In: International conference on learning representations

Atanov A, Ashukha A, Molchanov D, Neklyudov K, Vetrov D (2019) Uncertainty estimation via stochastic batch normalization. In: International symposium on neural networks, Springer, pp 261–269

Ayhan MS, Berens P (2018) Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks. In: First international conference on medical imaging with deep learning

Ayhan MS, Kühlewein L, Aliyeva G, Inhoffen W, Ziemssen F, Berens P (2020) Expert-validated estimation of diagnostic uncertainty for deep neural networks in diabetic retinopathy detection. Med Image Anal 64(101):724

Ba J, Grosse R, Martens J (2016) Distributed second-order optimization using Kronecker-factored approximations. In: International conference on learning representations

Bae J, Zhang G, Grosse R (2018) Eigenvalue corrected noisy natural gradient. arXiv preprint arXiv:1811.12565

Bailey T, Durrant-Whyte H (2006) Simultaneous localization and mapping (slam): Part ii. IEEE Robot Autom Mag 13(3):108–117

Bajcsy R (1988) Active perception. Proc IEEE 76(8):966–1005

Balan AK, Rathod V, Murphy KP, Welling M (2015) Bayesian dark knowledge. In: Advances in neural information processing systems 28

Barber D, Bishop CM (1998) Ensemble learning in Bayesian neural networks. Nato ASI Ser F Comput Syst Sci 168:215–238

Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, Tacchetti A, Raposo D, Santoro A, Faulkner R, et al. (2018) Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261

Becker S, LeCun Y (1989) Improving the convergence of back-propagation learning with second order methods. In: Proceedings of the 1988 connectionist models summer school, Morgan Kaufmann, pp 29–37

Begoli E, Bhattacharya T, Kusnezov D (2019) The need for uncertainty quantification in machine-assisted medical decision making. Nat Mach Intell 1(1):20–23

Beluch WH, Genewein T, Nürnberger A, Köhler JM (2018) The power of ensembles for active learning in image classification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9368–9377

Berkenkamp F, Schoellig AP, Krause A (2016) Safe controller optimization for quadrotors with gaussian processes. In: 2016 IEEE international conference on robotics and automation (ICRA), IEEE, pp 491–496

Berkenkamp F, Turchetta M, Schoellig A, Krause A (2017) Safe model-based reinforcement learning with stability guarantees. In: Advances in neural information processing systems 30

Bingham E, Chen JP, Jankowiak M, Obermeyer F, Pradhan N, Karaletsos T, Singh R, Szerlip P, Horsfall P, Goodman ND (2019) Pyro: deep universal probabilistic programming. J Mach Learn Res 20(1):973–978

Bishop CM, Nasrabadi NM (2006) Pattern recognition and machine learning, vol 4. Springer, New York

Blundell C, Cornebise J, Kavukcuoglu K, Wierstra D (2015) Weight uncertainty in neural network. In: International conference on machine learning, PMLR, pp 1613–1622

Boerdijk W, Sundermeyer M, Durner M, Triebel R (2021) "What's this?"–Learning to segment unknown objects from manipulation sequences. In: International conference on robotics and automation

Botev A, Ritter H, Barber D (2017) Practical Gauss-Newton optimisation for deep learning. In: International conference on machine learning, PMLR, pp 557–565

Buciluǎ C, Caruana R, Niculescu-Mizil A (2006) Model compression. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, pp 535–541

Buntine WL, Weigend AS (1991) Bayesian back-propagation. Complex Syst 5(6):603–643

Cabañas R, Salmerón A, Masegosa AR (2019) Inferpy: probabilistic modeling with tensorflow made easy. Knowl-Based Syst 168:25–27

Cao Y, Geddes TA, Yang JYH, Yang P (2020) Ensemble deep learning in bioinformatics. Nat Mach Intell 2(9):500–508

Cavalcanti GD, Oliveira LS, Moura TJ, Carvalho GV (2016) Combining diversity measures for ensemble pruning. Pattern Recognit Lett 74:38–45

Chandra R, Jain K, Deo RV, Cripps S (2019) Langevin-gradient parallel tempering for Bayesian neural learning. Neurocomputing 359:315–326

Charpentier B, Zügner D, Günnemann S (2020) Posterior network: uncertainty estimation without OOD samples via density-based pseudo-counts. In: Advances in neural information processing systems 33

Chen C, Ding N, Carin L (2015) On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In: Advances in neural information processing systems 28

Chen SW, Chou CN, Chang E (2018) BDA-PCH: block-diagonal approximation of positive-curvature hessian for training neural networks. CoRR, arxiv:1802.06502

Chen C, Liang J, Ma F, Glass L, Sun J, Xiao C (2021) Unite: uncertainty-based health risk prediction leveraging multi-sourced data. Proc Web Conf 2021:217–226

Chitta K, Alvarez JM, Lesnikowski A (2018) Large-scale visual active learning with deep probabilistic ensembles. arXiv preprint arXiv:1811.03575

Choi J, Chun D, Kim H, Lee HJ (2019) Gaussian YOLOv3: an accurate and fast object detector using localization uncertainty for autonomous driving. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 502–511

Cohn DA, Ghahramani Z, Jordan MI (1996) Active learning with statistical models. J Artif Intell Res 4:129–145

Comaniciu D, Ramesh V, Meer P (2000) Real-time tracking of non-rigid objects using mean shift. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, IEEE, pp 142–149

Corduneanu A, Bishop CM (2001) Variational Bayesian model selection for mixture distributions. Artificial intelligence and statistics. Morgan Kaufmann Waltham, Waltham, pp 27–34

Daftry S, Zeng S, Bagnell JA, Hebert M (2016) Introspective perception: learning to predict failures in vision systems. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, pp 1743–1750

Dai X, Wu X, Wang B, Zhang L (2019) Semisupervised scene classification for remote sensing images: a method based on convolutional neural networks and ensemble learning. IEEE Geosci Remote Sens Lett 16(6):869–873

Davis J, Goadrich M (2006) The relationship between Precision-Recall and ROC curves. In: International conference on machine learning, pp 233–240

Daxberger E, Nalisnick E, Allingham JU, Antoran J, Hernández-Lobato JM (2020) Expressive yet tractable Bayesian deep learning via subnetwork inference. In: Third symposium on advances in approximate Bayesian inference

De Bézenac E, Pajot A, Gallinari P (2019) Deep learning for physical processes: incorporating prior scientific knowledge. J Stat Mech: Theory Exp 12:124009

DeGroot MH, Fienberg SE (1983) The comparison and evaluation of forecasters. J R Stat Soc D 32(1–2):12–22

Dellaert F, Kaess M et al (2017) Factor graphs for robot perception. Found Trends Robot 6(1–2):1–139

Dempster AP (1968) A generalization of Bayesian inference. J R Stat Soc B 30(2):205–232

Deng L (2012) The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Process Mag 29(6):141–142

Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, IEEE, pp 248–255

Denker JS, LeCun Y (1991) Transforming neural-net output levels to probability distributions. In: Advances in neural information processing systems 4

Denker J, Schwartz D, Wittner B, Solla S, Howard R, Jackel L, Hopfield J (1987) Large automatic learning, rule extraction, and generalization. Complex Syst 1(5):877–922

Depeweg S, Hernández-Lobato JM, Udluft S, Runkler T (2017) Sensitivity analysis for predictive uncertainty in Bayesian neural networks. arXiv preprint arXiv:1712.03605

Depeweg S, Hernandez-Lobato JM, Doshi-Velez F, Udluft S (2018) Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In: International conference on machine learning, PMLR, pp 1184–1193

DeVries T, Taylor GW (2017) Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552

Ding N, Fang Y, Babbush R, Chen C, Skeel RD, Neven H (2014) Bayesian sampling using stochastic gradient thermostats. In: Advances in neural information processing systems 27

