



Wasserstein dropout

Joachim Sicking¹  · Maram Akila¹ · Maximilian Pintz¹ · Tim Wirtz¹ · Stefan Wrobel^{1,2,3} · Asja Fischer⁴

Received: 2 December 2021 / Revised: 15 June 2022 / Accepted: 29 July 2022 /
Published online: 8 September 2022
© The Author(s) 2022

Abstract

Despite of its importance for safe machine learning, uncertainty quantification for neural networks is far from being solved. State-of-the-art approaches to estimate neural uncertainties are often hybrid, combining parametric models with explicit or implicit (dropout-based) ensembling. We take another pathway and propose a novel approach to uncertainty quantification for regression tasks, *Wasserstein dropout*, that is purely non-parametric. Technically, it captures *aleatoric* uncertainty by means of dropout-based sub-network distributions. This is accomplished by a new objective which minimizes the Wasserstein distance between the label distribution and the model distribution. An extensive empirical analysis shows that Wasserstein dropout outperforms state-of-the-art methods, on vanilla test data as well as under distributional shift in terms of producing more accurate and stable uncertainty estimates.

Keywords Safe machine learning · Regression neural networks · Uncertainty estimation · Aleatoric uncertainty · Dropout · Object detection

1 Introduction

Having attracted great attention in both academia and digital economy, deep neural networks (DNNs, Goodfellow et al. (2016)) are about to become vital components of safety-critical applications. Examples are autonomous driving (Bojarski et al., 2016; Pomerleau, 1988) or medical diagnostics (Liu et al., 2014), where prediction errors potentially put humans at risk. These systems require methods that are robust not only under lab conditions (e.g. i.i.d. data sampling), but also under continuous domain shifts. Besides shifts in the data, the data distribution itself poses further challenges. Critical situations are (fortunately) rare and thus strongly under-represented in datasets. Despite their rareness, these critical situations have a significant impact on the safety

Editors: Dana Drachler Cohen, Javier Garcia, Mohammad Ghavamzadeh, Marek Petrik, Philip S. Thomas.

✉ Joachim Sicking
joachim.sicking@iaais.fraunhofer.de

Extended author information available on the last page of the article

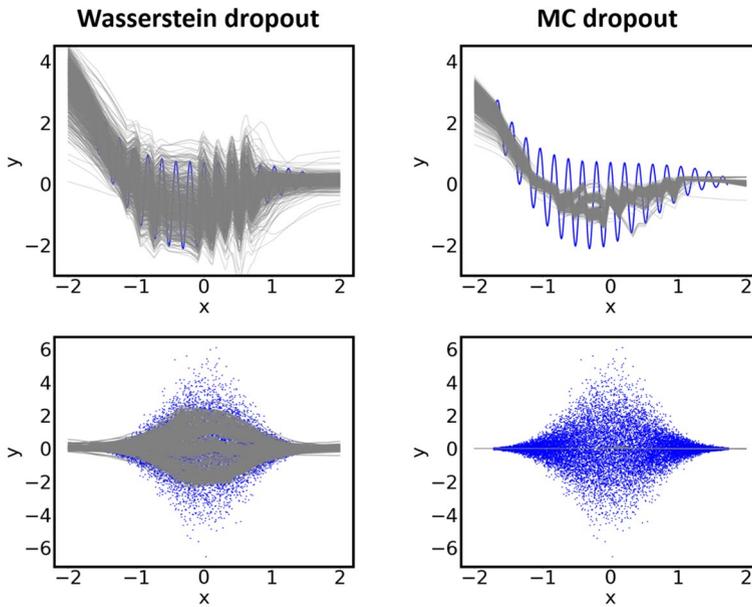


Fig. 1 Wasserstein dropout (left column) employs sub-networks to model *aleatoric uncertainty*, i.e. the heterogeneous noise of (in this case, toy) datasets is reflected by the sub-network distributions of the trained models. This is in contrast to other uncertainty methods like MC dropout (right column) that use sub-network distributions to model *epistemic uncertainty*. This type of uncertainty is small after training a model on the densely sampled toy datasets and consequently MC dropout’s sub-network distributions are significantly more narrow compared to Wasserstein dropout. The ground truth data is shown in blue. Each gray line represents the outputs of one of 500 random sub-networks that are obtained by applying dropout-based sampling to the trained full network. For details on the data sets (‘toy-hf’, ‘toy-noise’), the neural architecture and the uncertainty methods please refer to Sect. 4 and references therein (Color figure online)

of operations. This calls for comprehensive self-assessment capabilities of DNNs and recent uncertainty mechanisms can be seen as a step in that direction.

While a variety of approaches to model uncertainty of DNN predictions in regression tasks has been established, stable uncertainty quantification is still an open problem. Widely used techniques like Kendall and Gal (2017) and Lakshminarayanan et al. (2017) combine parametric and non-parametric (ensembling-based) mechanisms to account for aleatoric uncertainty (data noise) and epistemic uncertainty (model weight uncertainty). The employed parametric mechanisms represent uncertainty estimates by dedicated network output variables, which are often interpreted as variance parameters of Gaussian distributions. These modeling techniques are sometimes also referred to as “direct modeling” (Feng et al., 2020).

In this work, we take a different approach and propose to model (aleatoric) uncertainty in DNNs in a novel, *fully non-parametric* way. We introduce *Wasserstein dropout* (*W-dropout*) that is designed to capture heteroscedastic (i.e. input-dependent) data noise by means of its sub-network distribution (see Fig. 1). It builds on the idea of matching the network output distribution, resulting from randomly dropping neurons, to the (factual or implicit) data distribution by minimizing the Wasserstein distance.

In detail, we contribute

- by deriving a novel and surprisingly simple Wasserstein-based learning objective for sub-networks that simultaneously optimizes task performance and uncertainty quality,
- by conducting an extensive empirical evaluation where W-dropout outperforms state-of-the-art uncertainty techniques w.r.t. various benchmark metrics, not only in-data but also under data shifts,
- and by introducing two novel uncertainty measures: a non-saturating calibration score and a measure for distributional tails that allows to analyze worst-case scenarios w.r.t. uncertainty quality.

The remainder of the paper is organized as follows: first, we present related work on uncertainty estimation in neural networks in Sect. 2. Next, Wasserstein dropout is introduced in Sect. 3. We study the uncertainties induced by Wasserstein dropout on various datasets in Sect. 4, paying special attention to safety-relevant evaluation schemes and metrics. An outlook in Sect. 5 concludes the paper.

2 Related work

Approaches to estimate predictive uncertainties can be broadly categorized into three groups: Bayesian approximations, ensemble approaches and parametric models.

Monte Carlo dropout (Gal and Ghahramani, 2016) is a prominent representative of the first group. It offers a Bayesian motivation, conceptual simplicity and scalability to application-size neural networks (NNs). This combination distinguishes MC dropout from other Bayesian neural network (BNN) approximations like in Blundell et al. (2015) and Ritter et al. (2020). A computationally more efficient version of MC dropout is one-layer or last-layer dropout (see e.g. Kendall and Gal (2017)). Alternatively, analytical moment propagation allows sampling-free MC-dropout inference at the price of additional approximations (e.g. Postels et al. (2019)). Further extensions of MC dropout target tuned performance by learning layer-specific drop rates using Concrete distributions (Gal et al., 2017), the integration of aleatoric uncertainty (Kendall and Gal, 2017), using a parametric approach and input-dependent dropout distributions (Fan et al., 2021). Note that dropout training is used—independent from an uncertainty context—for better model generalization (Srivastava et al., 2014). An alternative sampling-based approach is SWAG which constructs a Gaussian model weight distribution from the (last segment of the) training trajectory (Madrox et al., 2019).

Ensembles of neural networks, so-called deep ensembles (Lakshminarayanan et al., 2017), pose another popular approach to uncertainty modeling. Comparative studies of uncertainty mechanisms (Gustafsson et al., 2020; Snoek et al., 2019) highlight their advantageous uncertainty quality, making deep ensembles a state-of-the-art method. Fort et al. (2019) argue that ensembles capture the multi-modality of loss landscapes thus yielding potentially more diverse sets of solutions. When used in practice, these ensembles additionally include parametric uncertainty prediction for each of their members.

The third group are the before mentioned parametric modeling approaches that extend point estimations by adding a model output that is interpreted as variance or covariance (Heskes, 1996; Nix and Weigend, 1994). Typically, these approaches optimize a (Gaussian) negative log-likelihood (NLL, Nix and Weigend (1994)) and can be easily integrated with other approaches, for a review see Khosravi et al. (2011). A more recent representative of this group is, e.g., deep evidential regression (Amini et al., 2020), which places a

prior distribution on Gaussian parameters. A closely related model class is deep kernel learning. It approaches uncertainty modeling by combining NNs and Gaussian processes (GPs) in various ways, e.g., via an additional layer (Iwata and Ghahramani, 2017; Wilson et al., 2016), by using networks as GP kernels (Garnelo et al., 2018) or by matching NN residuals with a GP (Qiu et al., 2020).

In the context of object detection, the number of applicable uncertainty methods is limited by the complexity of the employed NNs. Nonetheless, several variants can be encountered. For instance, MC dropout, see e.g. Bhattacharyya et al. (2018) or Miller et al. (2018), or parametric approaches, see He et al. (2019), can scale to network sizes relevant for such applications. Hall et al. (2020) stress the importance of uncertainty estimation for bounding box detection.

