# Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision

Fredrik K. Gustafsson[1]        Martin Danelljan[2]        Thomas B. Schön[1]

[1]Department of Information Technology, Uppsala University, Sweden
[2]Computer Vision Lab, ETH Zurich, Switzerland

## Abstract

*While deep neural networks have become the go-to approach in computer vision, the vast majority of these models fail to properly capture the uncertainty inherent in their predictions. Estimating this predictive uncertainty can be crucial, for example in automotive applications. In Bayesian deep learning, predictive uncertainty is commonly decomposed into the distinct types of aleatoric and epistemic uncertainty. The former can be estimated by letting a neural network output the parameters of a certain probability distribution. Epistemic uncertainty estimation is a more challenging problem, and while different scalable methods recently have emerged, no extensive comparison has been performed in a real-world setting. We therefore accept this task and propose a comprehensive evaluation framework for scalable epistemic uncertainty estimation methods in deep learning. Our proposed framework is specifically designed to test the robustness required in real-world computer vision applications. We also apply this framework to provide the first properly extensive and conclusive comparison of the two current state-of-the-art scalable methods: ensembling and MC-dropout. Our comparison demonstrates that ensembling consistently provides more reliable and practically useful uncertainty estimates. Code is available at* https://github.com/fregu856/evaluating_bdl.

## 1. Introduction

Deep Neural Networks (DNNs) have become the standard paradigm within most computer vision problems due to their astonishing predictive power compared to previous alternatives. Current applications include many safety-critical tasks, such as street-scene semantic segmentation [10, 6, 51], 3D object detection [42, 28] and depth completion [45, 32]. Since erroneous predictions can have disastrous consequences, such applications require an accurate measure of the predictive uncertainty. The vast majority

of these DNN models do however fail to properly capture the uncertainty inherent in their predictions. They are thus not fully capable of the type of *uncertainty-aware* reasoning that is highly desired e.g. in automotive applications.

The approach of *Bayesian deep learning* aims to address this issue in a principled manner. Here, predictive uncertainty is commonly decomposed into two distinct types, which both should be captured by the learned DNN [12, 25]. *Epistemic* uncertainty accounts for uncertainty in the DNN model parameters, while *aleatoric* uncertainty captures inherent and irreducible data noise. Input-dependent *aleatoric* uncertainty about the target $y$ arises due to e.g. noise and ambiguities inherent in the input $x$. This is present for instance in street-scene semantic segmentation, where image pixels at object boundaries are inherently ambiguous, and in 3D object detection where the location of a distant object is less certain due to noise and limited sensor resolution. In many computer vision applications, this aleatoric uncertainty can be effectively estimated by letting a DNN directly output the parameters of a certain probability distribution, modeling the conditional distribution $p(y|x)$ of the target given the input. For classification tasks, a predictive categorical distribution is commonly realized by a softmax output layer, although recent work has also explored Dirichlet models [14, 41, 34]. For regression, Laplace and Gaussian models have been employed [23, 9, 25, 27].

Directly predicting the conditional distribution $p(y|x)$ with a DNN does however not capture *epistemic* uncertainty, as information about the uncertainty in the model parameters is disregarded. This often leads to highly confident predictions that are incorrect, especially for inputs $x$ that are not well-represented by the training distribution [17, 27]. For instance, a DNN can fail to generalize to unfamiliar weather conditions or environments in automotive applications, but still generate confident predictions. Reliable estimation of epistemic uncertainty is thus of great importance. However, this task has proven to be highly challenging, largely due to the vast dimensionality of the parameter space, which renders standard

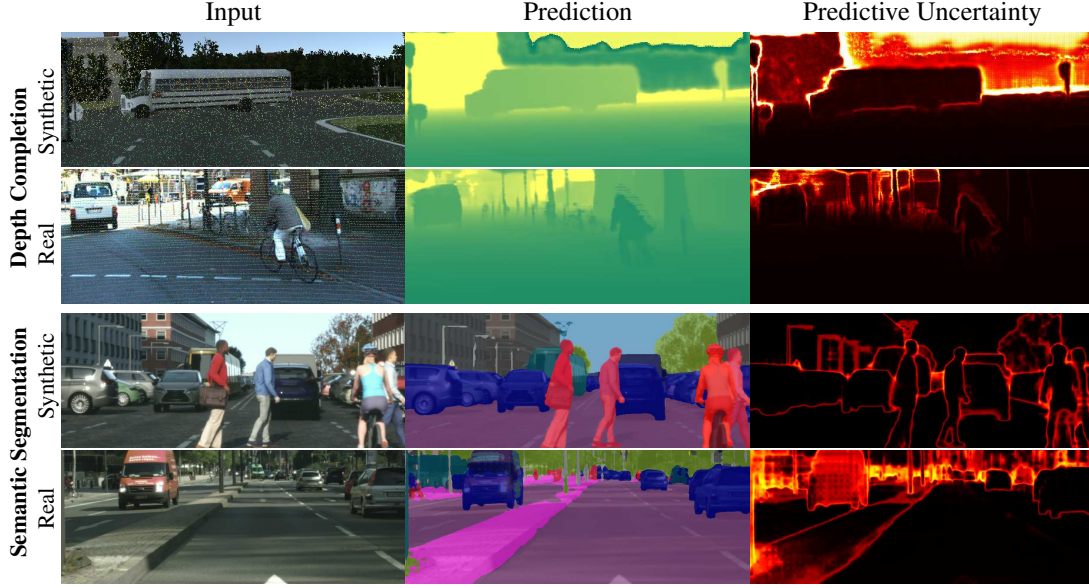| Input | Prediction | Predictive Uncertainty |

Figure 1: We propose a comprehensive evaluation framework for *scalable* epistemic uncertainty estimation methods in deep learning. The proposed framework employs state-of-the-art DNN models on the tasks of depth completion and street-scene semantic segmentation. All models are trained exclusively on synthetic data (the Virtual KITTI [11] and Synscapes [49] datasets). We here show the input (left), prediction (center) and estimated predictive uncertainty (right) for ensembling with $M = 8$ ensemble members, on both synthetic and real (the KITTI [15, 45] and Cityscapes [10] datasets) example validation images. Black pixels correspond to minimum predictive uncertainty, white pixels to maximum uncertainty.