Dua D, Graff C (2017) UCI machine learning repository. Retrieved June 19, 2021, from http://archive.ics.uci.edu/ml

Duane S, Kennedy AD, Pendleton BJ, Roweth D (1987) Hybrid Monte Carlo. Phys Lett B 195(2):216–222

Dubey KA, Reddi J, S, Williamson SA, Poczos B, Smola AJ, Xing EP, (2016) Variance reduction in stochastic gradient Langevin dynamics. In: Advances in neural information processing systems 29

Durmus A, Moulines E (2019) High-dimensional Bayesian inference via the unadjusted Langevin algorithm. Bernoulli 25(4A):2854–2882

Durmus A, Simsekli U, Moulines E, Badeau R, Richard G (2016) Stochastic gradient Richardson-Romberg Markov chain Monte Carlo. In: Advances in neural information processing systems 29

Durrant-Whyte H, Bailey T (2006) Simultaneous localization and mapping: part I. IEEE Robot Autom Mag 13(2):99–110

Dusenberry MW, Tran D, Choi E, Kemp J, Nixon J, Jerfel G, Heller K, Dai AM (2020) Analyzing the role of model uncertainty for electronic health records. In: Proceedings of the ACM conference on health, inference, and learning, pp 204–213

Eaton-Rosen Z, Bragman F, Bisdas S, Ourselin S, Cardoso MJ (2018) Towards safe deep learning: accurately quantifying biomarker uncertainty in neural network predictions. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 691–699

Ebrahimi S, Elhoseiny M, Darrell T, Rohrbach M (2020) Uncertainty-guided continual learning with Bayesian neural networks. In: International conference on learning representations

Eggenreich S, Payer C, Urschler M, Štern D (2020) Variational inference and Bayesian CNNs for uncertainty estimation in multi-factorial bone age prediction. arXiv preprint arXiv:2002.10819

Englesson E, Azizpour H (2019) Efficient evaluation-time uncertainty estimation by improved distillation. In: International conference on machine learning—workshop on uncertainty and robustness in deep learning

Farquhar S, Gal Y (2019) A unifying Bayesian view of continual learning. arXiv preprint arXiv:1902.06494

Farquhar S, Smith L, Gal Y (2020) Try depth instead of weight correlations: mean-field is a less restrictive assumption for deeper networks. arXiv preprint arXiv:2002.03704

Federici M, Ullrich K, Welling M (2017) Improved Bayesian compression. arXiv preprint arXiv:1711.06494

Feldman Y, Indelman V (2018) Bayesian viewpoint-dependent robust classification under model and localization uncertainty. In: 2018 IEEE international conference on robotics and automation (ICRA), IEEE, pp 3221–3228

Feng D, Rosenbaum L, Dietmayer K (2018) Towards safe autonomous driving: capture uncertainty in the deep neural network for lidar 3d vehicle detection. In: International conference on intelligent transportation systems (ITSC), IEEE, pp 3266–3273

Feng J, Durner M, Márton ZC, Bálint-Benczédi F, Triebel R (2019) Introspective robot perception using smoothed predictions from Bayesian neural networks. In: The international symposium of robotics research, Springer, pp 660–675

Feng J, Lee J, Durner M, Triebel R (2021) Bridging the last mile in sim-to-real robot perception via Bayesian active learning. arXiv preprint arXiv:2109.11547

Filos A, Farquhar S, Gomez AN, Rudner TG, Kenton Z, Smith L, Alizadeh M, De Kroon A, Gal Y (2019) A systematic comparison of Bayesian deep learning robustness in diabetic retinopathy tasks. arXiv preprint arXiv:1912.10481

Foong AY, Li Y, Hernández-Lobato JM, Turner RE (2019) 'In-between'uncertainty in Bayesian neural networks. arXiv preprint arXiv:1906.11537

Fort S, Hu H, Lakshminarayanan B (2019) Deep ensembles: a loss landscape perspective. arXiv preprint arXiv:1912.02757

Fox D (1998) Markov localization-a probabilistic framework for mobile robot localization and navigation. PhD Thesis, Universität Bonn

Fox D, Burgard W, Kruppa H, Thrun S (2000) A probabilistic approach to collaborative multi-robot localization. Auton Robots 8(3):325–344

Fu T, Luo L, Zhang Z (2016) Quasi-newton Hamiltonian Monte Carlo. In: Conference on uncertainty in artificial intelligence

Gal Y (1998) Uncertainty in deep learning. PhD Thesis, University of Cambridge

Gal Y, Ghahramani Z (2015) Bayesian convolutional neural networks with Bernoulli approximate variational inference. arXiv preprint arXiv:1506.02158

Gal Y, Ghahramani Z (2016) Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: International conference on machine learning, PMLR, pp 1050–1059

Gal Y, Hron J, Kendall A (2017a) Concrete dropout. In: Advances in neural information processing systems 30

Gal Y, Islam R, Ghahramani Z (2017b) Deep Bayesian active learning with image data. In: International conference on machine learning, PMLR, pp 1183–1192

Gast J, Roth S (2018) Lightweight probabilistic deep networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3369–3378

Gawlikowski J, Saha S, Kruspe A, Zhu XX (2022) An advanced Dirichlet prior network for out-of-distribution detection in remote sensing. IEEE Trans Geosci Remote Sens 60:1–19

George T, Laurent C, Bouthillier X, Ballas N, Vincent P (2018) Fast approximate natural gradient descent in a Kronecker factored eigenbasis. In: Advances in neural information processing systems 31

Ghandeharioun A, Eoff B, Jou B, Picard R (2019) Characterizing sources of uncertainty to proxy calibration and disambiguate annotator and data bias. In: 2019 IEEE/CVF international conference on computer vision workshop (ICCVW), IEEE, pp 4202–4206

Ghanem R, Higdon D, Owhadi H et al (2017) Handbook of uncertainty quantification, vol 6. Springer, Cham

Ghavamzadeh M, Mannor S, Pineau J, Tamar A (2015) Bayesian reinforcement learning: a survey. Found Trends Mach Learn 8(5–6):359–483

Ghesu FC, Georgescu B, Gibson E, Guendel S, Kalra MK, Singh R, Digumarthy SR, Grbic S, Comaniciu D (2019) Quantifying and leveraging classification uncertainty for chest radiograph assessment. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 676–684

Ghosh S, Yao J, Doshi-Velez F (2019) Model selection in Bayesian neural networks via horseshoe priors. J Mach Learn Res 20(182):1–46

Ghoshal B, Ghoshal B, Swift S, Tucker A (2021) Uncertainty estimation in sars-cov-2 b-cell epitope prediction for vaccine development. In: Artificial intelligence in medicine: 19th international conference on artificial intelligence in medicine, AIME 2021, Virtual Event, June 15–18, 2021, proceedings, Springer, pp 361–366

Gianfrancesco MA, Tamang S, Yazdany J, Schmajuk G (2018) Potential biases in machine learning algorithms using electronic health record data. JAMA Intern Med 178(11):1544–1547

Giardino C, Bresciani M, Villa P, Martinelli A (2010) Application of remote sensing in water resource management: the case study of lake Trasimeno, Italy. Water Resour Manage 24(14):3885–3899

Goodfellow I, Bengio Y, Courville A, Bengio Y (2016) Deep learning, vol 1. MIT Press, Cambridge

Graves A (2011) Practical variational inference for neural networks. In: Advances in neural information processing systems 24

Grimmett H, Triebel R, Paul R, Posner I (2016) Introspective classification for robot perception. Int J Robot Res 35(7):743–762

Grosse R, Martens J (2016) A kronecker-factored approximate fisher matrix for convolution layers. In: International conference on machine learning, PMLR, pp 573–582

Guo J, Gould S (2015) Deep CNN ensemble with data augmentation for object detection. arXiv preprint arXiv:1506.07224