The quality of uncertainties is typically evaluated using negative log-likelihood (Blei and Jordan, 2006; Gal and Ghahramani, 2016; Walker et al., 2016), expected calibration error (ECE, Naeini et al. (2015), Snoek et al. (2019)) and its variants and by considering correlations between uncertainty estimates and model errors, e.g., area under the sparsification error curve (AUSE, Ilg et al. (2018)) for image tasks. Moreover, it is common to study how useful uncertainty estimates are for solving auxiliary tasks such as out-of-distribution classification (Lakshminarayanan et al., 2017) or robustness w.r.t. adversarial attacks. An alternative approach is the investigation of qualitative uncertainty behaviors: Kendall and Gal (2017) check if the epistemic uncertainty decreases when increasing the training set and Wirges et al. (2019) study how the level of uncertainty depends on the distance of the object to a car for some 3D environment regression task.

3 Wasserstein dropout

Before we lay out our dropout-based approach to modeling *aleatoric* uncertainty, we analyze some central properties of Monte Carlo dropout. The latter also employs sub-networks, however, for the purpose of modeling *epistemic* uncertainty (Gal and Ghahramani, 2016): Given a neural network $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^m$ with parameters θ , MC dropout samples sub-networks $f_{\tilde{\theta}}$ by randomly dropping nodes from the main model f_θ yielding for each input x_i a distribution $\mathcal{D}_{\tilde{\theta}}(x_i)$ over network predictions. During MC dropout inference the final prediction is given by the mean of a sample from $\mathcal{D}_{\tilde{\theta}}(x_i)$, while the uncertainty associated with this prediction can be estimated as a sum of its variance and a constant uncertainty offset. The value of the latter term requires dataset-specific optimization. During MC dropout training, minimizing the objective function, e.g., the mean squared error (MSE), shifts all sub-network predictions towards the same training targets. For a more formal explanation of this behavior, and without loss of generality, let f_θ be a NN with one-dimensional output. The expected MSE for a training sample (x_i, y_i) under the model's output distribution $\mathcal{D}_{\tilde{\theta}}(x_i)$ is given by

$$E_{\tilde{\theta}}[(f_{\tilde{\theta}}(x_i) - y_i)^2] = (\mu_{\tilde{\theta}}(x_i) - y_i)^2 + \sigma_{\tilde{\theta}}^2(x_i), \quad (1)$$

with sub-network mean $\mu_{\tilde{\theta}}(x_i) = E_{\tilde{\theta}}[f_{\tilde{\theta}}(x_i)]$ and variance $\sigma_{\tilde{\theta}}^2(x_i) = E_{\tilde{\theta}}[f_{\tilde{\theta}}^2(x_i)] - E_{\tilde{\theta}}[f_{\tilde{\theta}}(x_i)]^2$. Therefore, training simultaneously minimizes the squared error between sub-network mean $\mu_{\tilde{\theta}}(x_i)$ and target y_i as well as the variance $\sigma_{\tilde{\theta}}^2(x_i)$.

As we, in contrast, seek to employ sub-networks to model *aleatoric* uncertainty, minimizing the variance over the sub-networks is not desirable for our purpose. Instead, we aim at explicitly fitting the sub-network variance $\sigma_{\tilde{\theta}}^2(x_i)$ to the input-dependent, i.e.

heteroscedastic, data variance. That is to say, we not only match the mean values as in (1) but seek to match the *entire* data distribution $\mathcal{D}_y(x_i)$ by means of the model’s output distribution $\mathcal{D}_{\hat{\theta}}(x_i)$. This output distribution is induced by applying Bernoulli dropout to all activations of the network. The matchings are technically realized by minimizing a distance measure between the two distributions $\mathcal{D}_{\hat{\theta}}(x_i)$ and $\mathcal{D}_y(x_i)$. While, in principle, various distances could be used, we, however, require two properties: i) the distance needs to be non-saturating, i.e. it needs to grow monotonously and unboundedly with the actual mismatch between the distributions. This is needed (or desirable) as for safety reasons we want to penalize strong mismatches. Additionally, ii) we require the distance to have a simple, closed form. This is needed for subsequent, bootstrap-inspired approximations (see below). The (squared) 2-Wasserstein distance (Villani, 2008) fulfills both of these properties¹ and is therefore employed in the following. Assuming that both distributions $\mathcal{D}_{\hat{\theta}}(x_i)$ and $\mathcal{D}_y(x_i)$ are Gaussian² then yields a compact analytical expression

$$\begin{aligned} \text{WS}_2^2(x_i) &= \text{WS}_2^2[\mathcal{D}_{\hat{\theta}}(x_i), \mathcal{D}_y(x_i)] \\ &= \text{WS}_2^2[\mathcal{N}(\mu_{\hat{\theta}}(x_i), \sigma_{\hat{\theta}}(x_i)), \mathcal{N}(\mu_y(x_i), \sigma_y(x_i))] \\ &= (\mu_{\hat{\theta}}(x_i) - \mu_y(x_i))^2 + (\sigma_{\hat{\theta}}(x_i) - \sigma_y(x_i))^2, \end{aligned} \tag{2}$$

with $\mu_{\hat{\theta}}(x_i) = E_{\hat{\theta}}[f_{\hat{\theta}}(x_i)]$ and $\sigma_{\hat{\theta}}^2(x_i) = E_{\hat{\theta}}[(f_{\hat{\theta}}(x_i) - E_{\hat{\theta}}[f_{\hat{\theta}}(x_i)])^2]$, and μ_y, σ_y defined analogously w.r.t. the data distribution.

In practice however, (2) cannot be readily used as the distribution of y given x_i is typically not accessible. Instead, for a given, fixed value of x_i from the training set only a single value of y_i is known. Therefore, we take y_i as a (rough) one-sample approximation of the mean $\mu_y(x_i)$ resulting in $\mu_y(x_i) \approx y_i$ and $\sigma_y^2(x_i) \approx E_y[(y - y_i)^2]$. However, $\sigma_y^2(x_i)$ cannot be inferred from a single sample. Inspired by parametric bootstrapping (Dekking et al., 2005; Hastie et al., 2009), we therefore approximate the empirical data variance (for a given mean value y_i and input x_i) with samples from our model, i.e., we approximate $E_y[(y - y_i)^2]$ by

$$E_{\hat{\theta}}[(f_{\hat{\theta}}(x_i) - y_i)^2] = (\mu_{\hat{\theta}}(x_i) - y_i)^2 + \sigma_{\hat{\theta}}^2(x_i). \tag{3}$$

Inserting our approximations $\mu_y(x_i) \approx y_i$ and $\sigma_y(x_i) \approx (\mu_{\hat{\theta}}(x_i) - y_i)^2 + \sigma_{\hat{\theta}}^2(x_i)$ into (2) yields the *Wasserstein dropout loss* (W-dropout) for a data point (x_i, y_i) from the training distribution:

$$\text{WS}_2^2(x_i) \approx (\mu_{\hat{\theta}}(x_i) - y_i)^2 + \left[\sqrt{\sigma_{\hat{\theta}}^2(x_i)} - \sqrt{(\mu_{\hat{\theta}}(x_i) - y_i)^2 + \sigma_{\hat{\theta}}^2(x_i)} \right]^2. \tag{4}$$

Considering a mini-batch of size M instead of a single data point, we arrive at the optimization objective $\text{WS}_{batch}^2 = \frac{1}{M} \sum_{i=1}^M \text{WS}_2^2(x_i)$. In practice, $\mu_{\hat{\theta}}(x_i) \approx \frac{1}{L} \sum_{l=1}^L f_{\hat{\theta}_l}(x_i)$ and $\sigma_{\hat{\theta}}^2(x_i) \approx \frac{1}{L} \sum_{l=1}^L f_{\hat{\theta}_l}^2(x_i) - (\frac{1}{L} \sum_{l=1}^L f_{\hat{\theta}_l}(x_i))^2$ are approximated by empirical estimators using a

¹ This is in contrast to other widely used metrics. The KS statistic, for example, is saturating and therefore violates the first requirement whereas the KL divergence possesses a more involved structure that violates the second requirement.

² An assumption shared by, e.g., the NLL optimization or the ECE. While different distributions, for example exponentially decaying or mixtures, could be used in principle, we restrict the scope here to this standard Gaussian case.

sample size L . In contrast to MC dropout we require thereby L stochastic forward passes per data point during training (instead of one), while at inference procedures are exactly the same.

Besides the regression tasks considered here our approach could be useful for other objectives which use or benefit from an underlying distribution, e.g., Dirichlet distributions to quantify uncertainty in classification, as discussed in the conclusion.

4 Experiments

We first outline the scope of our empirical study in Sect. 4.1 and begin with experiments on illustrative and visualizable toy datasets in Sect. 4.2. Next, we benchmark W-dropout on various 1D datasets (mostly from the UCI machine learning repository (Duam and Graff, 2017)) in Sect. 4.3, considering both in-data and distribution-shift scenarios. In Sect. 4.4, W-dropout is applied to the complex task of object detection using the compact Squeeze-Det architecture (Wu et al., 2017).