Bayesian inference approaches intractable. To tackle this problem, a wide variety of approximations have been explored [38, 22, 4, 48, 7, 21, 47], but only a small number have been demonstrated to be applicable even to the large-scale DNN models commonly employed in *real-world* computer vision tasks. Among such *scalable* methods, MC-dropout [12, 25, 24, 36] and ensembling [27, 9, 23] are clearly the most widely employed, due to their demonstrated effectiveness and simplicity. While *scalable* techniques for epistemic uncertainty estimation recently have emerged, the research community however lacks a common and comprehensive evaluation framework for such methods. Consequently, both researchers and practitioners are currently unable to properly assess and compare newly proposed methods. In this work, we therefore accept this task and set out to design exactly such an evaluation framework, aiming to benefit and inspire future research in the field.

Previous studies have provided only partial insight into the performance of different *scalable* methods for epistemic uncertainty estimation. Kendall and Gal [25] evaluated MC-dropout alone on the tasks of semantic segmentation and monocular depth regression, providing mainly qualitative results. Lakshminarayanan et al. [27] introduced ensembling as a non-Bayesian alternative and found it to generally outperform MC-dropout. Their experiments were however based on relatively small-scale models and datasets, limiting the real-world applicability. Ilg et al. [23] compared en-

sembling and MC-dropout on the task of optical-flow estimation, but only in terms of the AUSE metric which is a *relative* measure of the uncertainty estimation quality. While finding ensembling to be advantageous, their experiments were also limited to a fixed number ($M = 8$) of ensemble members and MC-dropout forward passes, not allowing a completely fair comparison. Ovadia et al. [43] also fixed the number of ensemble members, and moreover only considered classification tasks. We improve upon this previous work and propose an evaluation framework that actually enables a conclusive ranking of the compared methods.

**Contributions**   We propose a comprehensive evaluation framework for *scalable* epistemic uncertainty estimation methods in deep learning. The proposed framework is specifically designed to test the robustness required in *real-world* computer vision applications, and employs state-of-the-art DNN models on the tasks of depth completion (regression) and street-scene semantic segmentation (classification). It also employs a novel combination of quantitative evaluation metrics which explicitly measures the reliability and practical usefulness of estimated predictive uncertainties. We apply our proposed framework to provide the first properly extensive and conclusive comparison of the two current state-of-the-art *scalable* methods: ensembling and MC-dropout. This comparison demonstrates that ensembling consistently outperforms the highly popular MC-dropout method. Our work thus suggests that ensembling

should be considered the new go-to approach, and encourages future research to understand and further improve its efficacy. Figure 1 shows example predictive uncertainty estimates generated by ensembling. Our framework can also directly be applied to compare other *scalable* methods, and we encourage external usage with publicly available code.

In our proposed framework, we predict the conditional distribution $p(y|x)$ in order to estimate input-dependent aleatoric uncertainty. The methods for epistemic uncertainty estimation are then compared by quantitatively evaluating the estimated predictive uncertainty in terms of the relative AUSE metric and the *absolute* measure of uncertainty calibration. Our evaluation is the first to include *both* these metrics, and furthermore we apply them to both regression and classification tasks. To provide a deeper and more fair analysis, we also study all metrics as functions of the number of samples $M$, enabling a highly informative comparison of the rate of improvement. Moreover, we simulate challenging real-world conditions found e.g. in automotive applications, where robustness to out-of-domain inputs is required to ensure safety, by training our models exclusively on synthetic data and evaluating the predictive uncertainty on real-world data. By analyzing this important domain shift problem, we significantly increase the practical applicability of our evaluation. We also complement our real-world analysis with experiments on illustrative toy regression and classification problems. Lastly, to demonstrate the evaluation rigor necessary to achieve a conclusive comparison, we repeat each experiment multiple times and report results together with the observed variation.

## 2. Predictive Uncertainty Estimation using Bayesian Deep Learning

DNNs have been shown to excel at a wide variety of supervised machine learning problems, where the task is to predict a target value $y \in \mathcal{Y}$ given an input $x \in \mathcal{X}$. In computer vision, the input space $\mathcal{X}$ often corresponds to the space of images. For classification problems, the target space $\mathcal{Y}$ consists of a finite set of $C$ classes, while a regression problem is characterized by a continuous target space, e.g. $\mathcal{Y} = \mathbb{R}^K$. For our purpose, a DNN is defined as a function $f_\theta : \mathcal{X} \to \mathcal{U}$, parameterized by $\theta \in \mathbb{R}^P$, that maps an input $x \in \mathcal{X}$ to an output $f_\theta(x) \in \mathcal{U}$. Next, we cover alternatives for estimating both the aleatoric and epistemic uncertainty in the predictions of DNN models.

**Aleatoric Uncertainty** In classification problems, aleatoric uncertainty is commonly captured by predicting a categorical distribution $p(y|x, \theta)$. This is implemented by letting the DNN predict logit scores $f_\theta(x) \in \mathbb{R}^C$, which are then normalized by a Softmax function,

$$p(y|x, \theta) = \mathrm{Cat}(y; s_\theta(x)),$$
$$s_\theta(x) = \mathrm{Softmax}(f_\theta(x)). \tag{1}$$

Given a training set of i.i.d. sample pairs $\mathcal{D} = \{X, Y\} = \{(x_i, y_i)\}_{i=1}^N$, $(x_i, y_i) \sim p(x, y)$, the data likelihood is obtained as $p(Y|X, \theta) = \prod_{i=1}^N p(y_i|x_i, \theta)$. The maximum-likelihood estimate of the model parameters, $\hat{\theta}_{\mathrm{MLE}}$, is obtained by minimizing the negative log-likelihood $-\sum_i \log p(y_i|x_i, \theta)$. For the Categorical model (1), this is equivalent to minimizing the well-known cross-entropy loss. At test time, the trained model predicts the distribution $p(y^\star|x^\star, \hat{\theta}_{\mathrm{MLE}})$ over the target class variable $y^\star$, given a test input $x^\star$. These DNN models are thus able to capture input-dependent aleatoric uncertainty, by outputting less confident predictions for inherently ambiguous cases.