Guo C, Pleiss G, Sun Y, Weinberger KQ (2017) On calibration of modern neural networks. In: International conference on machine learning, PMLR, pp 1321–1330

Guo H, Liu H, Li R, Wu C, Guo Y, Xu M (2018) Margin & diversity based ordering ensemble pruning. Neurocomputing 275:237–246

Gurău C, Tong CH, Posner I (2016) Fit for purpose? Predicting perception performance based on past experience. In: International symposium on experimental robotics, Springer, pp 454–464

Gustafsson FK, Danelljan M, Schon TB (2020) Evaluating scalable Bayesian deep learning methods for robust computer vision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp 318–319

Han T, Li YF (2022) Out-of-distribution detection-assisted trustworthy machinery fault diagnosis approach with uncertainty-aware deep ensembles. Reliab Eng Syst Saf 226(108):648

Hansen LK, Salamon P (1990) Neural network ensembles. IEEE Trans Pattern Anal Mach Intell 12(10):993–1001

Hein M, Andriushchenko M, Bitterwolf J (2019) Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 41–50

Hendrycks D, Gimpel K (2017) A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: International conference on learning representations

Hendrycks D, Mazeika M, Dietterich T (2019) Deep anomaly detection with outlier exposure. In: International conference on learning representations

Hennig P (2013) Fast probabilistic optimization from noisy gradients. In: International conference on machine learning, PMLR, pp 62–70

Heo J, Lee HB, Kim S, Lee J, Kim KJ, Yang E, Hwang SJ (2018) Uncertainty-aware attention for reliable interpretation and prediction. In: Advances in neural information processing systems 31

Hernández S, López JL (2020) Uncertainty quantification for plant disease detection using Bayesian deep learning. Appl Soft Comput 96(106):597

Hernández-Lobato JM, Adams R (2015) Probabilistic backpropagation for scalable learning of Bayesian neural networks. In: International conference on machine learning, PMLR, pp 1861–1869

Hernández-Lobato JM, Li Y, Rowland M, Bui T, Hernández-Lobato D, Turner R (2016) Black-box alpha divergence minimization. In: International conference on machine learning, PMLR, pp 1511–1520

Herrmann F (2020) A deep-learning based Bayesian approach to seismic imaging and uncertainty quantification. In: EAGE 2020 annual conference & exhibition online, EAGE Publications BV, pp 1–5

Herron EJ, Young SR, Potok TE (2020) Ensembles of networks produced from neural architecture search. In: International conference on high performance computing, Springer, pp 223–234

Hinton GE, Van Camp D (1993) Keeping the neural networks simple by minimizing the description length of the weights. In: Proceedings of the sixth annual conference on computational learning theory, pp 5–13

Hinton GE, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv:abs/1503.02531

Hobbhahn M, Kristiadi A, Hennig P (2022) Fast predictive uncertainty for classification with Bayesian deep networks. In: Conference on uncertainty in artificial intelligence, PMLR, pp 822–832

Hsu YC, Shen Y, Jin H, Kira Z (2020) Generalized odin: detecting out-of-distribution image without learning from out-of-distribution data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10951–10960

Hu S, Worrall D, Knegt S, Veeling B, Huisman H, Welling M (2019) Supervised uncertainty quantification for segmentation with multiple annotations. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 137–145

Huang G, Li Y, Pleiss G, Liu Z, Hopcroft JE, Weinberger KQ (2017) Snapshot ensembles: train 1, get m for free. In: International conference on learning representations

Huang W, Zhang J, Huang K (2019a) Bootstrap estimated uncertainty of the environment model for model-based reinforcement learning. In: Proceedings of 28th the AAAI conference on artificial intelligence, pp 3870–3877

Huang X, Yang J, Li L, Deng H, Ni B, Xu Y (2019b) Evaluating and boosting uncertainty quantification in classification. arXiv preprint arXiv:1909.06030

Hüllermeier E, Waegeman W (2021) Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. Mach Learn 110(3):457–506

Humt M, Lee J, Triebel R (2020) Bayesian optimization meets laplace approximation for robotic introspection. arXiv preprint arXiv:2010.16141

Immer A, Korzepa M, Bauer M (2021) Improving predictions of Bayesian neural nets via local linearization. In: Proceedings of The 24th international conference on artificial intelligence and statistics, PMLR, pp 703–711

Ito Y, Srinivasan C, Izumi H (2005) Bayesian learning of neural networks adapted to changes of prior probabilities. In: International conference on artificial neural networks, Springer, pp 253–259

Iuzzolino ML, Umada T, Ahmed NR, Szafir DA (2020) In automation we trust: investigating the role of uncertainty in active learning systems. arXiv preprint arXiv:2004.00762

Izmailov P, Maddox WJ, Kirichenko P, Garipov T, Vetrov D, Wilson AG (2020) Subspace inference for Bayesian deep learning. In: Conference on uncertainty in artificial intelligence, PMLR, pp 1169–1179

Kaess M, Ila V, Roberts R, Dellaert F (2010) The Bayes tree: an algorithmic foundation for probabilistic robot mapping. In: Algorithmic foundations of robotics IX. Springer, Berlin, pp 157–173

Kahn G, Villaflor A, Pong V, Abbeel P, Levine S (2017) Uncertainty-aware reinforcement learning for collision avoidance. arXiv preprint arXiv:1702.01182

Kahn G, Villaflor A, Ding B, Abbeel P, Levine S (2018) Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. In: 2018 IEEE international conference on robotics and automation (ICRA), IEEE, pp 5129–5136

Kawashima T, Yu Q, Asai A, Ikami D, Aizawa K (2021) The aleatoric uncertainty estimation using a separate formulation with virtual residuals. In: 2020 25th international conference on pattern recognition (ICPR), IEEE, pp 1438–1445

Kendall A, Gal Y (2017) What uncertainties do we need in Bayesian deep learning for computer vision? In: Advances in neural information processing systems 30

Kendall AG (2019) Geometry and uncertainty in deep learning for computer vision. PhD Thesis, University of Cambridge, UK

Khan ME, Liu Z, Tangkaratt V, Gal Y (2017) Vprop: variational inference using rmsprop. arXiv preprint arXiv:1712.01038

Khan M, Nielsen D, Tangkaratt V, Lin W, Gal Y, Srivastava A (2018) Fast and scalable Bayesian deep learning by weight-perturbation in Adam. In: International conference on machine learning, PMLR, pp 2611–2620

Khan MEE, Immer A, Abedi E, Korzepa M (2019) Approximate inference turns deep networks into Gaussian processes. In: Advances in neural information processing systems 32

Kim W, Goyal B, Chawla K, Lee J, Kwon K (2018) Attention-based ensemble for deep metric learning. In: Proceedings of the European conference on computer vision (ECCV), pp 736–751

Kim I, Kim Y, Kim S (2020) Learning loss for test-time augmentation. In: Advances in neural information processing systems 33

Kim Q, Ko JH, Kim S, Park N, Jhe W (2021) Bayesian neural network with pretrained protein embedding enhances prediction accuracy of drug-protein interaction. Bioinformatics 37(20):3428–3435

Kingma DP, Salimans T, Welling M (2015) Variational dropout and the local reparameterization trick. In: Advances in neural information processing systems 28

Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A et al (2017) Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci 114(13):3521–3526

Kirsch A, Van Amersfoort J, Gal Y (2019) Batchbald: efficient and diverse batch acquisition for deep Bayesian active learning. In: Advances in neural information processing systems 32

Kocić J, Jovičić N, Drndarević V (2019) An end-to-end deep neural network for autonomous driving designed for embedded automotive platforms. Sensors 19(9):2064

