A Comparison of Uncertainty Estimation Approaches in Deep Learning Components for Autonomous Vehicle Applications

Fabio Arnez[1] , **Huascar Espinoza**[1] , **Ansgar Radermacher**[1] and **François Terrier**[1]

[1]CEA LIST, Gif-sur-Yvette, France

{fabio.arnez, huascar.espinoza, ansgar.radermacher, francois.terrier}@cea.fr

Abstract

A key factor for ensuring safety in Autonomous Vehicles (AVs) is to avoid any abnormal behaviors under undesirable and unpredicted circumstances. As AVs increasingly rely on Deep Neural Networks (DNNs) to perform safety-critical tasks, different methods for uncertainty quantification have recently been proposed to measure the inevitable source of errors in data and models. However, uncertainty quantification in DNNs is still a challenging task. These methods require a higher computational load, a higher memory footprint, and introduce extra latency, which can be prohibitive in safety-critical applications. In this paper, we provide a brief and comparative survey of methods for uncertainty quantification in DNNs along with existing metrics to evaluate uncertainty predictions. We are particularly interested in understanding the advantages and downsides of each method for specific AV tasks and types of uncertainty sources.

1 Introduction

In the last decade, Deep Neural Networks (DNNs) have witnessed great advances in real-world applications like Autonomous Vehicles (AVs) to perform complex tasks such as object detection and tracking or vehicle control. Despite substantial performance improvements introduced by DNNs, they still have significant safety shortcomings due to their complexity, opacity and lack of interpretability [McAllister et al., 2017]. In particular, DNNs are brittle to operational domain shift and even small data corruption or perturbations [Kuutti et al., 2020]. This impedes ensuring the reliability of the DNNs models, which is a precondition for safety-critical systems to ensure compliance with automotive industry safety standards and avoid jeopardizing human lives.

A concrete safety problem is to detect abnormal situations under uncertain environment conditions and DNN-specific unpredictability. These situations are difficult to analyze during system development phases, in a way that they can be properly mitigated at a real-time scale. Indeed, although a DNN model achieves great performance in a validation set from its operation environment, it is currently impossible to test and provide the same performance guarantees in all the possible environment configurations the system could encounter in the real world [Kuutti et al., 2020]. A common practice to overcome this problem is to use runtime monitoring of DNN components, so that safety can be ensured even if the component was not fully validated at design time [Henne et al., 2020; Koopman et al., 2019]. A central aspect to enable DNN monitoring is to provide a runtime treatment of uncertainties associated with DNN's predictions [McAllister et al., 2017; Koopman et al., 2019].

In this paper, we review common uncertainty estimation methods for DNNs and compare their performance and benefits for different AV tasks. These methods offer a potential solution for runtime DNN confidence prediction and detection of Out-of-Distribution (OOD) samples, since prediction probability scores in DNNs do not provide a true representation of uncertainty [Mohseni et al., 2019]. However, these methods still demand a high computational load, incorporate extra latency, and require a larger memory footprint. We compare these factors since they can represent a major impediment in safety-critical applications with tight time constraints and limited computation hardware. We also briefly focus on surveying uncertainty metrics that evaluate the performance of quantification methods, as another critical factor to ensure safety in AV systems.

The remainder of the paper is structured as follows. Section 2 describes the sources of uncertainty in deep learning for AVs. Section 3 presents a comparison of recent works in AV tasks that include uncertainty estimation methods for DNNs. It provides a brief review of common uncertainty estimation methods in deep learning as well as metrics for predictive uncertainty evaluation in classification and regression tasks. Section 4 discusses the open challenges and possible directions for future work.

2 Background

2.1 Sources of Uncertainty in Deep Learning for Autonomous Vehicles

Autonomous vehicles have to deal with dynamic, non-stationary and highly unpredictable operational environments. Taking into account all the details from the operational environment at design time is an intractable task. Instead, the operational environment is constrained in a way

that it considers only a subset of all possible situations that the system can encounter in operation. This process is known as Operational Design Domain (ODD) adoption [Koopman and Fratrik, 2019], and safety requirements are built on the top of the ODD specification.

Given the constrained operational environment within system ODD, ensuring safety in an AV requires the identification of unfamiliar contexts by modeling AV's uncertainty [McAllister *et al.*, 2017]. However, there are many factors, not only related to the environment, that affect the system performance by introducing some degree of uncertainty. [Czarnecki and Salay, 2018] identify a set of factors that contribute to uncertainty in the perception function in an AV, and in this manner affect its performance. From this set, we take special attention to sensor properties, model uncertainty, situation and scenario coverage, and operational domain uncertainty factors. In the context of DNNs, the first two factors can be modeled by using uncertainty estimation methods, while the last two correspond to some degree of dataset shift (i.e. breaking the *independent and identically distributed* assumption between training and testing data.) and Out-of-Distribution (OOD) samples [Quionero-Candela *et al.*, 2009; Mohseni *et al.*, 2019].