In regression, the most common approach is to let the DNN directly predict targets, $y^\star = f_{\hat{\theta}}(x^\star)$. The parameters $\hat{\theta}$ are learned by minimizing e.g. the $L^2$ or $L^1$ loss over the training dataset [42, 28]. However, such direct regression does not model aleatoric uncertainty. Instead, recent work [23, 25, 27] has explored predicting the distribution $p(y|x, \theta)$, similar to the classification case above. For instance, $p(y|x, \theta)$ can be parameterized by a Gaussian distribution [9, 27], giving the following model in the 1D case,

$$p(y|x, \theta) = \mathcal{N}\big(y; \mu_\theta(x), \sigma_\theta^2(x)\big),$$
$$f_\theta(x) = [\,\mu_\theta(x) \quad \log \sigma_\theta^2(x)\,]^\mathsf{T} \in \mathbb{R}^2. \tag{2}$$

Here, the DNN predicts the mean $\mu_\theta(x)$ and variance $\sigma_\theta^2(x)$ of the target $y$. The variance is naturally interpreted as a measure of input-dependent aleatoric uncertainty. As in classification, the model parameters $\theta$ are learned by minimizing the negative log-likelihood $-\sum_i \log p(y_i|x_i, \theta)$.

**Epistemic Uncertainty** While the above models can capture aleatoric uncertainty, stemming from the data, they are agnostic to the uncertainty in the model parameters $\theta$. A principled means to estimate this epistemic uncertainty is to perform Bayesian inference. The aim is to utilize the posterior distribution $p(\theta|\mathcal{D})$, which is obtained from the data likelihood and a chosen prior $p(\theta)$ by applying Bayes' theorem. The uncertainty in the parameters $\theta$ is then marginalized out to obtain the predictive posterior distribution,

$$p(y^\star|x^\star, \mathcal{D}) = \int p(y^\star|x^\star, \theta) p(\theta|\mathcal{D}) d\theta$$
$$\approx \frac{1}{M} \sum_{i=1}^M p(y^\star|x^\star, \theta^{(i)}), \quad \theta^{(i)} \sim p(\theta|\mathcal{D}). \tag{3}$$

Here, the generally intractable integral in (3) is approximated using $M$ Monte Carlo samples $\theta^{(i)}$, ideally drawn from the posterior. In practice however, obtaining samples from the true posterior $p(\theta|\mathcal{D})$ is virtually impossible, requiring an approximate posterior $q(\theta) \approx p(\theta|\mathcal{D})$ to be used. We thus obtain the approximate predictive posterior as,

$$\hat{p}(y^\star|x^\star, \mathcal{D}) \triangleq \frac{1}{M} \sum_{i=1}^M p(y^\star|x^\star, \theta^{(i)}), \quad \theta^{(i)} \sim q(\theta), \tag{4}$$
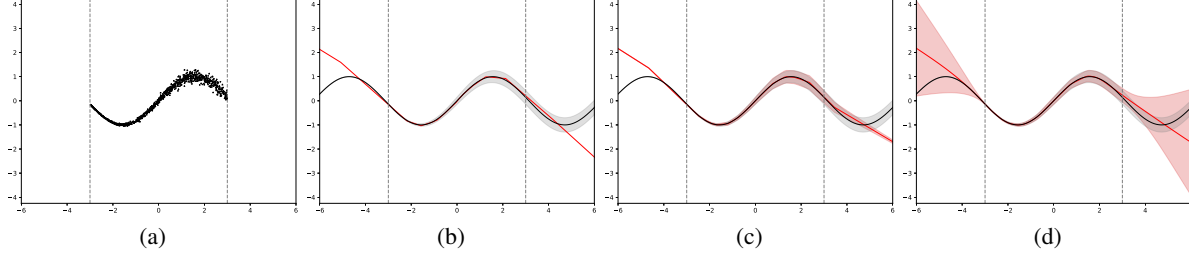
Figure 2: Toy regression problem illustrating the task of predictive uncertainty estimation with DNNs. The true data generator $p(y|x)$ is a Gaussian, where the mean is given by the solid black line and the variance is represented in shaded gray. The predictive mean and variance are given by the solid red line and the shaded red area, respectively. **(a)** Training dataset with $N = 1\,000$ examples. **(b)** A DNN trained to directly predict the target $y$ captures no notion of uncertainty. **(c)** A corresponding Gaussian DNN model (2) trained via maximum-likelihood captures *aleatoric* but not epistemic uncertainty. **(d)** The Gaussian model instead trained via approximate Bayesian inference (4) captures both *aleatoric* and *epistemic* uncertainty.

which enables us to estimate both aleatoric and epistemic uncertainty of the prediction. The quality of the approximation (4) depends on the number of samples $M$ and the method employed for generating $q(\theta)$. Prior work on such approximate Bayesian inference methods is discussed in Section 3. For the Categorical model (1), $\hat{p}(y^\star|x^\star, \mathcal{D}) = \text{Cat}(y^\star; \hat{s}(x^\star))$, $\hat{s}(x^\star) = \frac{1}{M}\sum_{i=1}^{M} s_{\theta^{(i)}}(x^\star)$. For the Gaussian model (2), $\hat{p}(y^\star|x^\star, \mathcal{D})$ is a uniformly weighted mixture of Gaussian distributions. We approximate this mixture with a single Gaussian, see Appendix A for details.