Koh PW, Sagawa S, Marklund H, Xie SM, Zhang M, Balsubramani A, Hu W, Yasunaga M, Phillips RL, Gao I, et al. (2021) Wilds: a benchmark of in-the-wild distribution shifts. In: International conference on machine learning, PMLR, pp 5637–5664

Kristiadi A, Hein M, Hennig P (2020) Being Bayesian, even just a bit, fixes overconfidence in relu networks. In: International conference on machine learning, PMLR, pp 5436–5446

Kristiadi A, Hein M, Hennig P (2021) Learnable uncertainty under laplace approximations. In: Conference on uncertainty in artificial intelligence, PMLR, pp 344–353

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

Krueger D, Huang CW, Islam R, Turner R, Lacoste A, Courville A (2017) Bayesian hypernetworks. arXiv preprint arXiv:1710.04759

Kuleshov V, Fenner N, Ermon S (2018) Accurate uncertainties for deep learning using calibrated regression. In: International conference on machine learning, PMLR, pp 2796–2804

Kull M, Flach PA (2014) Reliability maps: a tool to enhance probability estimates and improve classification accuracy. In: Joint European conference on machine learning and knowledge discovery in databases, Springer, pp 18–33

Kull M, Perello Nieto M, Kängsepp M, Silva Filho T, Song H, Flach P (2019) Beyond temperature scaling: obtaining well-calibrated multi-class probabilities with Dirichlet calibration. In: Advances in neural information processing systems 32

Kurutach T, Clavera I, Duan Y, Tamar A, Abbeel P (2018) Model-ensemble trust-region policy optimization. In: International conference on learning representations

Kushner H, Yin GG (2003) Stochastic approximation and recursive algorithms and applications, vol 35. Springer, New York

LaBonte T, Martinez C, Roberts SA (2019) We know where we don't know: 3d Bayesian CNNs for credible geometric uncertainty. arXiv preprint arXiv:1910.10793

Lakshminarayanan B, Pritzel A, Blundell C (2017) Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in neural information processing systems 30

Laves MH, Ihler S, Kortmann KP, Ortmaier T (2019) Well-calibrated model uncertainty with temperature scaling for dropout variational inference. arXiv preprint arXiv:1909.13550

Le Roux N, Fitzgibbon AW (2010) A fast natural newton method. In: International conference on machine learning

LeCun Y, Denker J, Solla S (1989) Optimal brain damage. In: Advances in neural information processing systems 2

LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324

Lee J, AlRegib G (2020) Gradients as a measure of uncertainty in neural networks. In: 2020 IEEE international conference on image processing (ICIP), IEEE, pp 2416–2420

Lee S, Purushwalkam S, Cogswell M, Crandall D, Batra D (2015) Why m heads are better than one: training a diverse ensemble of deep networks. arXiv preprint arXiv:1511.06314

Lee K, Lee H, Lee K, Shin J (2018a) Training confidence-calibrated classifiers for detecting out-of-distribution samples. In: International conference on learning representations

Lee K, Lee K, Lee H, Shin J (2018b) A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In: Advances in neural information processing systems 31

Lee J, Humt M, Feng J, Triebel R (2020) Estimating model uncertainty of neural networks in sparse information form. In: International conference on machine learning, PMLR, pp 5702–5713

Lee J, Feng J, Humt M, Müller MG, Triebel R (2022) Trust your robots! Predictive uncertainty estimation of neural networks with sparse gaussian processes. In: Conference on robot learning, PMLR, pp 1168–1179

Leimkuhler B, Reich S (2004) Simulating Hamiltonian dynamics, vol 14. Cambridge University Press, Cambridge

Leimkuhler B, Shang X (2016) Adaptive thermostats for noisy gradient systems. SIAM J Sci Comput 38(2):A712–A736

Leutbecher M, Palmer TN (2008) Ensemble forecasting. J Comput Phys 227(7):3515–3539

Levi D, Gispan L, Giladi N, Fetaya E (2022) Evaluating and calibrating uncertainty prediction in regression tasks. Sensors 22(15):5540

Li Y, Gal Y (2017) Dropout inference in Bayesian neural networks with α-divergences. In: International conference on machine learning, PMLR, pp 2052–2061

Li Z, Hoiem D (2020) Improving confidence estimates for unfamiliar examples. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2686–2695

Li C, Chen C, Carlson D, Carin L (2016a) Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In: Proceedings of the 30th AAAI conference on artificial intelligence

Li C, Stevens A, Chen C, Pu Y, Gan Z, Carin L (2016b) Learning weight uncertainty with stochastic gradient MCMC for shape classification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5666–5675

Li H, Barnaghi P, Enshaeifar S, Ganz F (2020) Continual learning using Bayesian neural networks. IEEE Trans Neural Netw Learn Syst 32(9):4243–4252

Liang F, Li Q, Zhou L (2018a) Bayesian neural networks for selection of drug sensitive genes. J Am Stat Assoc 113(523):955–972

Liang S, Li Y, Srikant R (2018b) Enhancing the reliability of out-of-distribution image detection in neural networks. In: International conference on learning representations

Lindqvist J, Olmin A, Lindsten F, Svensson L (2020) A general framework for ensemble distribution distillation. In: 2020 IEEE 30th international workshop on machine learning for signal processing (MLSP), IEEE, pp 1–6

Liu DC, Nocedal J (1989) On the limited memory BFGS method for large scale optimization. Math Program 45(1):503–528

Livieris IE, Iliadis L, Pintelas P (2021) On ensemble techniques of weight-constrained neural networks. Evol Syst 12(1):155–167

Loeliger HA (2004) An introduction to factor graphs. IEEE Signal Process Mag 21(1):28–41

Loftus TJ, Shickel B, Ruppert MM, Balch JA, Ozrazgat-Baslanti T, Tighe PJ, Efron PA, Hogan WR, Rashidi P, Upchurch GR Jr et al (2022) Uncertainty-aware deep learning in healthcare: a scoping review. PLoS Digit Health 1(8):e0000,085

Loquercio A, Segu M, Scaramuzza D (2020) A general framework for uncertainty estimation in deep learning. IEEE Robot Autom Lett 5(2):3153–3160

Louizos C, Welling M (2016) Structured and efficient variational deep learning with matrix gaussian posteriors. In: International conference on machine learning, PMLR, pp 1708–1716

Louizos C, Welling M (2017) Multiplicative normalizing flows for variational Bayesian neural networks. In: International conference on machine learning, PMLR, pp 2218–2227

Louizos C, Ullrich K, Welling M (2017) Bayesian compression for deep learning. In: Advances in neural information processing systems 30

Lukasik M, Bhojanapalli S, Menon A, Kumar S (2020) Does label smoothing mitigate label noise? In: International conference on machine learning, PMLR, pp 6448–6458

Lütjens B, Everett M, How JP (2019) Safe reinforcement learning with model uncertainty estimates. In: 2019 international conference on robotics and automation (ICRA), IEEE, pp 8662–8668

Lv F, Han M, Qiu T (2017) Remote sensing image classification based on ensemble extreme learning machine with stacked autoencoder. IEEE Access 5:9021–9031

Lyzhov A, Molchanova Y, Ashukha A, Molchanov D, Vetrov D (2020) Greedy policy search: a simple baseline for learnable test-time augmentation. In: Conference on uncertainty in artificial intelligence, PMLR, pp 1308–1317

Ma YA, Chen T, Fox E (2015) A complete recipe for stochastic gradient MCMC. In: Advances in neural information processing systems 28

MacKay D (1992a) Bayesian model comparison and backprop nets. In: Advances in neural information processing systems 4

MacKay DJ (1992b) Information-based objective functions for active data selection. Neural Comput 4(4):590–604

MacKay DJ (1992c) A practical Bayesian framework for backpropagation networks. Neural Comput 4(3):448–472

Maddox WJ, Izmailov P, Garipov T, Vetrov DP, Wilson AG (2019) A simple baseline for Bayesian uncertainty in deep learning. In: Advances in neural information processing systems 32