Sensor properties like range, resolution, noise characteristics, and calibration can influence the amount of information in the samples delivered to a machine learning model during training or testing. In consequence, the effect of these properties are captured as noise and ambiguities inherent to the obtained samples. This type of noise in the data is known as *Aleatoric* uncertainty, and represents the incapability of completely sensing all the details of the environment [Kendall and Gal, 2017; Lee *et al.*, 2019b; Gustafsson *et al.*, 2019]. Aleatoric uncertainty can be further further classified into *homoscedastic* uncertainty (uncertainty that remains constant for different samples), and *heteroscedastic* uncertainty (uncertainty that can vary between samples).

Model uncertainty is often referred to as *Epistemic* uncertainty, and accounts for uncertainty in the model parameters. This type of uncertainty captures the ignorance of the model as a consequence of a dataset that does not represent the ODD well, or that is not sufficiently large [Kendall and Gal, 2017; Lee *et al.*, 2019b]. Epistemic uncertainty is expected to increase in unknown situations (e.g. different environment ODD conditions such as weather or lightning), and can be explained away by incorporating more data.

Situation and scenario coverage is related to the degree in which situations and scenarios from an ODD are reflected in training and operation stages; while operational domain uncertainty refers to a discrepancy between ODD situations and scenarios present at training and those encountered at operation (e.g. scenarios from two different ODDs) [Czarnecki and Salay, 2018]. In both cases, uncertainty can be reduced by incorporating more data, or by adjusting the ODD specification. However, it is extremely important to detect and discover OOD samples (i.e. outliers), especially those that have not been seen before, since those can lead to highly confident predictions that are wrong, i.e., the *unknown-unknowns* [Bansal and Weld, 2018].

In a similar fashion as the cases presented before, automotive industry standard ISO/PAS 21448 or SOTIF (Safety Of The Intended Functionality) [ISO, 2019], provides a process to identify unknown and potentially unsafe scenarios to minimize the risk by recognizing the performance limitations from sensors, algorithms, or user misuse. Unsafe scenarios can be further classified into unsafe-known (e.g out of ODD samples) or unsafe-unknown (e.g. OOD samples). Once an unknown-unsafe scenario or situation is identified, it becomes a known-unsafe scenario that can be mitigated at design time [Rau *et al.*, ; Mohseni *et al.*, 2019].

2.2 Uncertainty Estimation Methods for DNNs

In recent years, many probabilistic deep learning methods have been proposed to obtain an uncertainty measure from an approximation to the (highly multi-modal) predictive distribution, as well as methods for calibrating the outputs of DNNs. In general, there are two approaches for DNN predictive uncertainty calculation: sampling-based and sampling-free methods. Sampling-based methods rely on taking multiple predictive samples based on the same input to get the estimator that will be associated with uncertainty. Sampling-free methods require one single predictive output. These methods are further discussed in Section 3.

Neural Network Calibration

Confidence calibration represents the degree to which a model's predicted probability estimates the true correctness likelihood [Guo *et al.*, 2017]. Under ideal circumstances, we expect that the normalized outputs from a DNN (i.e softmax outputs) correspond to the true correctness likelihood [Guo *et al.*, 2017]. From a frequentist perspective, this can be viewed as a discrepancy measure between local confidence (or uncertainty) predictions and the expected performance in the long-run [Hubschneider *et al.*, 2019; Lakshminarayanan *et al.*, 2017]. For example, we expect that a class predicted with probability p is correct $p\%$ of the time, i.e. from 100 samples predicted with confidence 0.9, we expect 90 correct predictions. DDNs can be calibrated by using *Temperature Scaling*, a simple post-processing technique [Guo *et al.*, 2017], or more recently, *Dirichelt calibration* [Kull *et al.*, 2019]. For a regression setting, [Kuleshov *et al.*, 2018; Hubschneider *et al.*, 2019] formalize the calibration notion for continuous variables, in which a $p\%$ confidence interval should contain the true outcome $p\%$ of the time.

Despite the improvements achieved with calibration methods, they can not be seen as a complete solution for uncertainty estimation problem, since calibration is performed relative to a validation dataset [Kull *et al.*, 2019; Ashukha *et al.*, 2020] (i.e., calibration methods rely on in-distribution samples to learn a calibration map). In the presence of OOD samples, a model is no longer calibrated. This limits the contribution of calibration techniques to scenarios where huge training datasets are available.