**Illustrative Example** To visualize and provide intuition for the problem of predictive uncertainty estimation with DNNs, we consider the problem of regressing a sinusoid corrupted by *input-dependent* Gaussian noise,

$$
\begin{aligned}
y &\sim \mathcal{N}\big(\mu(x), \sigma^2(x)\big), \\
\mu(x) &= \sin(x), \quad \sigma(x) = 0.15(1 + e^{-x})^{-1}.
\end{aligned}
\tag{5}
$$

Training data $\{(x_i, y_i)\}_{i=1}^{1000}$ is only given in the interval $[-3, 3]$, see Figure 2a. A DNN trained to directly predict the target $y$ is able to accurately regress the mean for $x^\star \in [-3, 3]$, see Figure 2b. However, this model does not capture any notion of uncertainty. A corresponding Gaussian DNN model (2) trained via maximum-likelihood obtains a predictive distribution that closely matches the ground truth for $x^\star \in [-3, 3]$, see Figure 2c. While correctly accounting for aleatoric uncertainty, this model generates overly confident predictions for inputs $|x^\star| > 3$ not seen during training. Finally, the Gaussian DNN model trained via approximate Bayesian inference (4), with a prior distribution $p(\theta) = \mathcal{N}(0, I_P)$ and $M = 1\,000$ samples obtained via Hamiltonian Monte Carlo [39], is additionally able to predict more reasonable uncertainties in the region with no available training data, see Figure 2d.

## 3. Related Work

Here, we discuss prior work on approximate Bayesian inference. We also note that ensembling, which is often considered a non-Bayesian alternative, in fact can naturally be viewed as an approximate Bayesian inference method.

**Approximate Bayesian Inference** The method employed for approximating the posterior $q(\theta) \approx p(\theta|\mathcal{D}) = p(Y|X,\theta)p(\theta)/p(Y|X)$ is a crucial choice, determining the quality of the approximate predictive posterior $\hat{p}(y^\star|x^\star, \mathcal{D})$ in (4). There exists two main paradigms for constructing $q(\theta)$, the first one being *Markov chain Monte Carlo (MCMC)* methods. Here, samples $\theta^{(i)}$ approximately distributed according to the posterior are obtained by simulating a Markov chain with $p(\theta|\mathcal{D})$ as its stationary distribution. For DNNs, this approach was pioneered by Neal [38], who employed Hamiltonian Monte Carlo (HMC) on small feed-forward neural networks. HMC entails performing Metropolis-Hastings [35, 19] updates using Hamiltonian dynamics based on the potential energy $U(\theta) \triangleq -\log p(Y|X,\theta)p(\theta)$. To date, it is considered a "gold standard" method for approximate Bayesian inference, but does not scale to large DNNs or large-scale datasets. Therefore, *Stochastic Gradient MCMC (SG-MCMC)* [33] methods have been explored, in which stochastic gradients are utilized in place of their full-data counterparts. SG-MCMC variants include Stochastic Gradient Langevin Dynamics (SGLD) [48], where samples $\theta^{(i)}$ are collected from the parameter trajectory given by the update equation $\theta_{t+1} = \theta_t - \alpha_t\nabla_\theta\widetilde{U}(\theta_t) + \sqrt{2\alpha_t}\epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, 1)$ and $\nabla_\theta\widetilde{U}(\theta)$ is the stochastic gradient of $U(\theta)$. Save for the noise term $\sqrt{2\alpha_t}\epsilon_t$, this update is identical to the conventional SGD update when minimizing the maximum-a-posteriori (MAP) objective $-\log p(Y|X,\theta)p(\theta)$. Similarly, Stochastic Gradient HMC (SGHMC) [7] corresponds to SGD with momentum injected with properly scaled noise. Given a limited computational budget, SG-MCMC methods can however struggle to explore the high-dimensional and highly multimodal posteriors of large DNNs. To mitigate this problem, Zhang et al. [53] proposed to use a cyclical stepsize schedule to help escaping local modes in $p(\theta|\mathcal{D})$.

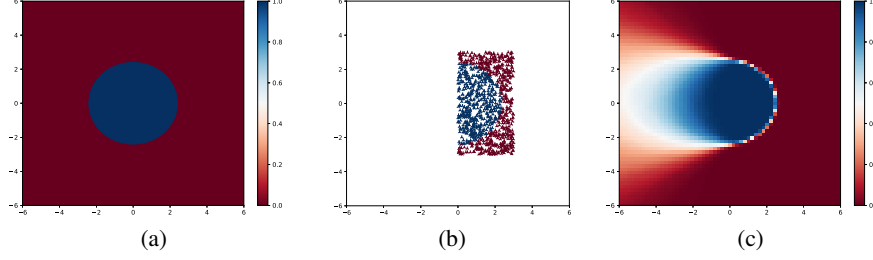The second paradigm is that of *Variational Inference*

Figure 3: Toy binary classification problem. **(a)** True data generator, red and blue represents the two classes. **(b)** Training dataset with $N = 1\,040$ examples. **(c)** "Ground truth" predictive distribution, obtained using HMC [39].

*(VI)* [22, 2, 16, 4]. Here, a distribution $q_\phi(\theta)$ parameterized by variational parameters $\phi$ is explicitly chosen, and the best possible approximation is found by minimizing the Kullback-Leibler (KL) divergence with respect to the true posterior $p(\theta|\mathcal{D})$. While principled, VI methods generally require sophisticated implementations, especially for more expressive variational distributions $q_\phi(\theta)$ [30, 31, 52]. A particularly simple and scalable method is MC-dropout [13]. It entails using dropout [44] also at test time, which can be interpreted as performing VI with a Bernoulli variational distribution [13, 24, 36]. The approximate predictive posterior $\hat{p}(y^\star|x^\star, \mathcal{D})$ in (4) is obtained by performing $M$ stochastic forward passes on the same input.