4.1 Benchmark approaches and evaluation measures

In this subsection, we present the considered benchmark approaches (first paragraph) and evaluation measures for uncertainty modeling. Aside established measures (second paragraph), we propose two novel uncertainty scores: an unbounded calibration measure and an uncertainty tail measure for the analysis of worst-case scenarios w.r.t. uncertainty quality (third and fourth paragraph). A brief overview of the technical setup (last paragraph) concludes the subsection.

Benchmark approaches

We compare W-dropout networks to archetypes of uncertainty modeling, namely approximate Bayesian techniques, parametric uncertainty, and ensembling approaches. From the first group, we pick MC dropout (abbreviated as **MC**, Gal and Ghahramani (2016)) and Concrete dropout (**CON-MC**, Gal et al. (2017)). The variance of MC is given as the sample variance plus a dataset-specific regularization term. The networks employing these methods do not exhibit parametric uncertainty outputs (see below). We additionally consider SWA-Gaussian (**SWAG**, Maddox et al. (2019)), which samples from a Gaussian model weight distribution that is constructed based on model parameter configurations along the (final segment of the) training trajectory. While these sampling-based approaches integrate uncertainty estimation into the structure of the entire network, *parametric* approaches model the variance directly as the output of the neural network (Nix and Weigend, 1994). Such networks typically output mean and variance of a Gaussian distribution (μ, σ^2) and are trained by likelihood maximization. This approach is denoted as **PU** for parametric uncertainty. Ensembles of PU-networks (Lakshminarayanan et al., 2017), referred to as deep ensembles, pose a widely used state-of-the-art method for uncertainty estimation (Snoek et al., 2019). Deep evidential regression (**PU-EV**, Amini et al. (2020)) extends this parametric approach and considers prior distributions over μ and σ . Kendall and Gal (2017) consider drawing multiple dropout samples from a parametric uncertainty model and aggregating multiple predictions for μ and σ . We denote this approach **PU-MC**. Moreover, we consider ensembles of non-parametric standard networks. We refer to the latter ones as **DEs** while we call those using additionally PU-based uncertainty **PU-DEs**. All considered types of networks provide estimates (μ_i, σ_i) where σ_i is obtained either as

direct network output (PU, PU-EV), by sampling (MC, CON-MC, SWAG, W-dropout) or as an ensemble aggregate (DE, PU-DE). For PU-MC, a combination of parametric output and sampling is employed. Throughout this section, we subsume PU, PU-EV, PU-DE and PU-MC as “parametric methods”.

Standard evaluation measures

In all experiments we evaluate both regression performance and uncertainty quality. Regression performance is quantified by the root-mean-square error $\sqrt{1/N \sum_i (\mu_i - y_i)^2}$ (RMSE, Bishop (2006)). Another established metric in the uncertainty community is the (Gaussian) negative log-likelihood (NLL), $1/N \sum_i (\log \sigma_i + (\mu_i - y_i)^2 / (2\sigma_i^2) + c)$, a hybrid between performance and uncertainty measure (Gneiting and Raftery, 2007), see Appendix C.2 for a discussion. Throughout the paper, we ignore the constant $c = \log \sqrt{2\pi}$ of the NLL. The expected calibration error (ECE, Kuleshov et al. (2018)) in contrast is not biased towards well-performing models and in that sense a pure uncertainty measure. It reads $ECE = \sum_{j=1}^B |\tilde{p}_j - 1/B|$ for B equally spaced bins in quantile space and $\tilde{p}_j = |\{r_i | q_j \leq \tilde{q}(r_i) < q_{j+1}\}|/N$ the empirical frequency of data points falling into such a bin. Their normalized prediction residuals r_i are defined as $r_i = (\mu_i - y_i)/\sigma_i$. Further, \tilde{q} is the cdf of the standard normal distribution $\mathcal{N}(0, 1)$ and $[q_j, q_{j+1})$ are equally spaced intervals on $[0, 1]$, i.e., $q_j = (j - 1)/B$.

An unbounded uncertainty calibration measure

A desirable property for uncertainty measures is a signal that grows (preferentially linearly) with the misalignment between predicted and ideal uncertainty estimates, especially when handling strongly deviating uncertainty estimates. As the Wasserstein metric fulfils this property, we not only use it for model optimization but propose to consider the 1-Wasserstein distance of normalized prediction residuals (WS) as a complementary uncertainty evaluation measure. It is generally applicable and by no means restricted to W-dropout networks. In detail, the 1-Wasserstein distance (Villani, 2008), also known as earth mover’s distance (Rubner et al., 1998), is a transport-based measure, denoted by d_{WS} , between two probability densities, with Wasserstein GANs (Arjovsky et al., 2017) as its most prominent application in machine learning. In the context of uncertainty estimation, we use the Wasserstein distance to measure deviations of uncertainty estimates $\{r_i\}_i$ from ideal (Gaussian)³ calibration that is given if $y_i \sim \mathcal{N}(\mu_i, \sigma_i)$ with accompanying normalized residuals of $r_i \sim \mathcal{N}(0, 1)$, i.e. we calculate $d_{WS}(\{r_i\}_i, \mathcal{N}(0, 1))$. As ECE, this is a pure uncertainty measure. However, it is not based on quantiles but directly on normalized residuals and can therefore resolve deviations on all scales. For example, two strongly ill-calibrated uncertainties would result in (almost) identical ECE values while WS would resolve this difference in magnitude. Let us compare ECE and WS more systematically: we consider normal distributions $\mathcal{N}(\mu, 1)$ and $\mathcal{N}(0, \sigma)$ (see Fig. 2) that are shifted (top left panel, dark blue) and squeezed/stretched (bottom left panel, dark blue), respectively. Their deviations from the ideal/normalized residual distribution (the standard normal, red) are measured in terms of both ECE (r.h.s., blue) and WS (r.h.s., orange). For large values of $|\mu|$ and σ , ECE is bounded while WS increases linearly showing the better sensitivity of the latter towards strong deviations. For small values, $\sigma \rightarrow 0$, ECE takes its maximum value, WS a value of 1. In Fig. 3, we visualize these value pairs (WS(σ), ECE(σ)) (gray lines), i.e. σ serves as curve parameter. The upper ‘branch’ corresponds to $0 < \sigma < 1$, the lower ‘branch’ to $\sigma > 1$. For comparison, the pairs (WS, ECE) of various networks trained on standard regression

³ As stated before, Gaussianity, while not always given, is a standard assumption for uncertainty modeling and typically used in ECE and NLL.

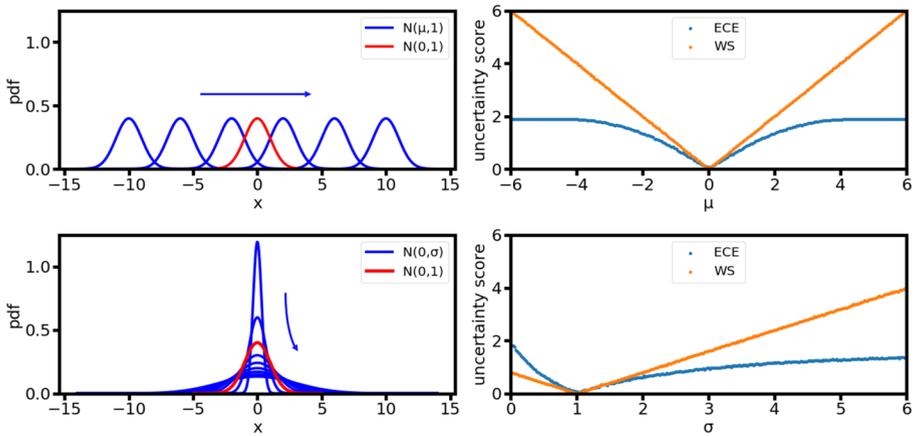


Fig. 2 Comparison of the proposed Wasserstein-based measure (WS) and the expected calibration error (ECE). We measure the deviation between a standard normal distribution $\mathcal{N}(0, 1)$ (lhs, red) and shifted normal distributions $\mathcal{N}(\mu, 1)$ (top left, dark blue) and squeezed/stretched normal distributions $\mathcal{N}(0, \sigma)$ (bottom left, dark blue), respectively. The resulting ECE values (orange) and WS values (blue) on the rhs emphasize the higher sensitivity of WS in case of large distributional differences. For details on ECE and WS, see text (Color figure online)

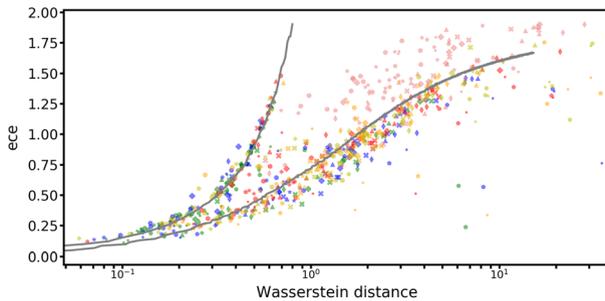


Fig. 3 Dependency between the Wasserstein-based measure and the expected calibration error for Gaussian toy data (gray curves) and for 1D standard datasets (point cloud, see Sect. 4.3 for details). The toy curves are obtained by plotting $(WS(\sigma), ECE(\sigma))$ from Fig. 2 (bottom right). For 1D standard datasets, uncertainty methods are encoded via plot markers, data splits via color. Datasets are not encoded and cannot be distinguished (see Appendix C for more details). Each plot point corresponds to a cross-validated trained network (Color figure online)

datasets are visualized (see Sect. 4.3 for experimental details and results). They approximately follow the theoretical σ -curve, emphasizing that both under- and overestimating variance is of practical relevance. A given WS value allows, due to lacking saturation for underestimation, to distinguish these two cases more easily compared to ECE. While one might rightfully argue that the higher sensitivity of WS leads to a certain susceptibility to potential outliers, this can be addressed by regularizing the normalized residuals or by filtering extreme outliers.