3 Comparison of Uncertainty Estimation Methods in AV Domain

In this section, we compare and analyze some common uncertainty estimation methods in terms of out-of-the-box calibration in the predictions (i.e. without a prior calibration), computational budget, memory footprint, and required changes in the DNN for applying each method (architecture, loss function, and others). We have chosen the most representative works to the best of our knowledge in each application. Some of the listed works introduce improvements by performing combinations between other methods. This is summarized in Table 1.

3.1 Methods Limited to Aleatoric Uncertainty

The first four methods listed in Table 1 exclusively deal with aleatoric uncertainty. In classification tasks, uncertainty is usually represented by normalized logits at the output layer (e.g. softmax output) which can be interpreted as a probability distribution related to aleatoric uncertainty [Gustafsson *et al.*, 2019]. Unfortunately, normalized outputs as probability distributions fail to capture model uncertainty and this very often results in overconfident predictions that are wrong [Guo *et al.*, 2017], especially in the presence of dataset-shift. To overcome the problems of softmax, [Gast and Roth, 2018] propose to use a Dirichlet distribution instead.

In a regression configuration, deep learning models do not have an uncertainty representation by default. The outputs of a DNN are intended to parameterize a probability distribution (e.g., Gaussian, Laplace) to obtain a probabilistic representation. This modification of the architecture allows DNNs to learn aleatoric uncertainty from the data itself by using thes heteroscedastic loss and maximum likelihood [Kendall and Gal, 2017; Ilg *et al.*, 2018]. Similarly, in the heteroscedastic version of the classification, [Kendall and Gal, 2017] place a Gaussian distribution over the output logits (i.e., each logit with its respective variance), before the softmax layer is applied. An alternative approach replaces the input, output and activation functions of a DNN with probability distributions [Gast and Roth, 2018]. This method allows the propagation of a fixed uncertainty at the input to the output of the DNN employing *Assumed Density Filtering* (ADF).

3.2 Bayesian Neural Networks

Bayesian Neural Networks (BNNs), aim to learn a distribution over the weights instead of point estimates. In this way, we look for the posterior distribution of the weights given the data $p(\mathbf{w}|\mathcal{D})$, by applying Bayes' theorem from the data likelihood and a chosen prior distribution over the weights $p(\mathbf{w})$:

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{\int p(\mathcal{D}|\mathbf{w})p(\mathbf{w})d\mathbf{w}} \quad (1)$$

Given the predictive posterior distribution $p(\mathbf{w}|\mathcal{D})$, we obtain the predictive posterior distribution for a new input $\mathbf{x}^*$ by marginalizing over the model parameters:

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \int p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w} \quad (2)$$

Instead of relying on only one configuration of the weights, we use every possible configuration of the weights (all possible models) weighted by the posterior on the parameters, to make a prediction, i.e. $p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \mathbb{E}_{p(\mathbf{w}|\mathcal{D})}[p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{w})]$. This represents the Bayesian Model Average (BMA) and accounts for *epistemic uncertainty* [Wilson and Izmailov, 2020; Gal, 2016; Blundell *et al.*, 2015].

Unfortunately, the integrals from (1) and (2) are intractable. Thus, we must build a distribution that approximates the true posterior distribution on the weights, $q(w) \approx p(\mathbf{w}|\mathcal{D})$. Two main paradigms exist to build $q(w)$: *Markov Chain Monte Carlo* (MCMC) and *Variational Inference* (VI) methods. In the former, the gold standard is Hamiltonian Monte Carlo (HMC), and other methods like Stochastic Gradient MCMC (SG-MCMC) have been explored. However, MCMC methods are in general hard to scale to large DNNs due to the high-dimensional and multi-modal posterior distribution [Gustafsson *et al.*, 2019]. In the latter case, VI methods approximate the posterior over the weights by approximating a simpler distribution $q_\phi(\mathbf{w})$ (e.g. a gaussian) parameterized by ϕ. The parameters of $q_\phi(\mathbf{w})$ are found by minimizing the KL-divergence to $p(\mathbf{w}|\mathcal{D})$.

A particular scalable and easy to implement sample-based method for approximate VI is Monte Carlo Dropout (MCD) [Gal and Ghahramani, 2016]. In this method, dropout regularization is also applied at test time, so that $q_\phi(\mathbf{w})$ is a Bernoulli distribution. Dropout is only performed in some of the deeper layers of the DNN to model better high-level features and to avoid slow training [Mukhoti and Gal, 2018; Kendall *et al.*, 2015]. Dropout probabilities can be set manually, or the network can tune dropout rates during training [Gal *et al.*, 2017].