**Ensembling** Lakshminarayanan et al. [27] created a parametric model $p(y|x, \theta)$ of the conditional distribution using a DNN $f_\theta$, and learned multiple point estimates $\{\hat{\theta}^{(m)}\}_{m=1}^M$ by repeatedly minimizing the MLE objective $-\log p(Y|X, \theta)$ with random initialization. They then averaged over the corresponding parametric models to obtain the following predictive distribution,

$$\hat{p}(y^\star|x^\star) \triangleq \frac{1}{M} \sum_{m=1}^M p(y^\star|x^\star, \hat{\theta}^{(m)}). \qquad (6)$$

The authors considered this a non-Bayesian alternative to predictive uncertainty estimation. However, since the point estimates $\{\hat{\theta}^{(m)}\}_{m=1}^M$ always can be seen as samples from some distribution $\hat{q}(\theta)$, we note that (6) is virtually identical to the approximate predictive posterior in (4). Ensembling can thus naturally be viewed as approximate Bayesian inference, where the level of approximation is determined by how well the implicit sampling distribution $\hat{q}(\theta)$ approximates the posterior $p(\theta|\mathcal{D})$. Ideally, $\{\hat{\theta}^{(m)}\}_{m=1}^M$ should be distributed exactly according to $p(\theta|\mathcal{D}) \propto p(Y|X, \theta)p(\theta)$. Since $p(Y|X, \theta)$ is highly *multi-modal* in the parameter space for DNNs [1, 8], so is $p(\theta|\mathcal{D})$. By minimizing $-\log p(Y|X, \theta)$ multiple times, starting from *randomly chosen* initial points, we are likely to find different local optima. Ensembling can thus generate a compact set of samples $\{\hat{\theta}^{(m)}\}_{m=1}^M$ that, even for small values of $M$, captures this important aspect of multi-modality in $p(\theta|\mathcal{D})$.

## 4. Experiments

We conduct experiments both on illustrative toy regression and classification problems (Section 4.1), and on the real-world computer vision tasks of depth completion (Section 4.2) and street-scene semantic segmentation (Section 4.3). Our evaluation is motivated by real-world conditions found e.g. in automotive applications, where robustness to varying environments and weather conditions is required to ensure safety. Since images captured in these different circumstances could all represent distinctly different regions of the vast input image space, it is infeasible to ensure that all encountered inputs will be well-represented by the training data. Thus, we argue that robustness to out-of-domain inputs is crucial in such applications. To simulate these challenging conditions and test the robustness required for such real-world scenarios, we train all models on synthetic data and evaluate them on real-world data. To improve rigour of our evaluation, we repeat each experiment multiple times and report results together with the observed variation. A more detailed description of all results are found in the Appendix (Appendix B.3, C.2, D.2). All experiments are implemented in PyTorch [40].

### 4.1. Illustrative Toy Problems

We first present results on illustrative toy problems to gain insights into how ensembling and MC-dropout fare against other approximate Bayesian inference methods. For regression, we conduct experiments on the 1D problem defined in (5) and visualized in Figure 2. We use the Gaussian model (2) with two separate feed-forward neural networks outputting $\mu_\theta(x)$ and $\log \sigma_\theta^2(x)$. We evaluate the methods by quantitatively measuring how well the obtained predictive distributions approximate that of the "gold standard" HMC [39] with $M = 1\,000$ samples and prior $p(\theta) = \mathcal{N}(0, I_P)$. We thus consider the predictive distribution visualized in Figure 2d ground truth, and take as our metric the KL divergence $D_{\mathrm{KL}}(p \parallel p_{\mathrm{HMC}})$ with respect to this target distribution $p_{\mathrm{HMC}}$. For classification, we conduct experiments on the binary classification problem in Figure 3. The true data generator is visualized in Figure 3a, where red and blue represents the two classes. The
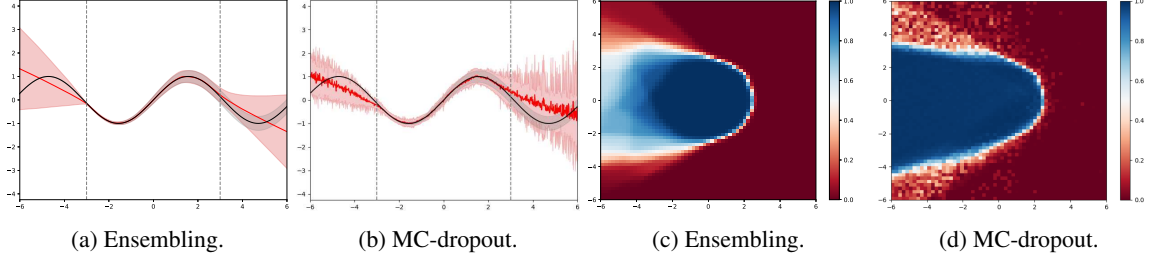
(a) Ensembling.     (b) MC-dropout.     (c) Ensembling.     (d) MC-dropout.

Figure 4: Illustrative toy problems - example predictive distributions for ensembling and MC-dropout with $M = 16$ samples.
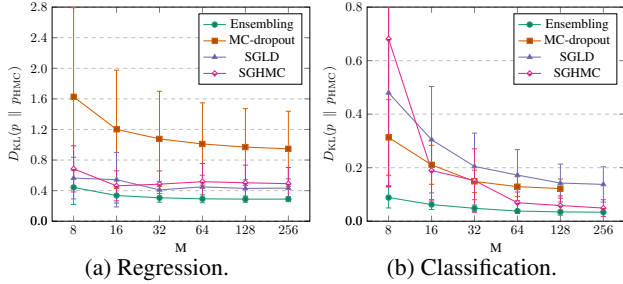


(a) Regression.     (b) Classification.

Figure 5: Illustrative toy problems - quantitative results. The plots show the KL divergence ($\downarrow$) between the predictive distribution estimated by each method and the HMC "ground truth", for different number of samples $M$.

training dataset contains 520 examples of each class, and is visualized in Figure 3b. We use the Categorical model (1) with a feed-forward neural network. As for regression, we quantitatively measure how well the obtained predictive distributions approximate that of HMC, which is visualized in Figure 3c. Further details are provided in Appendix B.