A novel uncertainty tail measure

We furthermore introduce a measure for distributional tails that allows to analyze worst-case scenarios w.r.t. uncertainty quality, thus reflecting safety considerations. Such potentially critical worst-case scenarios are signified by the above mentioned outliers, where

the locally predicted uncertainty strongly underestimates the actual model error. A better understanding of uncertainty estimates in these scenarios might allow to determine lower bounds on operation quality of safety-critical systems. For this, we consider normalized residuals $r_i = (\mu_i - y_i)/\sigma_i$ based on the prediction estimates (μ_i, σ_i) for a given data point (x_i, y_i) . As stated, we restrict our analysis to uncertainty estimates that *underestimate* model errors, i.e., $|r_i| \gg 1$. These cases might be more harmful than overly large uncertainties, $|r_i| \ll 1$, that likely trigger a conservative system behavior. We quantify uncertainty quality for worst-case scenarios as follows: for a given (test) dataset, the absolute normalized residuals $\{|r_i|\}_i$ are calculated. We determine the 99% quantile $q_{0.99}$ of this set and calculate the mean value over all $|r_i| > q_{0.99}$, the so-called *expected tail loss at quantile 99%* (**ETL_{0.99}**, Rockafellar and Uryasev (2002)). The **ETL_{0.99}** thus measures the average uncertainty quality of the worst 1%.

Technical setup

For the first two parts we use almost identical setups of 2 hidden layers with ReLU activations, using 50 neurons per layer for the toy datasets and 100 for the 1D standard datasets. All dropout-based networks (MC, CON-MC, W-dropout) apply Bernoulli dropout to all hidden activations. For W-dropout networks, we sample $L = 5$ sub-networks in each optimization step, other values of L are considered in Appendix B. On the smaller toy datasets, we afford $L = 10$. For MC and W-dropout, the drop rate is set to $p = 0.1$ (see Appendix B for other values of p). The drop rate of CON-MC in contrast is learned during training and (mostly) takes values between $p = 0.2$ and $p = 0.5$. For ensemble methods (DE, PU-DE) we employ 5 networks. All NNs are optimized using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001. Additionally, we apply standard normalization to the input and output features of all datasets to enable better comparability. The number of training epochs and cross validation runs depends on the dataset size. Further technical details on the networks, the training procedure, and the implementation of the uncertainty methods can be found in Appendix A.1. In using a least squares regression, we make the standard assumption that errors follow a Gaussian distribution. This is reflected in the (standard) definitions of above named measures, i.e., all uncertainty measures quantify the set of outputs $\{(\mu_i, \sigma_i)\}$ relative to a Gaussian distribution.

4.2 Toy datasets

To illustrate qualitative behaviors of the different uncertainty techniques, we consider two $\mathbb{R} \rightarrow \mathbb{R}$ toy datasets. This benchmark puts a special focus on the handling of aleatoric heteroscedastic uncertainty. The first dataset is Gaussian white noise with an x -dependent amplitude, see first row of Fig. 4. The second dataset is a polynomial overlaid with a high-frequency, amplitude-modulated sine, see fourth row of Fig. 4. The explicit equations for the toy datasets used here can be found in Appendix A.2.

While the uncertainty in the first dataset ('toy-noise') is clearly visible, it is less obvious for the fully deterministic second dataset ('toy-hf'). There is an effective uncertainty due to the insufficient expressivity of the model though, as the shallow networks employed are empirically not able to fit (all) fluctuations of 'toy-hf' (see fifth row of Fig. 4). One might (rightfully) argue that this is a sign of insufficient model capacity. But in more realistic, e.g., higher dimensional and sparser datasets the distinction between true noise and complex information becomes exceedingly difficult to make and regularization is actively used to suppress the modeling of (ideally) undesired fluctuations. As the Nyquist-Shannon sampling theorem states, with limited data

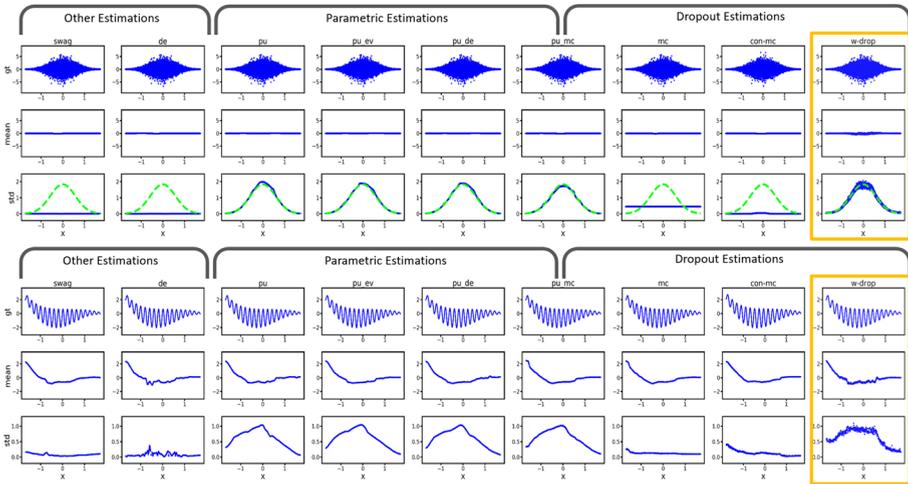


Fig. 4 Comparison of uncertainty approaches (columns) on two 1D toy datasets: a noisy one (top) and a high-frequency one (bottom). Test data ground truth (respective first row) is shown with mean estimates (resp. second row) and standard deviations (resp. third row). The light green dashed curve (third row) indicates the ground truth uncertainty. Similar uncertainty approaches (columns) are grouped together, W-dropout is highlighted by a yellow frame (Color figure online)

deterministic fluctuations above a cut-off frequency can no longer be resolved (Landau, 1967). They therefore become virtually indistinguishable from random noise.

The mean estimates of all uncertainty methods (second and fifth row in Fig. 4) look alike on both datasets. They approximate the noise mean and the polynomial, respectively. In the latter case, all methods rudimentarily fit some individual fluctuations. The variance estimation (third and sixth row in Fig. 4) in contrast reveals significant differences between the methods: MC dropout variants and other non-parametric ensembles are not capable of capturing heteroscedastic aleatoric uncertainty. This behavior of MC is expectable as it was primarily introduced to account for model uncertainty. The non-parametric DE is effectively optimized in a similar fashion. In contrast, NLL-optimized PU networks have a home-turf advantage on these datasets since the parametric variance is explicitly optimized to account for the present heteroscedastic aleatoric uncertainty. W-dropout is the only non-parametric approach that accounts for the presence of this kind of uncertainty. While the results look similar, the underlying mechanisms are fundamentally different. On the one hand explicit prediction of the uncertainty, on the other hand implicit modeling via distribution matching. Accompanying quantitative evaluations can be found in Table 7 in Appendix A.2. To collect further evidence that W-dropout approximates the true ground truth uncertainty σ_{true} appropriately, we fit it to ‘noisy line’ toy datasets in Appendix A.2. Both large and small σ_{true} values are correctly matched, indicating that W-dropout is not just adding an uncertainty offset but flexibly spreads/contracts its sub-networks as intended. In the following, we substantiate the corroborative results of W-dropout on toy data by an empirical study on 1D standard datasets and an application to a modern object detection network.

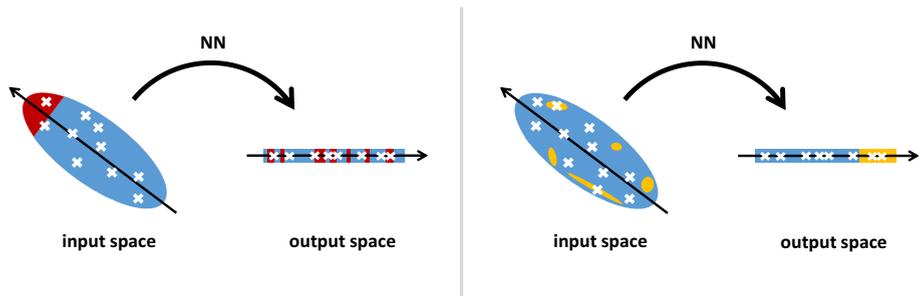


Fig. 5 Scheme of two non-i.i.d. splits: a PCA-based split in input space (left) and label-based split in output space (right). While datasets appear to be convex here, they are (most likely) not in reality

4.3 Standard 1D regression datasets

Next, we study standard regression datasets, extending the dataset selection in Gal and Ghahramani (2016) by adding four additional datasets: ‘diabetes’, ‘abalone’, ‘california’, and ‘superconduct’. Table 8 in Appendix A.3 provides details on dataset sources, pre-processing and basic statistics. Apart from train- and test-data results, we study regression performance and uncertainty quality *under data shift*. Such distributional changes and uncertainty quantification are closely linked since the latter ones are rudimentary “self-assessment” mechanisms that help to judge model reliability. These judgements gain importance for model inputs that are *structurally different* from the training data.