Malinin A (2019) Uncertainty estimation in deep learning with application to spoken language assessment. PhD Thesis, University of Cambridge

Malinin A, Gales M (2018) Predictive uncertainty estimation via prior networks. In: Advances in neural information processing systems 31

Malinin A, Gales M (2019) Reverse kl-divergence training of prior networks: improved uncertainty and adversarial robustness. In: Advances in neural information processing systems 32

Malinin A, Mlodozeniec B, Gales M (2020) Ensemble distribution distillation. In: International conference on learning representations

Marceau-Caron G, Ollivier Y (2017) Natural Langevin dynamics for neural networks. In: International conference on geometric science of information, Springer, pp 451–459

Maroñas J, Ramos-Castro D, Palacios RP (2020) Improving calibration in mixup-trained deep neural networks through confidence-based loss functions. arXiv:abs/2003.09946

Martens J, Grosse R (2015) Optimizing neural networks with Kronecker-factored approximate curvature. In: International conference on machine learning, PMLR, pp 2408–2417

Martinez WG (2021) Ensemble pruning via quadratic margin maximization. IEEE Access 9:48931-48951

Martínez-Muñoz G, Hernández-Lobato D, Suárez A (2008) An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Trans Pattern Anal Mach Intell 31(2):245–259

Marushko E, Doudkin A (2020) Methods of using ensembles of heterogeneous models to identify remote sensing objects. Pattern Recognit Image Anal 30(2):211–216

McClure P, Kriegeskorte N (2016) Robustly representing uncertainty through sampling in deep neural networks. arXiv preprint arXiv:1611.01639

McClure P, Rho N, Lee JA, Kaczmarzyk JR, Zheng CY, Ghosh SS, Nielson DM, Thomas AG, Bandettini P, Pereira F (2019) Knowing what you know in brain segmentation using Bayesian deep neural networks. Front Neuroinform 13:67

Mehrtash A, Wells WM, Tempany CM, Abolmaesumi P, Kapur T (2020) Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. IEEE Trans Med Imaging 39(12):3868–3878

Minka TP (2001) Expectation propagation for approximate Bayesian inference. In: Conference on uncertainty in artificial intelligence, pp 362–369

Minka T et al (2005) Divergence measures and message passing. Tech. rep, Microsoft Research

Mishkin A, Kunstner F, Nielsen D, Schmidt M, Khan ME (2018) Slang: fast structured covariance approximations for Bayesian deep learning with natural gradient. In: Advances in neural information processing systems 31

Mitros J, Mac Namee B (2019) On the validity of Bayesian neural networks for uncertainty estimation. arXiv preprint arXiv:1912.01530

Mobiny A, Yuan P, Moulik SK, Garg N, Wu CC, Van Nguyen H (2021) Dropconnect is effective in modeling uncertainty of Bayesian deep networks. Sci Rep 11(1):1–14

Monteiro M, Le Folgoc L, Coelho de Castro D, Pawlowski N, Marques B, Kamnitsas K, van der Wilk M, Glocker B (2020) Stochastic segmentation networks: modelling spatially correlated aleatoric uncertainty. Adv Neural Inf Process Syst 33:12756–12767

Montemerlo M, Thrun S, Koller D, Wegbreit B, et al. (2002) Fastslam: a factored solution to the simultaneous localization and mapping problem. In: AAAI conference on innovative applications of artificial intelligence

Moshkov N, Mathe B, Kertesz-Farkas A, Hollandi R, Horvath P (2020) Test-time augmentation for deep learning-based cell segmentation on microscopy images. Sci Rep 10(1):5068

Możejko M, Susik M, Karczewski R (2018) Inhibited softmax for uncertainty estimation in neural networks. arXiv preprint arXiv:1810.01861

Mukhoti J, Gal Y (2018) Evaluating Bayesian deep learning methods for semantic segmentation. arXiv preprint arXiv:1811.12709

Mukhoti J, Stenetorp P, Gal Y (2018) On the importance of strong baselines in Bayesian deep learning. arXiv preprint arXiv:1811.09385

Müller R, Kornblith S, Hinton GE (2019) When does label smoothing help? In: Advances in neural information processing systems 32

Mundt M, Pliushch I, Majumder S, Ramesh V (2019) Open set recognition through deep neural network uncertainty: does out-of-distribution detection require generative classifiers? In: Proceedings of the IEEE/CVF international conference on computer vision workshops

Nado Z, Snoek J, Grosse RB, Duvenaud D, Xu B, Martens J (2018) Stochastic gradient Langevin dynamics that exploit neural network structure. In: International conference on learning representations (workshop)

Nado Z, Band N, Collier M, Djolonga J, Dusenberry MW, Farquhar S, Feng Q, Filos A, Havasi M, Jenatton R, et al. (2021) Uncertainty baselines: benchmarks for uncertainty & robustness in deep learning. arXiv preprint arXiv:2106.04015

Naeini MP, Cooper G, Hauskrecht M (2015) Obtaining well calibrated probabilities using Bayesian binning. In: Proceedings of the 26th AAAI conference on artificial intelligence

Nair T, Precup D, Arnold DL, Arbel T (2020) Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. Med Image Anal 59(101):557

Nalepa J, Myller M, Kawulok M (2019) Training-and test-time data augmentation for hyperspectral image segmentation. IEEE Geosci Remote Sens Lett 17(2):292–296

Nandy J, Hsu W, Lee ML (2020) Towards maximizing the representation gap between in-domain & out-of-distribution examples. In: Advances in neural information processing systems 33

Nanni L, Brahnam S, Maguolo G (2019) Data augmentation for building an ensemble of convolutional neural networks. In: Innovation in medicine and healthcare systems, and multimedia. Springer, pp 61–69

Nanni L, Ghidoni S, Brahnam S (2020) Ensemble of convolutional neural networks for bioimage classification. Appl Comput Inform 17:19–35

Narr A, Triebel R, Cremers D (2016) Stream-based active learning for efficient and adaptive classification of 3d objects. In: 2016 IEEE international conference on robotics and automation (ICRA), IEEE, pp 227–233

Neal RM (1992) Bayesian training of backpropagation networks by the hybrid Monte Carlo method. University of Toronto, Tech. rep

Neal RM (1994) An improved acceptance procedure for the hybrid Monte Carlo algorithm. J Comput Phys 111(1):194–203

Neal RM (1995) Bayesian learning for neural networks. PhD Thesis, University of Toronto

Neal RM et al (2011) MCMC using Hamiltonian dynamics. Handb Markov chain Monte Carlo 2(11):2

Nemeth C, Fearnhead P (2021) Stochastic gradient Markov chain Monte Carlo. J Am Stat Assoc 116(533):433–450

Netzband M, Stefanov WL, Redman C (2007) Applied remote sensing for urban planning, governance and sustainability. Springer, Berlin

Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY (2011) Reading digits in natural images with unsupervised feature learning. In: Advances in neural information processing systems (workshops)

Nguyen A, Yosinski J, Clune J (2015) Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 427–436

Nguyen CV, Li Y, Bui TD, Turner RE (2018) Variational continual learning. In: International conference on learning representations

Nguyen VL, Destercke S, Hüllermeier E (2019) Epistemic uncertainty sampling. In: International conference on discovery science, Springer, pp 72–86

Niraula P, Mateu J, Chaudhuri S (2022) A Bayesian machine learning approach for spatio-temporal prediction of covid-19 cases. Stoch Environ Res Risk Assess 36(8):2265–2283

Nixon J, Dusenberry MW, Zhang L, Jerfel G, Tran D (2019) Measuring calibration in deep learning. In: Conference on computer vision and pattern recognition (workshops)

Oala L, Heiß C, Macdonald J, März M, Samek W, Kutyniok G (2020) Interval neural networks: uncertainty scores. arXiv preprint arXiv:2003.11566