**Results** A comparison of ensembling, MC-dropout, SGLD and SGHMC in terms of $D_{KL}(p \parallel p_{HMC})$ is found in Figure 5. The Adam optimizer [26] is here used for both ensembling and MC-dropout. We observe that ensembling consistently outperforms the compared methods, and MC-dropout in particular. Even compared to SG-MCMC variants such as SGLD and SGHMC, ensembling thus provides a better approximation to the MCMC method HMC. This result is qualitatively supported by visualized predictive distributions found in Appendix B.5. Example predictive distributions for ensembling and MC-dropout with $M = 16$ are shown in Figure 4. We observe that ensembling provides reasonable approximations to HMC even for quite small values of $M$, especially compared to MC-dropout.

## 4.2. Depth Completion

In depth completion, we are given an image $x_{img} \in \mathbb{R}^{h \times w \times 3}$ from a forward-facing camera and an associated *sparse* depth map $x_{sparse} \in \mathbb{R}^{h \times w}$. Only non-zero pixels of $x_{sparse}$ correspond to LiDAR depth measurements, projected onto the image plane. The goal is to predict a dense depth map $y \in \mathbb{R}^{h \times w}$ of the scene. We utilize the KITTI

depth completion [15, 45] and Virtual KITTI [11] datasets. KITTI depth completion contains more than $80\,000$ images $x_{img}$, sparse depth maps $x_{sparse}$ and semi-dense target maps $y$. There are $1\,000$ selected validation examples, which we use for evaluation. Only about $4\%$ of the pixels in $x_{sparse}$ are non-zero and thus correspond to depth measurements. The semi-dense target maps are created by merging the Li-DAR scans from 11 consecutive frames into one, producing $y$ in which roughly $30\%$ of the pixels are non-zero. Virtual KITTI contains synthetic images $x_{img}$ and dense depth maps $x_{dense}$ extracted from 5 driving sequences in a virtual world. It contains $2\,126$ unique frames, of which there are 10 different versions corresponding to various simulated weather and lighting conditions. We take sequence 0002 as our validation set, leaving a total of $18\,930$ training examples. We create targets $y$ for training by setting all pixels in $x_{dense}$ corresponding to a depth $> 80$m to 0, and then also randomly sample $5\%$ of the remaining non-zero pixels uniformly to create $x_{sparse}$. We use the DNN model presented by Ma et al. [32]. The inputs $x_{img}$, $x_{sparse}$ are separately processed by initial convolutional layers, concatenated and fed to an encoder-decoder architecture based on ResNet34 [20]. We employ the Gaussian model (2) by duplicating the final layer, outputting both $\mu \in \mathbb{R}^{h \times w}$ and $\log \sigma^2 \in \mathbb{R}^{h \times w}$ instead of only the predicted depth map $\hat{y} \in \mathbb{R}^{h \times w}$. We also employ the same basic training procedure as Ma et al. [32] to train all our models, see Appendix C.1 for details. For the MC-dropout comparison, we take inspiration from Kendall et al. [24] and place a dropout layer with drop probability $p = 0.5$ after the three last encoder blocks and the four first decoder blocks.

**Evaluation Metrics** We evaluate the methods in terms of the *Area Under the Sparsification Error curve (AUSE)* metric, as introduced by Ilg et al. [23]. AUSE is a *relative* measure of the uncertainty estimation quality, comparing the ordering of predictions induced by the estimated predictive uncertainty (sorted from least to most uncertain) with the "oracle" ordering in terms of the true prediction error. The metric thus reveals how well the estimated uncertainty can be used to sort predictions from worst (large true prediction error) to best (small prediction error). We compute AUSE in terms of *Root Mean Squared Error (RMSE)* and based on all pixels in the entire evaluation dataset. A
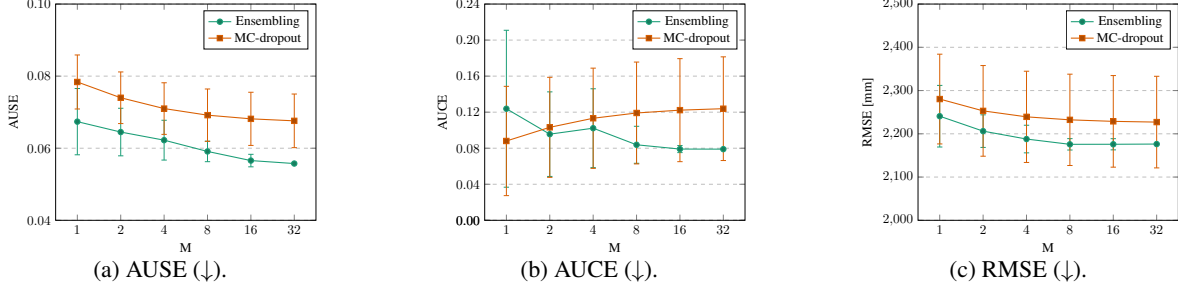
Figure 6: Depth completion - quantitative results. The plots show a comparison of ensembling and MC-dropout in terms of AUSE, AUCE and RMSE on the KITTI depth completion validation dataset, for different number of samples $M$.
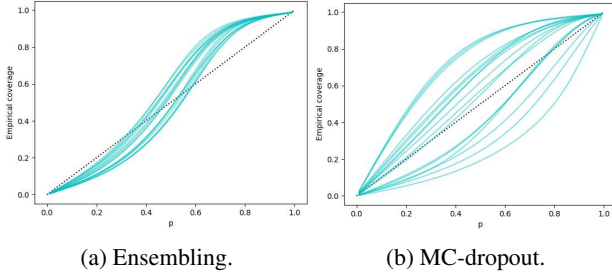


Figure 7: Depth completion - condensed calibration plots for ensembling and MC-dropout with $M = 16$.