Data splits

Natural candidates for such non-i.i.d. splits are splits along the main directions of data in input and output space, respectively. Here, we consider 1D regression tasks. Therefore, output-based splits are simply done on a scalar label variable (see Fig. 5, right). We call such a split *label-based* (for a comparable split, see, e.g., Foong et al. (2019)). In input space, the first component of a principal component analysis (PCA) provides a natural direction (see Fig. 5, left). Projecting the data points onto this first PCA-axis yields the scalar values the *PCA-split* is based on. Note that these projections are only considered for data splitting, they are not used for model training. Splitting data along such a direction in input or output space in, e.g., 10 equally large chunks, creates 2 *outer* data chunks and 8 *inner* data chunks. Training a model on 9 of these chunks such that the remaining chunk for evaluation is an inner chunk is called data *interpolation*. If the remaining test chunk is an outer chunk, it is data *extrapolation*. For example, for labels running from 0 to 1, (label-based) extrapolation testing would consider only data with a label larger 0.9, while training would be performed on the smaller label values. We introduce this distinction as extrapolation is expected to be considerably more difficult than ‘bridging’ between feature combinations that were seen during training.

More general information on training and dataset-dependent modifications to the experimental setup are relegated to the technical Appendix A.1. The presented results are obtained as follows: for each of the 14 standard datasets, we calculate (for each uncertainty method) the per-dataset scores: RMSE, mean NLL, ECE and WS. To improve statistical significance, these scores are 5- or 10-fold cross-validated, i.e. averages across a respective number of folds. Given the (fold-averaged) per-dataset scores for all 14 standard datasets, we calculate and visualize their mean and median values as well as quantile intervals

Table 1 Regression performance (RMSE) and uncertainty quality (NLL, ECE, WS) of W-dropout and various uncertainty benchmarks. W-dropout yields the best uncertainty scores while providing a competitive RMSE value. Each number is the average across 14 standard 1D (test) datasets. The figures in this table correspond to the blue crosses in the second columns of Figs. 6 and 7, respectively. See text for further details

unc. method	RMSE (\downarrow)	NLL (\downarrow)	ECE (\downarrow)	WS (\downarrow)
SWAG	0.456	7.695	0.828	1.847
DE	0.407	6.184	0.796	1.628
PU	0.447	1.47×10^7	0.614	2.10×10^6
PU-EV	0.442	2.838	0.626	49.180
PU-DE	0.418	0.307	0.515	0.542
PU-MC	0.420	0.250	0.565	0.433
MC	0.412	0.190	0.788	0.643
CON-MC	0.436	1.513	0.669	0.964
W-Dropout	0.421	- 0.428	0.501	0.430

Bold values indicate the best value that was reached for a given evaluation measure

Table 2 Out-of-data analysis of W-dropout and various uncertainty benchmarks. Regression performance (RMSE) and uncertainty quality (NLL, ECE, WS) are displayed. As for in-domain test data, W-dropout outperforms the other uncertainty methods without sacrificing regression quality. Each number is obtained by two-fold averaging: firstly, across two types of out-of-data test sets (label-based and PCA-based splits) and secondly, across 14 standard 1D datasets. The figures in this table are based on the blue crosses in the last four columns of Figs. 6 and 7, respectively. See text for further details

unc. method	RMSE (\downarrow)	NLL (\downarrow)	ECE (\downarrow)	WS (\downarrow)
SWAG	0.641	27.602	1.138	3.818
DE	0.599	14.055	0.988	2.554
PU	0.632	1.50×10^7	0.968	1.55×10^6
PU-EV	0.611	6.290	0.941	44.447
PU-DE	0.594	1448.391	0.783	5.892
PU-MC	0.591	397.022	0.823	3.215
MC	0.589	2.330	0.923	1.207
CON-MC	0.621	13.820	0.963	2.109
W-Dropout	0.615	2.287	0.763	1.203

Bold values indicate the best value that was reached for a given evaluation measure

(see Figs. 6 and 7). For high-level summaries of the results on in-data and out-of-data test sets please refer to Table 1 and Table 2, respectively. While the mean values characterize the average behavior of the uncertainty methods, the displayed 75% quantiles indicate how well methods perform on the more challenging datasets. A small 75% quantile value thus hints at *consistent* stability of an uncertainty mechanism across a variety of tasks.

Regression quality

First, we consider regression performance, see Table 1 and the first two panels in the top row of Fig. 6. Averaging the RMSE values across the 14 datasets yields almost identical test results for all uncertainty methods (see Table 1). On training data (Fig. 6, first panel in top row) in contrast, we find the parametric methods to exhibit larger train data

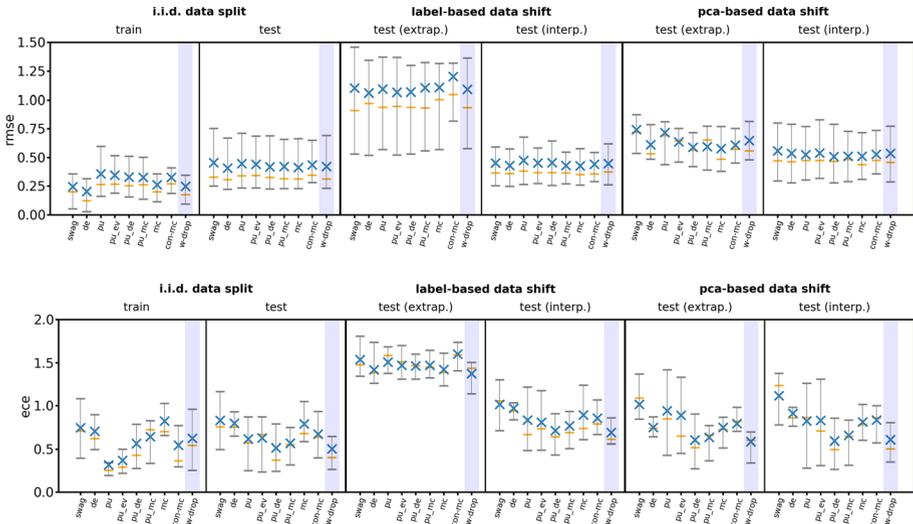


Fig. 6 Root-mean-square errors (RMSEs) (↓, top row) and expected calibration errors (ECEs) (↓, bottom row) of different uncertainty methods under i.i.d. conditions (first and second panel in each row) and under various kinds of data shift (third to sixth panel in each row, see text for details). W-dropout (light blue background) is compared to 8 benchmark approaches. Each blue cross is the mean over 14 1D regression datasets. Orange line markers indicate median values. The gray vertical bars reach from the 25% quantile (bottom horizontal line) to the 75% quantile (top horizontal line) (Color figure online)

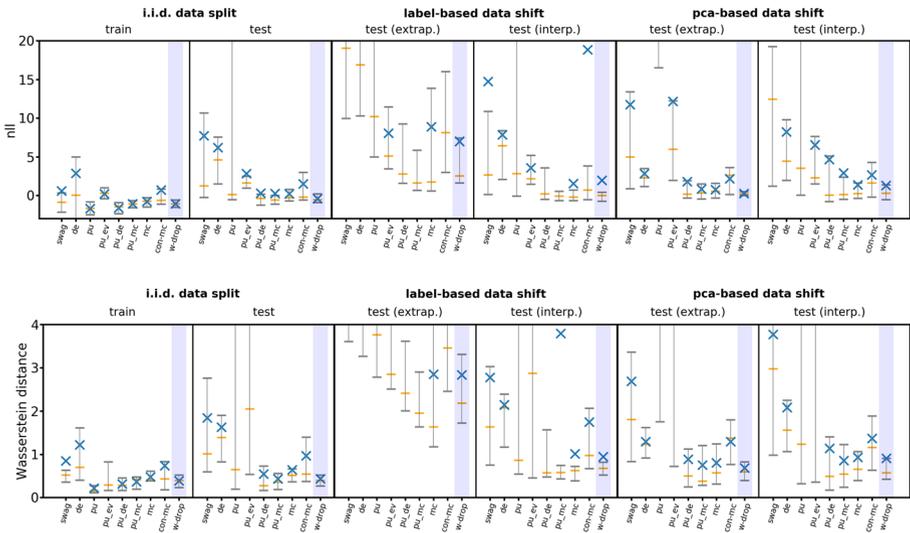


Fig. 7 Negative log-likelihoods (NLLs) (↓, top row) and Wasserstein distances (↓, bottom row) of different uncertainty methods under i.i.d. conditions (first and second panel in each row) and under various kinds of data shift (third to sixth panel in each row, see text for details). W-dropout (light blue background) is compared to 8 benchmark approaches. Each blue cross is the mean over ECE values from 14 standard regression datasets. Orange line markers indicate median values. The gray vertical bars reach from the 25% quantile (bottom horizontal line) to the 75% quantile (top horizontal line) (Color figure online)

RMSEs which could be due to NLL optimization favoring to adapt variance rather than mean. However, this regularizing NLL training comes along with a smaller generalization gap, leading to competitive test RMSEs (see Table 1 and the second panel in the top row of Fig. 6). W-dropout is on a par with the benchmark approaches, i.e. our optimization objective does not lead to degraded regression quality.⁴ Next, we investigate model performance under data shift, visualized in the third to sixth panel in the top row of Fig. 6. For interpolation setups (fourth and sixth panel), regression quality is comparable between all methods. As expected, performances under these data shifts are (slightly) worse compared to those on i.i.d. test sets. The more challenging extrapolation setups (third and fifth panel) amplify the deterioration in performance across all methods. Again, W-dropout yields competitive RMSE values (see also Table 2).