All the MCD-related methods listed in Table 1 refer to this approximation of BNNs. It can be noted from the performance comparison criteria, that the need to take multiple forward passes (output samples) for the same input to approximate the distribution from Equation 2 represents a major impediment to safety-critical applications with tight time constraints and limited computation hardware.

To get a representation of both types of uncertainty (aleatoric and epistemic), the methods presented in Section 3.1 have been used in combination with MCD. For example, in a regression configuration, a set of T samples are taken from the predictions of a DNN that parameterize a distribution in its output: $\{\hat{\mathbf{y}}_\mathbf{t}, \hat{\sigma}_t\}_{t=1}^T$. However, since aleatoric uncertainty is learned from the data itself (by using the heteroscedastic loss), this approach could produce wrong uncertainty estimations in samples that include a higher level of uncertainty than that observed during training. Another approach presented in [Loquercio *et al.*, 2020], applies MCD to take samples from a DNN where the input, output and activation functions are replaced by probability distributions according to [Gast and Roth, 2018]. This method permits uncertainty propagation at the input to the output of the DNN using ADF (e.g., sensor noise can be propagated to the output of the DNNs). This is an appealing method for AV applications where sensor properties are commonly known. Interestingly, the authors show that this method can be applied to trained DNNs and is architecture agnostic.

Method	Autonomous Vehicle Task	Uncertainty Captured		Out-of-the box Calibration	Comparison Criteria		
		Aleatoric	Epistemic		Computational Budget	Memory Footprint	Changes in DNN
Softmax logits as parameters of a prob. dist.	- Object Detection [Feng et al., 2019b]	✓	✗	Bad	Fair	Slow	Small
Outputs as parameters of a prob. dist.	- Object Detection [Feng et al.,2019b]	✓	✗	Bad	Fair	Slow	Small
Inputs, activation and output as prob. dist. & ADF	- Optical Flow [Gast and Roth, 2018]	✓	✗	Undefined	Low	Low	Mid
Point estimate & MCD regression	- Steering Angle Prediction [Hubschneider et al., 2019; Michelmore et al., 2018; Michelmore et al., 2019]	✗	✓	Fair	Fair	Low	None
Softmax & MCD	- Traffic Sign Recognition [Henne et al., 2020] - Semantic Segmentation [Phan et al., 2019; Mukhoti and Gal, 2018; Gustafsson et al., 2019]	✗	✓	Fair	Fair	Low	None
Deep Ensembles	- Steering Angle Prediction [Hubschneider et al., 2019] - Traffic Sign Recognition [Henne et al., 2020] - Semantic Segmentation [Gustafsson et al., 2019] - Depth Estimation [Gustafsson et al., 2019]	✓	✓	Good	High	High	Small
Bootstrap Ensembles	- Steering Angle Prediction [Hubschneider et al., 2019] - Optical Flow [Ilg et al., 2018]	✓	✓	Bad	Fair	Fair	Mid
Softmax logits as parameters of a prob. dist. & MCD	- Object Detection [Feng et al., 2018]	✓	✓	Fair	High	Low	Small
Outputs as parameters of a prob. dist. & MCD	- Object Detection [Feng et al., 2018] - Steering Angle Prediction [Lee et al., 2019b; Lee et al., 2019c; Lee et al., 2019a] - Depth Estimation [Gustafsson et al., 2019]	✓	✓	Fair	High	Low	Small
Inputs, activation and output as prob. dist. & ADF &MCD	- Steering Angle Prediction [Loquercio et al., 2020]	✓	✓	Undefined	High	Low	Mid
MDNs	- Steering Angle Prediction [Hubschneider et al., 2019; Choi et al., 2018] - Future Prediction [Makansi et al., 2019]	✓	✓	Bad	Low	Low	None
MDNs with stages	- Future Prediction [Makansi et al., 2019]	✓	✓	Undefined	Low	Low	High

Table 1: Uncertainty Estimation Methods Comparison

3.3 Deep Ensembles

A Deep Ensemble (DE) is another sample-based method, in which M DNNs are trained to obtain the predictive distribution $p(\mathbf{y}|\mathbf{x})$ [Lakshminarayanan *et al.*, 2017]. Each DNN learns a set of parameters $\mathbf{w}$ that are point estimates, starting for different random initialization and repeating the minimization M times. In an ensemble, predictions are averaged and can be considered as a mixture model that is equally weighted:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} p(\mathbf{y}|\mathbf{x}, \hat{\mathbf{w}}^i), \; \{\hat{\mathbf{w}}^{(i)}\}_{i=1}^{M} \quad (3)$$