perfect AUSE score can however be achieved even if the true predictive uncertainty is consistently underestimated. As an *absolute* measure of uncertainty estimation quality, we therefore also evaluate the methods in terms of calibration [5, 46]. In classification, the *Expected Calibration Error (ECE)* [17, 37] is a standard metric used to evaluate calibration. A well-calibrated model should then produce classification confidences which match the observed prediction accuracy, meaning that the model is not over-confident (outputting highly confident predictions which are incorrect), nor over-conservative. We here employ a metric that can be considered a natural generalization of ECE to the regression setting. Since our models output the mean $\mu \in \mathbb{R}$ and variance $\sigma^2 \in \mathbb{R}$ of a Gaussian distribution for each pixel, we can construct pixel-wise prediction intervals $\mu \pm \Phi^{-1}(\frac{p+1}{2})\sigma$ of confidence level $p \in ]0,1[$, where $\Phi$ is the CDF of the standard normal distribution. When computing the proportion of pixels for which the prediction interval covers the true target $y \in \mathbb{R}$, we expect this value, denoted $\hat{p}$, to equal $p \in ]0,1[$ for a perfectly calibrated model. We compute the absolute error with respect to perfect calibration, $|p - \hat{p}|$, for 100 values of $p \in ]0,1[$ and use the area under this curve as our metric, which we call *Area Under the Calibration Error curve (AUCE)*. Lastly, we also evaluate in terms of the standard RMSE metric.

**Results** A comparison of ensembling and MC-dropout in terms of AUSE, AUCE and RMSE on the KITTI depth completion validation dataset is found in Figure 6. We observe in Figure 6a that ensembling consistently outperforms MC-

dropout in terms of AUSE. However, the curves decrease as a function of $M$ in a similar manner. Sparsification plots and sparsification error curves are found in Appendix C.3. A ranking of the methods can be more readily conducted based on Figure 6b, where we observe a clearly improving trend as $M$ increases for ensembling, whereas MC-dropout gets progressively worse. This result is qualitatively supported by the calibration plots found in Appendix C.3 and Figure 7. Note that $M = 1$ corresponds to the baseline of only estimating aleatoric uncertainty.

### 4.3. Street-Scene Semantic Segmentation

In this task, we are given an image $x \in \mathbb{R}^{h \times w \times 3}$ from a forward-facing camera. The goal is to predict $y$ of size $h \times w$, in which each pixel is assigned to one of $C$ different class labels (road, sidewalk, car, etc.). We utilize the popular Cityscapes [10] and recent Synscapes [49] datasets. Cityscapes contains $5\,000$ finely annotated images, mainly collected in various German cities. The annotations includes 30 class labels, but only $C = 19$ are used in the training of models. Its validation set contains 500 examples, which we use for evaluation. Synscapes contains $25\,000$ synthetic images, all captured in virtual urban environments. To match the size of Cityscapes, we randomly select $2\,975$ of these for training and 500 for validation. The images are annotated with the same class labels as Cityscapes. We use the DeepLabv3 DNN model presented by Chen et al. [6]. The input image $x$ is processed by a ResNet101 [20], outputting a feature map of stride 8. The feature map is further processed by an ASPP module and a $1 \times 1$ convolutional layer, outputting logits at 1/8 of the original resolution. These are then upsampled to image resolution using bilinear interpolation. The conventional Categorical model (1) is thus used for each pixel. We base our implementation on the one by Yuan and Wang [51], and also follow the same basic training procedure, see Appendix D.1 for details. For reference, the model obtains an mIoU [29] of $76.04\%$ when trained on Cityscapes and evaluated on its validation set. For the MC-dropout comparison, we take inspiration from Mukhoti and Gal [36] and place a dropout layer with $p = 0.5$ after the four last ResNet blocks.
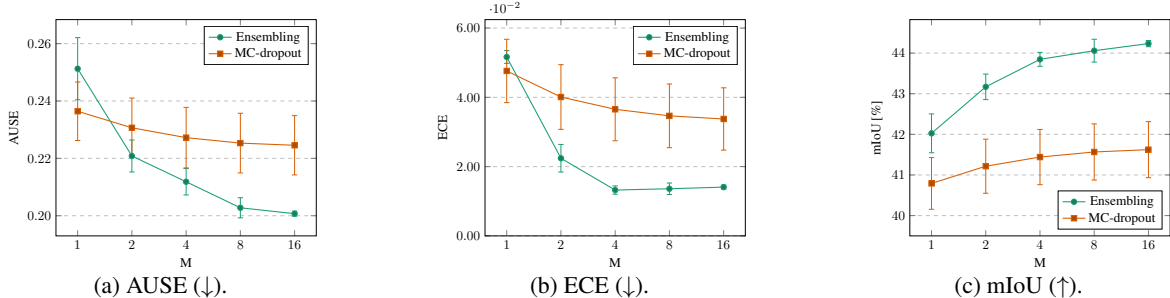
Figure 8: Street-scene semantic segmentation - quantitative results. The plots show a comparison of ensembling and MC-dropout in terms of AUSE, ECE and mIoU on the Cityscapes validation dataset, for different number of samples $M$.
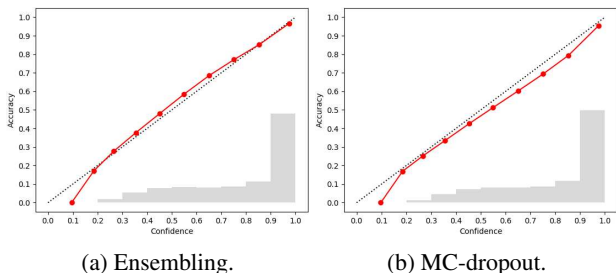


Figure 9: Street-scene semantic segmentation - example reliability diagrams for the two methods with $M = 16$.

**Evaluation Metrics** As for depth completion, we evaluate the methods in terms of the AUSE metric. In this classification setting, we compare the "oracle" ordering of predictions with the one induced by the predictive entropy. We compute AUSE in terms of Brier score and based on all pixels in the evaluation dataset. We also evaluate in terms of calibration by the ECE metric [17, 37]. All predictions are here partitioned into $L$ bins based on the maximum assigned confidence. For each bin, the difference between the average predicted confidence and the actual accuracy is then computed, and ECE is obtained as the weighted average of these differences. We use $L = 10$ bins of equal size.