Oberdiek P, Rottmann M, Gottschalk H (2018) Classification uncertainty of deep neural networks based on gradient information. In: IAPR workshop on artificial neural networks in pattern recognition, Springer, pp 113–125

Osawa K, Swaroop S, Khan MEE, Jain A, Eschenhagen R, Turner RE, Yokota R (2019) Practical deep learning with Bayesian principles. In: Advances in neural information processing systems 32

Ovadia Y, Fertig E, Ren J, Nado Z, Sculley D, Nowozin S, Dillon J, Lakshminarayanan B, Snoek J (2019) Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In: Advances in neural information processing systems 32

Parker WS (2013) Ensemble modeling, uncertainty and robust predictions. Wiley Interdiscip Rev: Climate Change 4(3):213–223

Patel K, Beluch W, Zhang D, Pfeiffer M, Yang B (2021) On-manifold adversarial data augmentation improves uncertainty calibration. In: 2020 25th international conference on pattern recognition (ICPR), IEEE, pp 8029–8036

Patterson S, Teh YW (2013) Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In: Advances in neural information processing systems 26

Pearce T, Brintrup A, Zaki M, Neely A (2018) High-quality prediction intervals for deep learning: a distribution-free, ensembled approach. In: International conference on machine learning, PMLR, pp 4075–4084

Peretroukhin V, Giamou M, Rosen DM, Greene WN, Roy N, Kelly J (2020) A smooth representation of belief over so (3) for deep rotation learning with uncertainty. arXiv preprint arXiv:2006.01031

Pereyra G, Tucker G, Chorowski J, Kaiser Ł, Hinton G (2017) Regularizing neural networks by penalizing confident output distributions. arXiv preprint arXiv:1701.06548

Peterson JC, Battleday RM, Griffiths TL, Russakovsky O (2019) Human uncertainty makes classification more robust. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9617–9626

Pop R, Fulop P (2018) Deep ensemble Bayesian active learning: addressing the mode collapse issue in Monte Carlo dropout via ensembles. arXiv preprint arXiv:1811.03897

Postels J, Ferroni F, Coskun H, Navab N, Tombari F (2019) Sampling-free epistemic uncertainty estimation using approximated variance propagation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 2931–2940

Pulgar FJ, Rivera AJ, Charte F, Jesus MJd (2017) On the impact of imbalanced data in convolutional neural networks performance. In: International conference on hybrid artificial intelligence systems, Springer, pp 220–232

Qiu R, Jia Y, Hadzikadic M, Dulin M, Niu X, Wang X (2019) Modeling the uncertainty in electronic health records: a Bayesian deep learning approach. arXiv preprint arXiv:1907.06162

Raghu M, Blumer K, Sayres R, Obermeyer Z, Kleinberg B, Mullainathan S, Kleinberg J (2019) Direct uncertainty prediction for medical second opinions. In: International conference on machine learning, PMLR, pp 5281–5290

Rahaman R et al (2021) Uncertainty quantification and deep ensembles. In: Advances in neural information processing systems 34

Rajeswaran A, Ghotra S, Ravindran B, Levine S (2017) EPOpt: learning robust neural network policies using model ensembles. In: International conference on learning representations

Ramalho T, Miranda M (2020) Density estimation in representation space to predict model uncertainty. In: International workshop on engineering dependable and secure machine learning systems, Springer, pp 84–96

Rawat M, Wistuba M, Nicolae MI (2017) Harnessing model uncertainty for detecting adversarial examples. In: Advances in neural information processing systems—workshop on Bayesian deep learning

Reich S, Mueller D, Andrews N (2020) Ensemble distillation for structured prediction: calibrated, accurate, fast-choose three. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 5583–5595

Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N et al (2019) Deep learning and process understanding for data-driven earth system science. Nature 566(7743):195–204

Reinhold JC, He Y, Han S, Chen Y, Gao D, Lee J, Prince JL, Carass A (2020) Validating uncertainty in medical image translation. In: 2020 IEEE 17th international symposium on biomedical imaging (ISBI), IEEE, pp 95–98

Ren J, Liu PJ, Fertig E, Snoek J, Poplin R, Depristo M, Dillon J, Lakshminarayanan B (2019) Likelihood ratios for out-of-distribution detection. In: Advances in neural information processing systems 32

Renda A, Barsacchi M, Bechini A, Marcelloni F (2019) Comparing ensemble strategies for deep learning: an application to facial expression recognition. Expert Syst Appl 136:1–11

Rewicki F (2021) Estimating uncertainty of deep learning multi-label classifications using Laplace approximation. PhD Thesis, Friedrich-Schiller-Universität Jena

Rezende D, Mohamed S (2015) Variational inference with normalizing flows. In: International conference on machine learning, PMLR, pp 1530–1538

Richards SM, Berkenkamp F, Krause A (2018) The Lyapunov neural network: adaptive stability certification for safe learning of dynamical systems. In: Conference on robot learning, PMLR, pp 466–476

Richter C, Roy N (2017) Safe visual navigation via deep learning and novelty detection. Robotics: Science and Systems Foundation

Ritter H, Botev A, Barber D (2018) A scalable laplace approximation for neural networks. In: International conference on learning representations

Roberts GO, Stramer O (2002) Langevin diffusions and metropolis-hastings algorithms. Methodol Comput Appl Probab 4(4):337–357

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 234–241

Ross S, Pineau J, Paquet S, Chaib-Draa B (2008) Online planning algorithms for POMDPs. J Artif Intell Res 32:663–704

Rossky PJ, Doll JD, Friedman HL (1978) Brownian dynamics as smart Monte Carlo simulation. J Chem Phys 69(10):4628–4633

Roy AG, Conjeti S, Navab N, Wachinger C, Initiative ADN et al (2019) Bayesian quicknat: model uncertainty in deep whole-brain segmentation for structure-wise quality control. NeuroImage 195:11–22

Rußwurm M, Ali M, Zhu XX, Gal Y, Körner M (2020) Model and data uncertainty for satellite time series forecasting with deep recurrent models. In: IEEE international geoscience and remote sensing symposium, IEEE, pp 7025–7028

Ruzicka V, D'Aronco S, Wegner JD, Schindler K (2020) Deep active learning in remote sensing for data efficient change detection. In: Proceedings of MACLEAN: MAChine Learning for EArth ObservatioN workshop co-located with the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML/PKDD 2020), RWTH Aachen University

Sagi O, Rokach L (2018) Ensemble learning: a survey. Wiley Interdiscip Rev: Data Min Knowl Discov 8(4):e1249

Salimans T, Kingma DP (2016) Weight normalization: a simple reparameterization to accelerate training of deep neural networks. In: Advances in neural information processing systems 29

Sato MA (2001) Online model selection based on the variational Bayes. Neural Comput 13(7):1649–1681

Sato I, Nakagawa H (2014) Approximation analysis of stochastic gradient Langevin dynamics by using Fokker-Planck equation and ito process. In: International conference on machine learning, PMLR, pp 982–990

Scalia G, Grambow CA, Pernici B, Li YP, Green WH (2020) Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. J Chem Inf Model 60(6):2697–2717

Schmitt M, Zhu XX (2016) Data fusion and remote sensing: an ever-growing relationship. IEEE Geosci Remote Sens Mag 4(4):6–23

Seebock P, Orlando JI, Schlegl T, Waldstein SM, Bogunovic H, Klimscha S, Langs G, Schmidt-Erfurth U (2020) Exploiting epistemic uncertainty of anatomy segmentation for anomaly detection in retinal OCT. IEEE Trans Med Imaging 39:87–98

Semenova E, Williams DP, Afzal AM, Lazic SE (2020) A Bayesian neural network for toxicity prediction. Comput Toxicol 16(100):133