For classification, equation (3) corresponds to an average of the softmax probabilities. For regression, the outputs that parameterize a probability distribution are averaged to represent the mean and variance of the mixture. In this manner, both types of uncertainty (aleatoric and epistemic) can be easily captured. Although DE is considered a non-Bayesian method, expression (3) represents an approximation of (2) since $\{\hat{\mathbf{w}}^{(i)}\}_{i=1}^{M}$ can be seen as samples taken from distribution that approximates the true posterior, by exploring different modes of from $p(\mathbf{w}|\mathcal{D})$ [Fort *et al.*, 2019; Wilson and Izmailov, 2020].

As presented in Table 1, the DE method tends to outperform approximate Bayesian inference methods like MCD, for both, uncertainty estimates and accuracy [Gustafsson *et al.*, 2019]. A recent work from [Snoek *et al.*, 2019] also shows, that DE is more robust to dataset shift. These works suggest that DE should be considered as the new standard method for predictive distributions and uncertainty estimation. However, DE has some drawbacks, especially if the target application is a safety-critical application. DE requires a higher computational load and a larger memory footprint, as shown in Table 1. For the training and testing stage, the number of parameters, and the inference times scale linearly with M. To mitigate this problem, [Osband *et al.*, 2016] propose a fused version of ensembles with multiple heads. All the heads share the convolutional layers (feature extractors) and each head is trained using boostrap samples.

3.4 Mixture Density Networks

Mixture Density Networks (MDN) [Bishop, 1994], is a sample-free method for regression tasks, where the aim is to train a DNN that predicts the parameters of a Gaussian Mixture Model (GMM) given an input $\mathbf{x}$. A GMM is formed by a weighted sum of K Gaussians, to model the conditional distribution:

$$p(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{K} \pi_i(\mathbf{x}) \mathcal{N}(\mathbf{y}|\mu_i(\mathbf{x}), \Sigma_i(\mathbf{x})) \quad (4)$$

where $\pi_i(\mathbf{x}), \mu_i(\mathbf{x}), \Sigma_i(\mathbf{x})$ represent the set of parameters of the GMM as a function of the input $\mathbf{x}$ for K mixtures. For training, *Negative Log-likelihood* (NLL) is used as loss function.

By using the law of total variance, [Choi *et al.*, 2018] formalized the acquisition of aleatoric and epistemic uncertainty in MDNs. As a first step, the expectation of the GMM is obtained as a combination of the mixture components in a weighted sum: $\mathbb{E}[\mathbf{y}|\mathbf{x}] = \sum_{i=1}^{K} \pi_i(\mathbf{x})\mu_i(\mathbf{x})$. The predicted variance is composed of the weighted sum of the variances and the weighted variances of the means:

$$\mathbb{V}[\mathbf{y}|\mathbf{x}] = \sum_{i=1}^{K} \pi_i(\mathbf{x})\Sigma_i(\mathbf{x}) + \sum_{i}^{K} \pi_i(\mathbf{x}) \left\| \mu_i(\mathbf{x}) - \sum_{i}^{K} \pi_i(\mathbf{x})\mu_i(\mathbf{x}) \right\|^2 \quad (5)$$

where the first term represents the aleatoric uncertainty and the second term represents the epistemic uncertainty. We refer the reader to [Choi *et al.*, 2018] for more details about uncertainty acquisition in MDNs.

As pointed out in Table 1, the sampling-free nature of this method reduces the computation load, memory footprint, and permits complex distribution modeling with respect to the methods described before. These characteristics are attractive for real-time applications. However, MDNs suffer from numerical instability for high dimensional problems and mode collapse when using regularization techniques [Makansi *et al.*, 2019].

3.5 Quality Metrics for Uncertainty Estimation

In this section, we discuss common metrics for evaluating the quality of uncertainty estimation.

Classification Metrics. Different methods for uncertainty estimation exist for classification tasks. *Variation Ratio* and information metrics such as *Predictive Entropy, Mutual Information*, can be used in classification settings to represent uncertainty [Gal, 2016]. Variation ratio is a measure of dispersion; mutual information captures model confidence, and predictive entropy accounts for epistemic and aleatoric uncertainty [Mukhoti and Gal, 2018; Michelmore *et al.*, 2018; Phan *et al.*, 2019]. [Mukhoti and Gal, 2018] propose specific performance metrics for semantic segmentation to evaluate Bayesian models. Since there is no ground-truth for uncertainty estimation, [Snoek *et al.*, 2019; Lakshminarayanan *et al.*, 2017] argue that proper scoring rules are NLL and *Brier* score. NLL depends on predictive uncertainty and is commonly evaluated in a held-out set, however, it can overestimate tail probabilities; whereas Brier-score measures the accuracy of predictive probabilities by a sum of squared differences between the predicted probability vector and the target, nonetheless, this score is prone to avoid capturing infrequent events. Other evaluation metrics independent of score values are: the *Area Under the Receiver Operating Characteristic* (AUROC), *Area Under Precision Recall Curve* (AUPRC), and Area Under Risk-Coverage (AURC) [Hendrycks and Gimpel, 2016; Ding *et al.*, 2019].