**Results** A comparison of ensembling and MC-dropout in terms of AUSE, ECE and mIoU on the Cityscapes validation dataset is found in Figure 8. We observe that the metrics clearly improve as functions of $M$ for both ensembling and MC-dropout, demonstrating the importance of epistemic uncertainty estimation. The rate of improvement is generally greater for ensembling. For ECE, we observe in Figure 8b a drastic improvement for ensembling as $M$ is increased, followed by a distinct plateau. According to the condensed reliability diagrams in Appendix D.3, this corresponds to a transition from clear model over-confidence to slight over-conservatism. For MC-dropout, the corresponding diagrams suggest a stagnation while the model still is somewhat over-confident. Example reliability diagrams for $M = 16$ are shown in Figure 9, in which this over-confidence for MC-dropout can be observed. Note

that the relatively low mIoU scores reported in Figure 8c, obtained by models trained exclusively on Synscapes, are expected [49] and caused by the intentionally challenging domain gap between synthetic and real-world data.

## 5. Discussion & Conclusion

We proposed a comprehensive evaluation framework for *scalable* epistemic uncertainty estimation methods in deep learning. The proposed framework is specifically designed to test the robustness required in real-world computer vision applications. We applied our proposed framework and provided the first properly extensive and conclusive comparison of ensembling and MC-dropout, the results of which demonstrates that ensembling consistently provides more reliable and practically useful uncertainty estimates. We attribute the success of ensembling to its ability, due to the random initialization, to capture the important aspect of multi-modality present in the posterior distribution $p(\theta|\mathcal{D})$ of DNNs. MC-dropout has a large design-space compared to ensembling, and while careful tuning of MC-dropout potentially could close the performance gap on individual tasks, the simplicity and general applicability of ensembling must be considered key strengths. The main drawback of both methods is the computational cost at test time that grows linearly with $M$, limiting real-time applicability. Here, future work includes exploring the effect of model pruning techniques [50, 18] on predictive uncertainty quality. For ensembling, sharing early stages of the DNN among ensemble members is also an interesting future direction. A weakness of ensembling is the additional training required, which also scales linearly with $M$. The training of different ensemble members can however be performed in parallel, making it less of an issue in practice given appropriate computing infrastructure. In conclusion, our work suggests that ensembling should be considered the new go-to method for *scalable* epistemic uncertainty estimation.

# References

[1] Peter Auer, Mark Herbster, and Manfred K Warmuth. Exponentially many local minima for single neurons. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 316–322, 1996. 5

[2] David Barber and Christopher M Bishop. Ensemble learning in Bayesian neural networks. *Nato ASI Series F Computer and Systems Sciences*, 168:215–238, 1998. 5

[3] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 2018. 11

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning (ICML)*, pages 1613–1622, 2015. 2, 5

[5] Jochen Bröcker. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society*, 135(643):1512–1519, 2009. 7

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1, 7

[7] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning (ICML)*, pages 1683–1691, 2014. 2, 4

[8] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204, 2015. 5

[9] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4759–4770, 2018. 1, 2, 3

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. 1, 2, 7

[11] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 6

[12] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016. 1, 2

[13] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016. 5

[14] Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3369–3378, 2018. 1

[15] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 2013. 2, 6

[16] Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2348–2356, 2011. 5

[17] 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 (ICML)*, pages 1321–1330, 2017. 1, 7, 8

[18] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016. 8

[19] W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. 4

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6, 7

[21] José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning (ICML)*, pages 1861–1869, 2015. 2

[22] Geoffrey Hinton and Drew Van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Proceedings of the 6th Annual ACM Conference on Computational Learning Theory (COLT)*, 1993. 2, 5

[23] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Bro. Uncertainty estimates and multi-hypotheses networks for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 652–667, 2018. 1, 2, 3, 6

[24] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017. 2, 5, 6

[25] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5574–5584, 2017. 1, 2, 3

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[27] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6402–6413, 2017. 1, 2, 3, 5

[28] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 7

[30] Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix Gaussian posteriors. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1708–1716, 2016. 5

[31] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational Bayesian neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 2218–2227, 2017. 5

[32] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019. 1, 6, 16

[33] Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2917–2925, 2015. 4

[34] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7047–7058, 2018. 1

[35] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953. 4

[36] Jishnu Mukhoti and Yarin Gal. Evaluating Bayesian deep learning methods for semantic segmentation. *arXiv preprint arXiv:1811.12709*, 2018. 2, 5, 7

[37] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. 7, 8

[38] Radford M Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995. 2, 4

[39] Radford M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2:113–162, 2011. 4, 5

[40] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NeurIPS - Autodiff Workshop*, 2017. 5

[41] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3179–3189, 2018. 1

[42] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointR-CNN: 3D object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3

[43] 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 (NeurIPS)*, pages 13969–13980, 2019. 2

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 5

[45] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant CNNs. In *International Conference on 3D Vision (3DV)*, 2017. 1, 2, 6

[46] Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas B Schön. Evaluating model calibration in classification. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019. 7

[47] Hao Wang, SHI Xingjian, and Dit-Yan Yeung. Natural-parameter networks: A class of probabilistic neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 118–126, 2016. 2

[48] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning (ICML)*, pages 681–688, 2011. 2, 4

[49] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *arXiv preprint arXiv:1810.08705*, 2018. 2, 7, 8

[50] Tien-Ju Yang, Yu-Hsin Chen, and Vivienne Sze. Designing energy-efficient convolutional neural networks using energy-aware pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5687–5695, 2017. 8

[51] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 1, 7, 23

[52] Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 5847–5856, 2018. 5

[53] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. *arXiv preprint arXiv:1902.03932*, 2019. 4