Expected calibration errors

Figure 6 (bottom row) provides average ECE values of the outlined uncertainty methods under i.i.d. conditions (first and second panel), under label-based data shifts (third and fourth panel) and under PCA-based data shifts (fifth and sixth panel). On training data, PU performs best, followed by PU-EV and all other methods. Interestingly, both SWAG and W-dropout show a relatively broad range of ECE values on the various training datasets. This could be interpreted as a form of over-estimation of the present uncertainty and for W-dropout this effect occurs on mostly smaller datasets with lower data variability. However, looking at the i.i.d. test results (Table 1 and second panel in the bottom row of Fig. 6) we find W-dropout to provide the lowest averaged ECE (Table 1), followed by the PU-based (implicit) ensembles of PU-DE and PU-MC. The calibration quality of W-dropout is moreover the most consistent one across the datasets as can be seen from its small 75% quantile value (Fig. 6, second panel in bottom row).

Looking at the stability w.r.t. data shift, i.e., extra- and interpolation based on label-split or PCA-split, again W-dropout reaches the smallest calibration errors (followed by PU-DE and PU-MC, see Table 2). Regarding the 75% quantiles, W-dropout consistently provides one of the best results on all out-of-data (OOD) test sets.

Negative log-likelihoods

For the unbounded NLL (see Table 1 and the top row of Fig. 7), the results are more widely distributed compared to the (bounded) ECE values. W-dropout reaches the smallest mean value on i.i.d. test sets, followed by MC and PU-MC (Table 1). The mean NLL value of PU is above the upper plot limit in Fig. 7 (second panel in the upper row) indicating a rather weak stability of this method. On PCA-interpolate and PCA-extrapolate test sets (Fig. 7, last two panels in the upper row), MC, PU-MC and W-dropout networks perform best. On label-interpolate and label-extrapolate test sets, only MC and W-dropout networks are in first place when considering average values, followed by PU-EV. The mean NLLs of many other approaches are above the upper plot limit. Averaging all these OOD results in Table 2, we find W-dropout to provide the overall smallest NLL values, narrowly followed by MC. Note that median results are not as widely spread and PU-DE, MC, PU-MC and W-dropout perform comparably well. These qualitative differences between mean and median behavior indicate that most methods perform poorly ‘once in a while’. A noteworthy observation as *stability across a variety of data shifts and datasets* can be seen as a crucial requirement for an uncertainty method. W-dropout models yield high stability in that sense w.r.t. NLL.

⁴ This observation does not only hold relative to the other uncertainty methods but, moreover, relative to a deterministic network, see Fig. 12 and surrounding discussions in Appendix A.3.

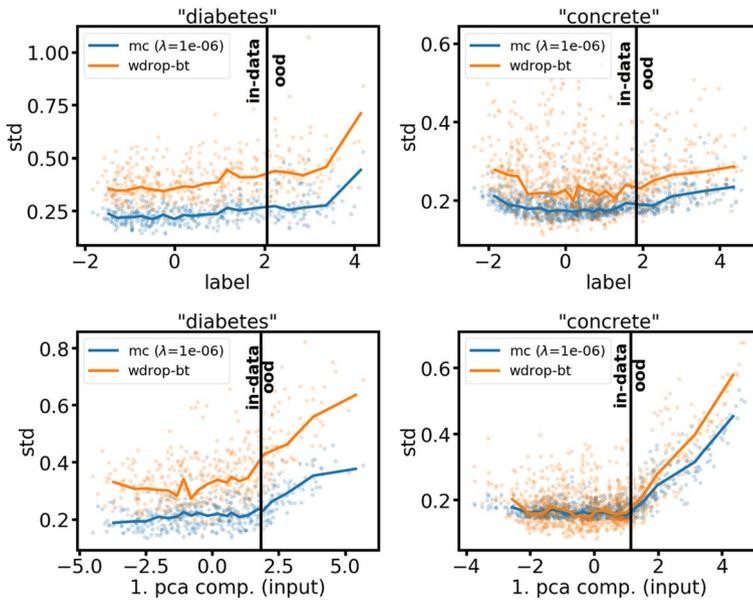


Fig. 8 Extrapolation behavior of W-dropout (orange) and MC dropout (blue). Two extrapolation “directions” (rows) and two datasets (columns) are considered. The vertical bar in each panel separates training data (left) from out-of-data (OOD, right). Scatter points show the predicted standard deviation for individual data points. The colored solid lines show averages over points in equally-sized bins and reflect the expected growth of epistemic uncertainty in the OOD-region. For details on the data splits and extrapolations please refer to Sect. 4.3 and Appendix A.3 (Color figure online)

Wasserstein distances

Studying Wasserstein distances, we again observe the smallest scores on test data for W-dropout, followed by PU-MC and PU-DE (see Table 1 and the second panel in the bottom row of Fig. 7). While PU provides the best WS value on training data, its generalization behavior is less stable: on test data, its mean and 75% quantile take high values beyond the plot range. Under data shift (Table 2 and third to sixth panel in bottom row of Fig. 7), W-dropout and MC are in the lead, CON-MC and DE follow on ranks three and four. On label-based data shifts, MC and W-dropout outperform all other methods by a significant margin when considering average values. As for NLL, we find the mean values for PU-DE and PU-MC to be significantly above their respective median values indicating again weaknesses w.r.t. the stability of parametric methods. Here as well, not only good *average* results, but also consistency over the datasets and splits, is a hallmark of Wasserstein dropout.

Epistemic uncertainty

Summarizing these evaluations on 1D regression datasets, we find W-dropout to yield better and more stable uncertainty estimates than the state-of-the-art methods of PU-DE and PU-MC. We moreover observe advantages for W-dropout under PCA- and label-based data shifts. These results suggest that W-dropout induces uncertainties which increase under data shift, i.e., it approximately models *epistemic uncertainty*. This conjecture is supported by Fig. 8 that visualizes the uncertainties of MC dropout (blue) and W-dropout (orange) for transitions from in-data to out-of-data. As expected, these shifts

Table 3 Study of worst-case scenarios for different uncertainty methods: W-dropout (W-Drop), PU-DE and PU-MC are compared to the ideal Gaussian case for i.i.d. and non-i.i.d. data splits. Uncertainty quality in these scenarios is quantified by the expected tail loss at the 99% quantile ($ETL_{0.99}$). Each mean and max value is taken over the ETLs of 110 models trained on 15 different datasets

Measure	Split	$\mathcal{N}(0, 1)$	W-Drop	PU-DE	PU-MC
mean $ETL_{0.99}$	i.i.d.	2.89	4.68	7.77	6.24
max $ETL_{0.99}$	i.i.d.	3.01	8.86	31.14	91.75
mean $ETL_{0.99}$	label	2.89	7.28	86.79	44.00
max $ETL_{0.99}$	label	3.01	93.55	2267.93	1224.27
mean $ETL_{0.99}$	pca	2.89	5.93	9.78	8.62
max $ETL_{0.99}$	pca	3.01	18.35	64.13	93.49

lead to increased (epistemic) uncertainty for MC dropout. This holds true for W-dropout that behaves highly similar under data shift indicating that it “inherits” this ability from MC dropout: both approaches match sub-networks to training data and these sub-networks “spread” when leaving the training data distribution. Since W-dropout models heteroscedastic, i.e. input-dependent, aleatoric uncertainty, we notice a higher variability of its uncertainties in Fig. 8 compared to the ones of MC dropout.

For further (visual) inspections of uncertainty quality, see the residual-uncertainty scatter plots in Appendix A.4. A reflection on NLL and comparisons of the different uncertainty measures on 1D regression datasets can be found in Appendix A.3.

Expected tail loss

For both toy and standard regression datasets, we calculate the expected tail loss at the 99% quantile ($ETL_{0.99}$) on test data. Doing this for all trained networks yields a total of 110 $ETL_{0.99}$ values per uncertainty method when including cross-validation. As a tail measure, the $ETL_{0.99}$ evaluates a specific aspect of the distribution of uncertainty estimates. Studying such a property is useful if the uncertainty estimate distribution as a whole is appropriate, as measured e.g. by the ECE. We thus restrict the $ETL_{0.99}$ analysis to the three methods that provide the best ECE values, namely PU-MC, PU-DE and W-dropout. The mean and maximum values of their $ETL_{0.99}$ ’s are reported in Table 3. While none of these methods gets close to the ideal $ETL_{0.99}$ ’s of the desired $\mathcal{N}(0, 1)$ Gaussian, W-dropout networks exhibit significantly less pronounced tails and therefore higher stability compared to PU-MC and PU-DE. This holds true over all considered test sets. Deviations from standard normal increase from the i.i.d. train-test split over the PCA-based train-test split to the label-based one. We attribute the lower stability of PU-DE to the nature of the PU networks that compose the ensemble, although their inherent instability (see Table 9 in Appendix A.3) is largely suppressed by ensembling. Considering the tail of the distribution of the prediction residuals $|r_i|$, however, reveals that regularization of PU by ensembling might not work in every single case. It is then unlikely that larger ensemble are able to fully cure this instability issue. Regularizing PU by applying dropout (PU-MC) leads to only mild improvement. W-dropout networks in contrast encode uncertainty into the structure of the entire network thus yielding improved stability compared to parametric approaches. Further analysis shows that the large normalized residuals $r_i = (\mu_i - y_i)/\sigma_i$, which cause the large $ETL_{0.99}$

values, correspond (on average) to large absolute errors ($\mu_i - y_i$).⁵ This underpins the practical relevance of the ETL analysis, as large absolute errors are more harmful than small ones in many contexts, e.g. when detecting traffic participants.