Sensoy M, Kaplan L, Kandemir M (2018) Evidential deep learning to quantify classification uncertainty. In: Advances in neural information processing systems 31

Seo S, Seo PH, Han B (2019) Learning for single-shot confidence calibration in deep neural networks through stochastic inferences. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9030–9038

Serban AC, Poll E, Visser J (2018) Adversarial examples-a complete characterisation of the phenomenon. arXiv preprint arXiv:1810.01185

Settles B (2009) Active learning literature survey. University of Wisconsin-Madison Department of Computer Sciences, Tech. rep

Shafaei A, Schmidt M, Little JJ (2019) A less biased evaluation of out-of-distribution sample detectors. In: British machine learning conference

Shang X, Zhu Z, Leimkuhler B, Storkey AJ (2015) Covariance-controlled adaptive langevin thermostat for large-scale Bayesian sampling. In: Advances in neural information processing systems 28

Shanmugam D, Blalock D, Balakrishnan G, Guttag J (2020) When and why test-time augmentation works. arXiv preprint arXiv:2011.11156

Shinde K, Lee J, Humt M, Sezgin A, Triebel R (2020) Learning multiplicative interactions with Bayesian neural networks for visual-inertial odometry. In: Workshop on AI for autonomous driving at the 37th international conference on machine learning

Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6(1):1–48

Silver D, Veness J (2010) Monte-Carlo planning in large POMDPs. In: Advances in neural information processing systems 23

Simsekli U, Badeau R, Cemgil T, Richard G (2016) Stochastic Quasi-Newton Langevin Monte Carlo. In: International conference on machine learning, PMLR, pp 642–651

Smith L, Gal Y (2018) Understanding measures of uncertainty for adversarial example detection. In: Conference on uncertainty in artificial intelligence, pp 560–569

Soberanis-Mukul RD, Navab N, Albarqouni S (2020) Uncertainty-based graph convolutional networks for organ segmentation refinement. In: Medical imaging with deep learning, PMLR, pp 755–769

Soleimany AP, Suresh H, Ortiz JJG, Shanmugam D, Gural N, Guttag J, Bhatia SN (2019) Image segmentation of liver stage malaria infection with spatial uncertainty sampling. arXiv preprint arXiv:1912.00262

Soleimany AP, Amini A, Goldman S, Rus D, Bhatia SN, Coley CW (2021) Evidential deep learning for guided molecular property prediction and discovery. ACS Central Sci 7(8):1356–1367

Ståhl N, Falkman G, Karlsson A, Mathiason G (2020) Evaluation of uncertainty quantification in deep learning. In: International conference on information processing and management of uncertainty in knowledge-based systems, Springer, pp 556–568

Stulp F, Theodorou E, Buchli J, Schaal S (2011) Learning to grasp under uncertainty. In: 2011 IEEE international conference on robotics and automation, IEEE, pp 5703–5708

Su D, Ting YY, Ansel J (2018) Tight prediction intervals using expanded interval minimization. arXiv preprint arXiv:1806.11222

Sun S, Chen C, Carin L (2017) Learning structured weight uncertainty in Bayesian neural networks. In: Artificial intelligence and statistics, PMLR, pp 1283–1292

Sun S, Zhang G, Shi J, Grosse R (2018) Functional variational Bayesian neural networks. In: International conference on learning representations

Sünderhauf N, Brock O, Scheirer W, Hadsell R, Fox D, Leitner J, Upcroft B, Abbeel P, Burgard W, Milford M et al (2018) The limits and potentials of deep learning for robotics. Int J Robot Res 37(4–5):405–420

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2818–2826

Tagasovska N, Lopez-Paz D (2019) Single-model uncertainties for deep learning. In: Advances in neural information processing systems 32

Tassi N, Rovile C (2019) Bayesian convolutional neural network: Robustly quantify uncertainty for misclassifications detection. In: Mediterranean conference on pattern recognition and artificial intelligence, Springer, pp 118–132

Tchuiev V, Indelman V (2018) Inference over distribution of posterior class probabilities for reliable Bayesian classification and object-level perception. IEEE Robot Autom Lett 3(4):4329–4336

Teh YW, Thiery AH, Vollmer SJ (2016) Consistency and fluctuations for stochastic gradient Langevin dynamics. J Mach Learn Res 17:1–33

Thrun S (2002) Probabilistic robotics. Commun ACM 45(3):52–57

Thrun S, Fox D, Burgard W, Dellaert F (2001) Robust Monte Carlo localization for mobile robots. Artif Intell 128(1–2):99–141

Thulasidasan S, Chennupati G, Bilmes JA, Bhattacharya T, Michalak S (2019) On mixup training: improved calibration and predictive uncertainty for deep neural networks. In: Advances in neural information processing systems 32

Tishby N, Levin E, Solla SA (1989) Consistent inference of probabilities in layered networks: predictions and generalization. In: International joint conference on neural networks, IEEE, pp 403–409

Tran D, Kucukelbir A, Dieng AB, Rudolph M, Liang D, Blei DM (2016) Edward: a library for probabilistic modeling, inference, and criticism. arXiv preprint arXiv:1610.09787

Tran D, Hoffman MD, Saurous RA, Brevdo E, Murphy K, Blei DM (2017) Deep probabilistic programming. In: International conference on learning representations

Triebel R, Grimmett H, Paul R, Posner I (2016) Driven learning for driving: how introspection improves semantic mapping. In: Robotics research. Springer, pp 449–465

Tsiligkaridis T (2021) Failure prediction by confidence estimation of uncertainty-aware Dirichlet networks. In: ICASSP 2021–2021 IEEE international conference on acoustics. Speech and signal processing (ICASSP), IEEE, pp 3525–3529

Tsiligkaridis T (2021b) Information robust Dirichlet networks for predictive uncertainty estimation. US Patent App. 17/064,046

Vaicenavicius J, Widmann D, Andersson C, Lindsten F, Roll J, Schön T (2019) Evaluating model calibration in classification. In: Proceedings of the 22nd international conference on artificial intelligence and statistics, PMLR, pp 3459–3467

Valdenegro-Toro, M. (2019). Deep sub-ensembles for fast uncertainty estimation in image classification. arXiv preprint arXiv:1910.08168. https://github.com/mvaldenegro/papersubensemblesimage-classification

Van Amersfoort J, Smith L, Teh YW, Gal Y (2020) Uncertainty estimation using a single deep deterministic neural network. In: International conference on machine learning, PMLR, pp 9690–9700

Van Westen C (2000) Remote sensing for natural disaster management. Int Arch Photogram Remote Sens 33(B7/4; PART 7):1609–1617

Vasudevan VT, Sethy A, Ghias AR (2019) Towards better confidence estimation for neural models. In: ICASSP 2019–2019 IEEE international conference on acoustics. Speech and signal processing (ICASSP), IEEE, pp 7335–7339

Venkatesh B, Thiagarajan JJ (2019) Heteroscedastic calibration of uncertainty estimators in deep learning. arXiv preprint arXiv:1910.14179

Vyas A, Jammalamadaka N, Zhu X, Das D, Kaul B, Willke TL (2018) Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In: Proceedings of the European conference on computer vision (ECCV), pp 550–564

Wang H, Yeung DY (2016) Towards Bayesian deep learning: a framework and some existing methods. IEEE Trans Knowl Data Eng 28(12):3395–3408

Wang H, Yeung DY (2020) A survey on Bayesian deep learning. ACM Comput Surv (CSUR) 53(5):1–37

Wang S, Clark R, Wen H, Trigoni N (2017) DeepVO: towards end-to-end visual odometry with deep recurrent convolutional neural networks. In: 2017 IEEE international conference on robotics and automation (ICRA), IEEE, pp 2043–2050

Wang G, Li W, Ourselin S, Vercauteren T (2018a) Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation. In: International MICCAI brainlesion workshop, Springer, pp 61–72