Regression Metrics. Similarly, in regression tasks, NLL is a proper scoring rule for a likelihood that follows Gaussian distribution [Lakshminarayanan *et al.*, 2017; Kendall and Gal, 2017]. Furthermore, [Ilg *et al.*, 2018] introduces a relative measure for uncertainty estimation, the *Area Under the Sparsification Error* (AUSE) curve, that measures the difference between the dispersion of predictions (affected by predictive uncertainty), and a oracle in terms of true prediction error, e.g. *Root Mean Squared Error* (RMSE) [Gustafsson *et al.*, 2019].

Calibration Metrics. For classification tasks, common quality metrics are *Expected Calibration Error* (ECE) and *Maximum Calibration Error* (MCE) [Guo *et al.*, 2017]. The former measures the difference between expected accuracy and expected confidence; the latter identifies the largest discrepancy between accuracy and confidence, which is of particular interest in safety-critical applications. For a regression configuration, [Kuleshov *et al.*, 2018] use calibration error as a metric that represents the sum of weighted squared differences between the expected and observed (empirical) confidence levels; correspondingly in [Gustafsson *et al.*, 2019], the authors propose to use the *Area Under the Calibration Error* curve (AUCE) as an absolute measure of uncertainty. The before-mentioned authors use reliability diagrams (i.e. calibration plots) to get a visual representation of model calibration. Regardless of drawbacks with OOD samples, calibration plots and measures are used extensively to compare the predictive quality of other uncertainty estimation methods.

3.6 Considerations per AV Task Type

In the context of AVs, for (end-to-end) steering angle prediction, a broad variety of uncertainty estimation methods have been applied. In some works only epistemic uncertainty was captured by using MCD [Michelmore *et al.*, 2018; Michelmore *et al.*, 2019]. However, usually both types of uncertainty are captured [Lee *et al.*, 2019b; Lee *et al.*, 2019c; Lee *et al.*, 2019a] by using the method proposed by [Kendall and Gal, 2017], or by using DE, boostrap ensembles, or MDNs. The calibration plots presented in [Hubschneider *et al.*, 2019] show that MCD has better out-of-the-box calibration than bootstrap ensembles or MDNs; the last two methods are overconfident in their predictions. In this particular task, safety mechanisms have been proposed when uncertainty estimations surpass a given or learned threshold in order to improve vehicle safety [Michelmore *et al.*, 2018; Michelmore *et al.*, 2019; Lee *et al.*, 2019b].

Under the modular pipeline paradigm for AV control, probabilistic modeling has mainly been applied to perception tasks like object detection from 3D Lidar, semantic segmentation and depth estimation. For 3D object detection from Lidar point-clouds, [Feng *et al.*, 2018] estimate aleatoric and epistemic uncertainty using the methods proposed by [Kendall and Gal, 2017]. However, epistemic uncertainty estimation with MCD introduces a high computational cost. A later work from [Feng *et al.*, 2019b] leverages aleatoric uncertainties to greatly improve the performance and reduce the computational load from MCD. In [Feng *et al.*, 2019a] the authors show that predictions for classification and regression are miscalibrated, and propose methods to fix calibration of DNNs and produce better uncertainty estimates.

For semantic segmentation, [Phan *et al.*, 2019; Mukhoti and Gal, 2018; Gustafsson *et al.*, 2019] model aleatoric uncertainty from the softmax output, and epistemic uncertainty by using MCD or ensembles. Common uncertainty metrics in this case are predictive entropy and mutual information [Mukhoti and Gal, 2018]. For Depth estimation, [Gustafsson *et al.*, 2019] compares DE with the heteroscedastic regression in combination with MCD [Kendall and Gal, 2017]. In both previous tasks (semantic segmentation and depth estimation)

DE achieves better performance and calibration than MCD variants [Gustafsson *et al.*, 2019]. However, in DE the computational cost at training and testing grows linearly with the number of ensembles. Similarly for traffic sign recognition, DE exhibit the best-calibrated outputs, but in this case, MCD in combination with softmax also produces well-calibrated outputs close to that from DE [Henne *et al.*, 2020].