Dependencies between uncertainty measures

All uncertainty-related measures (NLL, ECE, WS, ETL) relate predicted uncertainties to actually occurring model residuals. Each of them putting emphasize on different aspects of the considered samples: NLL is biased towards well-performing models, ECE measures deviations within quantile ranges, Wasserstein distance resolves distances between normalized residuals and ETL focuses on distribution tails. The empirically observed dependencies between WS and ECE are visualized in Fig. 3. Additionally to WS and ECE, we consider Kolmogorov–Smirnov (KS) distances (Stephens, 1974) on normalized residuals in Fig. 21 in Appendix C.

While all these scores are expectably correlated, noteworthy deviations from ideal correlation occur. Therefore, we advocate for uncertainty evaluations based on various measures to avoid overfitting to a specific formalization of uncertainty. The top panel of Fig. 21 reflects the higher sensitivity of the Wasserstein distance compared to ECE: we observe two “slopes”, the first one corresponds to models that overestimate uncertainties, i.e., $\sigma_{\hat{\theta}} > |\mu_{\hat{\theta}} - y_i|$ on average. In these scenarios, WS is typically below 1 as 1 would be the WS distance between a delta distribution at zero (corresponding to $\sigma_{\hat{\theta}} \rightarrow \infty$) and the expected $\mathcal{N}(0, 1)$ Gaussian. The second “slope” contains models that underestimate uncertainties, i.e., $\sigma_{\hat{\theta}} < |\mu_{\hat{\theta}} - y_i|$. WS is not bounded in these scenarios and is thus—unlike ECE—able to resolve differences between any two uncertainty estimators.

4.4 Application to object regression

After studying toy and standard regression datasets, we turn towards the challenging task of object detection (OD), namely the SqueezeDet model (Wu et al., 2017), a fully convolutional neural network. First, we adopt the W-dropout objective to SqueezeDet (see the following paragraph). Next, we introduce the six considered OD datasets and sketch central technical aspects of training and inference. Since OD networks are often employed in open-world applications (like autonomous vehicles or drones), they likely encounter various types of concept shifts during operations. In such novel scenarios, well-calibrated “self-assessment” capabilities help to foster safe functioning. We therefore evaluate Wasserstein-SqueezeDet not only in-domain but on corrupted and augmented test data as well as on other object detection datasets (see last paragraphs of this subsection).

Architecture

SqueezeDet takes an RGB input image and predicts three quantities: (i) 2D bounding boxes for detected objects (formalized as a 4D regression task), (ii) a confidence score for each predicted bounding box and (iii) the class of each detection. Its architecture is as follows: First, a sequence of convolutional layers extracts features from the input image. Next, dropout with a drop rate of $p = 0.5$ is applied to the final feature representations. Another convolutional layer, the ConvDet layer, finally estimates prediction candidates.

⁵ They are (on average) not due to small absolute residuals $\ll 1$ that go along with even smaller uncertainty estimates.

Table 4 Basic statistics of the harmonized object detection datasets. Dataset size and number of annotated objects are reported for train data (first two columns) and test data (last two columns). For details on dataset harmonization, see text and references therein

Dataset	Train		Test	
	# Images	# Objects	# Images	# Objects
KITTI	3622	15,254	3387	12,673
SynScapes	19,998	906,827	4998	226,390
A2D2	22,731	121,320	4186	36,544
Nightowls	30,064	50,225	6595	10,766
NuImages	58,803	410,462	14,377	97,014
BDD100k	69,281	843,963	9919	123,752

In more detail, SqueezeDet predictions are based on so-called anchors, initial bounding boxes with prototypical shapes. The ConvDet layer computes for each such anchor a confidence score, class scores and offsets to the initial position and shape. The final prediction outputs are obtained by applying a non-maximum-suppression (NMS) procedure to the prediction candidates. The original loss of SqueezeDet is the sum of three terms. It reads $L_{\text{SqueezeDet}} = L_{\text{regres}} + L_{\text{conf}} + L_{\text{class}}$ with the bounding box regression loss L_{regres} , a confidence-score loss L_{conf} and the object-classification loss L_{class} . Our modification of the learning objective is restricted to the L2 regression loss:

$$L_{\text{regres}} = \frac{\lambda_{\text{bbox}}}{N_{\text{obj}}} \sum_{i=1}^W \sum_{j=1}^H \sum_{k=1}^K \sum_{\xi \in \{x,y,w,h\}} I_{ijk} \left[(\delta \xi_{ijk} - \delta \xi_{ijk}^G)^2 \right] \tag{5}$$

with $\delta \xi_{ijk}$ and $\delta \xi_{ijk}^G$ being estimates and ground truth expressed in coordinates relative to the k -th anchor at grid point (i, j) where $\xi \in \{x, y, w, h\}$. See Wu et al. (2017) for descriptions of all other loss parameters. Applying W-dropout component-wise to this 4D regression problem yields

$$L_{\text{regres,W}} = \frac{\lambda_{\text{bbox}}}{N_{\text{obj}}} \sum_{i=1}^W \sum_{j=1}^H \sum_{k=1}^K \sum_{\xi \in \{x,y,w,h\}} I_{ijk} \left[\mathcal{W}(\xi_{ijk}) \right],$$

where

$$\mathcal{W}(\xi_{ijk}) = \left(\mu_{\delta \xi_{ijk}} - \delta \xi_{ijk}^G \right)^2 + \left(\sqrt{\sigma_{\delta \xi_{ijk}}^2} - \sqrt{\left(\mu_{\delta \xi_{ijk}} - \delta \xi_{ijk}^G \right)^2 + \sigma_{\delta \xi_{ijk}}^2} \right)^2$$

with $\mu_{\delta \xi_{ijk}} = \frac{1}{L} \sum_{l=1}^L \delta \xi_{ijk}^{(l)}$ being the sample mean and $\sigma_{\delta \xi_{ijk}}^2 = \frac{1}{L} \sum_{l=1}^L (\delta \xi_{ijk}^{(l)} - \mu_{\delta \xi_{ijk}})^2$ being the sample variance over L dropout predictions $\delta \xi_{ijk}^{(l)}$ for $\xi \in \{x, y, w, h\}$.

Datasets

We train SqueezeDet networks on six traffic scene datasets: KITTI (Geiger et al., 2012), SynScapes (Wrenninge and Unger, 2018), A2D2 (Geyer et al., 2020), Nightowls (Neumann et al., 2018), NuImages (NuScenes) (Caesar et al., 2020) and BDD100k (Yu et al., 2020). They differ from each other in dataset size (the large BDD100k dataset contains almost 20 times more images than the small KITTI dataset, see Table 4), time of day (Nightowls comprises only nighttime images) and data acquisition (SynScapes is simulation-based). For further information on the datasets, see Table 10 in Appendix A.5. We employ image

Table 5 Regression performance and uncertainty quality of SqueezeDet-type networks on KITTI data. W-SqueezeDet (W-SqzDet) is compared with the default MC-SqueezeDet (MC-SqzDet). The values of NLL, ECE and WS are aggregated across their respective four dimensions, for details see Appendix A.5 and Table 12 therein

Measure	Train		Test	
	MC-SqzDet	W-SqzDet	MC-SqzDet	W-SqzDet
mIoU (\uparrow)	0.705	0.691	0.695	0.694
RMSE (\downarrow)	8.769	9.832	14.666	14.505
NLL (\downarrow)	8.497	2.770	25.704	6.309
ECE (\downarrow)	0.615	0.193	0.825	0.433
WS (\downarrow)	1.421	0.315	2.831	0.900
ETL _{0.99}	22.358	8.853	42.101	18.223

Bold values indicate the best value that was reached for a given evaluation measure

sizes of 672×384 and rescale all datasets (except for KITTI⁶) accordingly. To facilitate cross-dataset model evaluations (see paragraphs on OOD analyses in this section), we group the various object classes of the six datasets into three main categories: ‘pedestrian’, ‘cyclist’ and ‘vehicle’ (see Table 11 in Appendix A.5 for the object class mapping). Some static or rare object classes are discarded.

Technical aspects We compare MC-SqueezeDet, i.e., standard SqueezeDet with activated dropout at inference, with W-SqueezeDet that uses W-dropout instead of the original MSE regression loss. All models are trained for 300, 000 mini-batches of size 20. After training, we keep dropout active and compute 50 forward passes for each test image. The detections from all forward passes are clustered using k-means (Bishop, 2006).⁷ The number of clusters is chosen for each image to match the average number of detections across the 50 forward passes. Each cluster is summarized by its mean detection and standard deviation. To ensure meaningful statistics, we discard clusters with 4 or less detections. The cluster means are matched with ground truth. We exclude predictions from the evaluation if their IoU with ground truth is ≤ 0.1 . For each dataset, SqueezeDet’s maximum number of detections is chosen proportionally to the average number of ground truth objects per image.