Wang KC, Vicol P, Lucas J, Gu L, Grosse R, Zemel R (2018b) Adversarial distillation of Bayesian neural network posteriors. In: International conference on machine learning, PMLR, pp 5190–5199

Wang G, Li W, Aertsen M, Deprest J, Ourselin S, Vercauteren T (2019) Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. Neurocomputing 338:34–45

Wei L, Wan S, Guo J, Wong KK (2017) A novel hierarchical selective ensemble classifier with bioinformatics application. Artif Intell Med 83:82–90

Welling M, Teh YW (2011) Bayesian learning via stochastic gradient Langevin dynamics. In: International conference on machine learning, PMLR, pp 681–688

Wen Y, Tran D, Ba J (2019) BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning. In: International conference on learning representations

Wen Q, Sun L, Yang F, Song X, Gao J, Wang X, Xu H (2021a) Time series data augmentation for deep learning: a survey. In: Proceedings of the thirtieth international joint conference on artificial intelligence. Survey track, pp 4653–4660

Wen Y, Jerfel G, Muller R, Dusenberry MW, Snoek J, Lakshminarayanan B, Tran D (2021b) Combining ensembles and data augmentation can harm your calibration. In: International conference on learning representations

Wenger J, Kjellström H, Triebel R (2020) Non-parametric calibration for classification. In: Proceedings of the 23rd international conference on artificial intelligence and statistics, PMLR, pp 178–190

Wenzel F, Roth K, Veeling B, Swiatkowski J, Tran L, Mandt S, Snoek J, Salimans T, Jenatton R, Nowozin S (2020) How good is the Bayes posterior in deep neural networks really? In: International conference on machine learning, PMLR, pp 10248–10259

Willard J, Jia X, Xu S, Steinbach M, Kumar V (2020) Integrating physics-based modeling with machine learning: a survey. arXiv preprint arXiv:2003.04919

Wilson AG, Izmailov P (2020) Bayesian deep learning and a probabilistic perspective of generalization. In: Advances in neural information processing systems 33

Wong K, Wang S, Ren M, Liang M, Urtasun R (2020) Identifying unknown instances for autonomous driving. In: Conference on robot learning, PMLR, pp 384–393

Wu A, Nowozin S, Meeds E, Turner RE, Hernández-Lobato JM, Gaunt AL (2018) Deterministic variational inference for robust Bayesian neural networks. In: International conference on learning representations

Wu Q, Li H, Li L, Yu Z (2019) Quantifying intrinsic uncertainty in classification via deep Dirichlet mixture networks. arXiv preprint arXiv:1906.04450

Xiao H, Rasul K, Vollgraf R (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747

Yang S, Fevens T (2021) Uncertainty quantification and estimation in medical image classification. In: Artificial neural networks and machine learning–ICANN 2021: 30th international conference on artificial neural networks, Bratislava, Slovakia, September 14–17, 2021, proceedings, Part III 30, Springer, pp 671–683

Yang J, Wang F (2020) Auto-ensemble: an adaptive learning rate scheduling based deep learning model ensembling. IEEE Access 8:217,499-217,509

Yang N, Stumberg Lv, Wang R, Cremers D (2020) D3vo: deep depth, deep pose and deep uncertainty for monocular visual odometry. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1281–1292

Yao J, Pan W, Ghosh S, Doshi-Velez F (2019) Quality of uncertainty quantification for Bayesian neural network inference. arXiv preprint arXiv:1906.09686

Ye N, Zhu Z (2018) Stochastic fractional Hamiltonian Monte Carlo. In: Proceedings of the 27th international joint conference on artificial intelligence, pp 3019–3025

Ye N, Zhu Z, Mantiuk R (2017) Langevin dynamics with continuous tempering for training deep neural networks. In: Advances in neural information processing systems 30

Yu Q, Aizawa K (2019) Unsupervised out-of-distribution detection by maximum classifier discrepancy. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9518–9526

Yun P, Liu M (2023) Laplace approximation based epistemic uncertainty estimation in 3d object detection. In: Conference on robot learning, PMLR, pp 1125–1135

Zadrozny B, Elkan C (2001) Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In: International conference on machine learning, PMLR, pp 609–616

Zeng J, Lesnikowski A, Alvarez JM (2018) The relevance of Bayesian layer positioning to model uncertainty in deep Bayesian active learning. arXiv preprint arXiv:1811.12535

Zhang Y, Sutton C (2011) Quasi-Newton methods for Markov chain Monte Carlo. In: Advances in neural information processing systems 24

Zhang G, Sun S, Duvenaud D, Grosse R (2018a) Noisy natural gradient as variational inference. In: International conference on machine learning, PMLR, pp 5852–5861

Zhang H, Cisse M, Dauphin YN, Lopez-Paz D (2018b) Mixup: beyond empirical risk minimization. In: International conference on learning representations

Zhang Z, Dalca AV, Sabuncu MR (2019) Confidence calibration for convolutional neural networks using structured dropout. arXiv preprint arXiv:1906.09551

Zhang J, Kailkhura B, Han TYJ (2020) Mix-n-match: ensemble and compositional methods for uncertainty calibration in deep learning. In: International conference on machine learning, PMLR, pp 11,117–11,128

Zhao X, Ou Y, Kaplan L, Chen F, Cho JH (2019) Quantifying classification uncertainty using regularized evidential neural networks. arXiv preprint arXiv:1910.06864

Zhao J, Liu X, He S, Sun S (2020) Probabilistic inference of Bayesian neural networks with generalized expectation propagation. Neurocomputing 412:392–398

Zhu XX, Tuia D, Mou L, Xia GS, Zhang L, Xu F, Fraundorfer F (2017) Deep learning in remote sensing: a comprehensive review and list of resources. IEEE Geosci Remote Sens Mag 5(4):8–36

Zou D, Xu P, Gu Q (2018) Stochastic variance-reduced Hamilton Monte Carlo methods. In: International conference on machine learning, PMLR, pp 6028–6037

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Jakob Gawlikowski[1,2] · Cedrique Rovile Njieutcheu Tassi[3,4] · Mohsin Ali[2,5] · Jongseok Lee[6] · Matthias Humt[3,6] · Jianxiang Feng[3,6] · Anna Kruspe[2] · Rudolph Triebel[3,6] · Peter Jung[2,4,7] · Ribana Roscher[2,8] · Muhammad Shahzad[2] · Wen Yang[9] · Richard Bamler[10] · Xiao Xiang Zhu[2]

✉ Xiao Xiang Zhu
xiaoxiang.zhu@tum.de

[1] Institute of Data Science, German Aerospace Center, Mälzerstraße 5-7, 07745 Jena, Germany

[2] Data Science in Earth Observation, Technical University of Munich, Arcisstraße 21, 80333 Munich, Germany

[3] Department of Informatics, Technical University of Munich, Boltzmannstrasse 3, 85748 Garching, Germany

[4] Institute of Optical Sensor Systems, German Aerospace Center, Rutherfordstraße 2, 12489 Berlin, Germany

[5] Remote Sensing Technology Institute, German Aerospace Center, Münchener Straße 20, 82234 Weßling, Germany

[6] Institute of Robotics and Mechatronics, German Aerospace Center, Münchener Straße 20, 82234 Weßling, Germany

[7] Department of Telecommunication Systems, Technical University Berlin, Einsteinufer 25, 10587 Berlin, Germany

[8] Institute of Geodesy and Geoinformation, Rheinische Friedrich-Wilhelms-Universität Bonn, Niebuhrstr. 1a, 53113 Bonn, Germany

[9] School of Electronic Information, Wuhan University, Wuhan 430072, China

[10] Excellence Senior Faculty, Technical University of Munich, Arcisstraße 21, 80333 Munich, Germany