For optical flow, [Gast and Roth, 2018] capture aleatoric uncertainty by replacing the input, output and activation functions with probability distributions. This method allows propagating a fixed value of uncertainty at the input to the output of the DNN. [Ilg *et al.*, 2018] present an alternative approach, where DE and bootstrap ensembles were used to obtain the predictive uncertainty.

For future prediction, [Makansi *et al.*, 2019] propose an improvement to MDNs to predict the multi-modal distribution of positions of a vehicle in the future. This method presents two stages: a sampling and a fitting network. The former network receives the current position of the vehicle as an input and outputs a fixed number of hypotheses for future positions. The latter network fits a mixture distribution to the hypothesis estimated in the first network. This improvement helps to avoid mode collapse in MDNs, however, high dimensional outputs remain challenging for this approach.

4 Conclusions

We presented a comparative survey for uncertainty estimation methods for both, classification and regression tasks, in the AV domain. We also provide a general comparative analysis of these methods. From this analysis we can see that DE has become a gold-standard for uncertainty quantification in many AV tasks thanks to its high-quality uncertainty predictions and its robustness to OOD samples. However, the high computational load and large memory footprint, can hinder its use in safety-critical applications that have hardware limitations or tight time-constraints. Here, sampling-free methods are an interesting avenue for future research. New robust (to OOD) and lightweight approaches should be explored in the AV domain, to produce good-quality uncertainty estimates. We also observed that predictions from these methods are uncalibrated (overconfident or underconfident) and are usually applied to classification tasks. We encourage the application of calibration methods also for regression tasks by using the methods proposed by [Kuleshov *et al.*, 2018] instead of limiting the assessment of predictions with only reliability diagrams. We also suggest to study and compare uncertainty estimation methods under dataset-shift conditions to assess their robustness. For future work, we plan to incorporate uncertainty information into the *Responsability-Sensitive Safety* model [Shalev-Shwartz *et al.*, 2017]. This generalizes the approach from [Salay *et al.*, 2020] by considering component uncertainty from different AV subsystems and propagating it through them. These subsystems could include DNNs e.g. for planning and control.

References

[Ashukha *et al.*, 2020] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*, 2020.

[Bansal and Weld, 2018] Gagan Bansal and Daniel S Weld. A coverage-based utility model for identifying unknown unknowns. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Bishop, 1994] Christopher M Bishop. Mixture density networks. 1994.

[Blundell *et al.*, 2015] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622, 2015.

[Choi *et al.*, 2018] Sungjoon Choi, Kyungjae Lee, Sungbin Lim, and Songhwai Oh. Uncertainty-aware learning from demonstration using density networks with sampling-free variance modeling. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6915–6922. IEEE, 2018.

[Czarnecki and Salay, 2018] Krzysztof Czarnecki and Rick Salay. Towards a framework to manage perceptual uncertainty for safe automated driving. In *International Conference on Computer Safety, Reliability, and Security*, pages 439–445. Springer, 2018.

[Ding *et al.*, 2019] Yukun Ding, Jinglan Liu, Jinjun Xiong, and Yiyu Shi. Evaluation of neural network uncertainty estimation with application to resource-constrained platforms. *arXiv preprint arXiv:1903.02050*, 2019.

[Feng *et al.*, 2018] Di Feng, Lars Rosenbaum, and Klaus Dietmayer. Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3266–3273. IEEE, 2018.

[Feng *et al.*, 2019a] Di Feng, Lars Rosenbaum, Claudius Glaeser, Fabian Timm, and Klaus Dietmayer. Can we trust you? on calibration of a probabilistic object detector for autonomous driving. *arXiv preprint arXiv:1909.12358*, 2019.

[Feng *et al.*, 2019b] Di Feng, Lars Rosenbaum, Fabian Timm, and Klaus Dietmayer. Leveraging heteroscedastic aleatoric uncertainties for robust real-time lidar 3d object detection. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1280–1287. IEEE, 2019.

[Fort *et al.*, 2019] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

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

[Gal *et al.*, 2017] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Advances in neural information processing systems*, pages 3581–3590, 2017.

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

[Gast and Roth, 2018] Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3369–3378, 2018.

[Guo *et al.*, 2017] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

[Gustafsson *et al.*, 2019] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schön. Evaluating scalable bayesian deep learning methods for robust computer vision. *arXiv preprint arXiv:1906.01620*, 2019.

[Hendrycks and Gimpel, 2016] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[Henne *et al.*, 2020] Maximilian Henne, Adrian Schwaiger, Karsten Roscher, and Gereon Weiss. Benchmarking uncertainty estimation methods for deep learning with safety-related metrics. 2020.

[Hubschneider *et al.*, 2019] Christian Hubschneider, Robin Hutmacher, and J Marius Zöllner. Calibrating uncertainty models for steering angle estimation. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1511–1518. IEEE, 2019.

[Ilg *et al.*, 2018] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 652–667, 2018.

[ISO, 2019] ISO ISO. Pas 21448-road vehicles-safety of the intended functionality. *International Organization for Standardization*, 2019.

[Kendall and Gal, 2017] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

[Kendall *et al.*, 2015] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[Koopman and Fratrik, 2019] Philip Koopman and Frank Fratrik. How many operational design domains, objects, and events? 2019.

[Koopman *et al.*, 2019] Philip Koopman, Beth Osyk, and Jack Weast. Autonomous vehicles meet the physical world: Rss, variability, uncertainty, and proving safety. In *International Conference on Computer Safety, Reliability, and Security*, pages 245–253. Springer, 2019.

[Kuleshov *et al.*, 2018] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. *arXiv preprint arXiv:1807.00263*, 2018.

[Kull *et al.*, 2019] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems*, pages 12295–12305, 2019.

[Kuutti *et al.*, 2020] Sampo Kuutti, Richard Bowden, Yaochu Jin, Phil Barber, and Saber Fallah. A survey of deep learning applications to autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[Lakshminarayanan *et al.*, 2017] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

[Lee *et al.*, 2019a] Keuntaek Lee, Gabriel Nakajima An, Viacheslav Zakharov, and Evangelos A Theodorou. Perceptual attention-based predictive control. *arXiv preprint arXiv:1904.11898*, 2019.

[Lee *et al.*, 2019b] Keuntaek Lee, Kamil Saigol, and Evangelos A Theodorou. Early failure detection of deep end-to-end control policy by reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8543–8549. IEEE, 2019.

[Lee *et al.*, 2019c] Keuntaek Lee, Ziyi Wang, Bogdan Vlahov, Harleen Brar, and Evangelos A Theodorou. Ensemble bayesian decision making with redundant deep perceptual control policies. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 831–837. IEEE, 2019.

[Loquercio *et al.*, 2020] Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160, 2020.

[Makansi *et al.*, 2019] Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7144–7153, 2019.

[McAllister *et al.*, 2017] Rowan McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. International Joint Conferences on Artificial Intelligence, Inc., 2017.

[Michelmore *et al.*, 2018] Rhiannon Michelmore, Marta Kwiatkowska, and Yarin Gal. Evaluating uncertainty quantification in end-to-end autonomous driving control. *arXiv preprint arXiv:1811.06817*, 2018.

[Michelmore *et al.*, 2019] Rhiannon Michelmore, Matthew Wicker, Luca Laurenti, Luca Cardelli, Yarin Gal, and Marta Kwiatkowska. Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control. *arXiv preprint arXiv:1909.09884*, 2019.

[Mohseni *et al.*, 2019] Sina Mohseni, Mandar Pitale, Vasu Singh, and Zhangyang Wang. Practical solutions for machine learning safety in autonomous vehicles. *arXiv preprint arXiv:1912.09630*, 2019.

[Mukhoti and Gal, 2018] Jishnu Mukhoti and Yarin Gal. Evaluating bayesian deep learning methods for semantic segmentation. *arXiv preprint arXiv:1811.12709*, 2018.

[Osband *et al.*, 2016] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pages 4026–4034, 2016.

[Phan *et al.*, 2019] Buu Phan, Samin Khan, Rick Salay, and Krzysztof Czarnecki. Bayesian uncertainty quantification with synthetic data. In *International Conference on Computer Safety, Reliability, and Security*, pages 378–390. Springer, 2019.

[Quionero-Candela *et al.*, 2009] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[Rau *et al.*,] Paul Rau, Christopher Becker, and John Brewer. Approach for deriving scenarios for safety of the intended functionality.

[Salay *et al.*, 2020] Rick Salay, Krzysztof Czarnecki, Maria Soledad Elli, Ignacio J Alvarez, Sean Sedwards, and Jack Weast. Purss: Towards perceptual uncertainty aware responsibility sensitive safety with ml. In *SafeAI@AAAI*, pages 91–95, 2020.

[Shalev-Shwartz *et al.*, 2017] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. *arXiv preprint arXiv:1708.06374*, 2017.

[Snoek *et al.*, 2019] Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D Sculley, Joshua Dillon, Jie Ren, and Zachary Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13969–13980, 2019.

[Wilson and Izmailov, 2020] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.