In-data evaluation

To assess model performance, we report the mean intersection over union (mIoU) and RMSE (in pixel space) between predicted bounding boxes and matched ground truths. The quality of the uncertainty estimates is measured by (coordinate-wise) NLL, ECE, WS and ETL. Table 5 shows a summary of our results on train and test data for the KITTI dataset. The results for NLL, ECE, WS and ETL have been averaged across the 4 regression coordinates. MC-SqueezeDet (abbreviated as MC-SqzDet) and W-SqueezeDet (W-SqzDet) show comparable regression results in terms of RMSE and mIoU, with slight advantages for MC-SqueezeDet. At this point, we only consider versions of SqueezeDet that provide uncertainty scores. For a discussion regarding performance degradation w.r.t. the deterministic

⁶ For KITTI, we crop images in x-direction to avoid strong distortions due to its high aspect ratio. In y-direction, only a minor upscaling is applied.

⁷ Using the density-based clustering technique HDBSCAN (Campello et al., 2013) yields comparable results especially w.r.t. the relative ordering of the methods.



Fig. 9 In-data and out-of-data evaluation of MC-SqueezeDet (lhs) and W-SqueezeDet (rhs) on six OD datasets. We consider regression quality (RMSE, top row) and uncertainty quality (ECE, bottom row). For each heatmap entry, the row label refers to the training dataset, the column label to the test dataset. Thus, diagonal matrix elements are in-data evaluations, non-diagonal elements are OOD analyses. W-SqueezeDet provides substantially smaller ECE values both in-data and out-of-data

SqueezeDet (approximately 10%, see Table 13), please refer to Appendix A.5. Considering uncertainty quality, we find substantial advantages for W-SqueezeDet across all evaluation measures. These advantages are due to the estimation of heteroscedastic aleatoric uncertainty during training (see also the test statistics ‘trajectories’ during training for BDD100k in Fig. 18 in Appendix A.5).

The test RMSE and ECE values of all six OD datasets are visualized as diagonal elements in Fig. 9. The (mostly) ‘violet’ RMSE diagonals for MC-SqueezeDet and W-SqueezeDet (top row of Fig. 9) again indicate comparable regression performances. Datasets are ordered by size from small (top) to large (bottom). The large NuImages test set occurs to be the most challenging one. Regarding ECE (bottom row of Fig. 9), W-SqueezeDet performs consistently stronger, see the ‘violet’ W-SqueezeDet diagonal (smaller values) and the ‘red’ MC-SqueezeDet diagonal (higher values). These findings qualitatively resemble those on the standard regression datasets and indicate that W-dropout works well on a modern application-scale network.

To analyze how well these OD uncertainty mechanisms function on test data that is structurally different from training data, we consider two types of out-of-data analyses in the following: first, we study SqueezeDet models that are trained on one OD dataset and evaluated on the test sets of the remaining five OD datasets. A rather ‘semantic’ OOD study as features like object statistics and scene composition vary between training and OOD test sets. Second, we consider networks that are trained on one OD dataset and evaluated on corrupted versions (defocus blur, Gaussian noise) of the respective test set, thus

Table 6 Out-of-data evaluation of MC-SqueezeDet (MC-SqzDet) and W-SqueezeDet (W-SqzDet) on distorted OD datasets. Each model is trained on the original dataset and evaluated on two modified versions of the respective test set: a blurred one (first two columns) and a noisy one (last two columns), see text for details. We report the expected calibration error (ECE) and find W-SqueezeDet to perform better than MC-SqueezeDet on most datasets

Dataset	Defocus blur		Gaussian noise	
	MC-SqzDet	W-SqzDet	MC-SqzDet	W-SqzDet
KITTI	1.034	1.082	1.021	1.084
SynScapes	1.081	0.503	0.941	0.910
A2D2	0.921	0.295	1.143	0.617
Nightowls	1.067	0.803	0.992	0.682
NuImages	0.908	0.332	0.760	0.849
BDD100k	1.012	0.390	0.833	0.633

Bold values indicate the best value that was reached for a given evaluation measure

facing changed ‘low-level’ features, i.e. less sharp edges due to blur and textures overlaid with pixel noise, respectively.

Out-of-data evaluation on other OD datasets

We train one SqueezeDet on each of the six OD datasets and evaluate each of these models on the test sets of the remaining 5 datasets. The resulting OOD regression scores and OOD ECE values are visualized as off-diagonal elements in Fig. 9 for MC-SqueezeDet (left column) and W-SqueezeDet (right column). Since datasets are ordered by size (a rough proxy to dataset complexity), the upper triangular matrix corresponds to cases in which the evaluation dataset is especially challenging (“easy to hard”), while the lower triangular matrix subsumes easier test sets compared to the respective i.i.d. test set (“hard to easy”). Accordingly, we observe (on average) lower RMSE values in the lower triangular matrix for both SqueezeDet variants. The ECE values of W-SqueezeDet are once more smaller (‘violet’) compared to MC-SqueezeDet (‘red’). The ECE diagonal of W-SqueezeDet is visually more pronounced compared to the one of MC-SqueezeDet since uncertainty calibration is effectively optimized during the training of W-SqueezeDet. The Nightowls dataset causes a cross-shaped pattern, indicating that neither transfers of Nightowls models to other datasets nor transfers from other models to Nightowls work well. This behavior can be understood as the feature distributions of Nightowls’ nighttime images diverge from the (mostly) daytime images of the other datasets. The high uncertainty quality of W-SqueezeDet is underpinned by the evaluations of NLL and WS (see Fig. 17 and text in Appendix A.5).

Out-of-data evaluation on corrupted datasets

In contrast to the analysis above, we now focus on ‘non-semantic’ data shifts due to technical distortions. For each test set, we generate a blurred and a noisy version.⁸ Two examples of these transformations can be found in Fig. 16 in Appendix A.5. In accordance with previous results, W-SqueezeDet provides smaller ECE values compared to MC-SqueezeDet on most blurred and noisy test sets (see Table 6). We observe a less substantial

⁸ We employ the `imgaug` library (<https://github.com/aleju/imgaug>) and apply defocus blur (severity of “1”) and additive Gaussian noise (i.i.d. per pixel, drawn from the distribution $\mathcal{N}(0, 20)$), respectively.

deterioration of uncertainty quality for blurring compared to adding pixel noise, possibly because the latter one more strongly affects short-range pixel correlations that the networks rely on.

5 Conclusion

The prevailing approaches to uncertainty quantification rely on parametric uncertainty estimates by means of a dedicated network output. In this work, we propose a novel type of uncertainty mechanism, *Wasserstein dropout*, that quantifies (aleatoric) uncertainty in a purely non-parametric manner: by revisiting and newly assembling core concepts from existing dropout-based uncertainty methods, we construct distributions of randomly drawn sub-networks that closely approximate the actual data distributions. This is achieved by a natural extension of the Euclidean metric (L_2 -loss) for points to the 2-Wasserstein metric for distributions. In the limit of vanishing distribution width, i.e. vanishing uncertainty, both metrics coincide. Assuming Gaussianity and making a bootstrap approximation, the metric can be replaced by a compact loss objective affording stable training. To the best of our knowledge, W-dropout is the first *non-parametric* method to model aleatoric uncertainty in neural networks. It outperforms the ubiquitous parametric approaches, as, e.g., shown by our comparison to deep ensembles (PU-DE).

An extensive additional study of uncertainties under data shift further reveals advantages of W-dropout models compared to deep ensembles (PU-DE) and parametric models combined with dropout (PU-MC): the Wasserstein-based technique still provides (on average) better calibrated uncertainty estimates while coming along with a higher stability across a variety of datasets and data shifts. In contrast, we find parametric uncertainty estimation (PU) to be prone to instabilities that are only partially cured by the regularizing effects of explicit or implicit (dropout-based) ensembling (PU-DE, PU-MC). With respect to worst-case scenarios, W-dropout networks are by a large margin better than either PU-DE or PU-MC. This makes W-dropout especially suitable for safety-critical applications like automated driving or medical diagnosis where (even rarely occurring) inadequate uncertainty estimates might lead to injuries and damage. Furthermore, while our theoretical derivation focuses on aleatoric uncertainty, the presented distribution-shift experiments suggest that W-dropout is also able to capture epistemic uncertainty. Finding a theoretical explanation for that is subject of future research.

With respect to computational demands, W-dropout is roughly equivalent to MC dropout (MC) and, in fact, could be used as a drop-in replacement for the latter. While L -fold sampling of sub-networks increases the training complexity, we observe an increase of training time that is significantly below L in our implementation. Inference is performed in the same way for both methods and thus also their run-time complexities are equivalent. In comparison to deep ensembles, W-dropout's use of a single network reduces requirements on training and storage at the expense of multiple forward passes during inference. This property is shared with MC and approaches exist to reduce the prediction cost, for instance last-layer MC allows sampling-free inference (see also Postels et al. (2019)).

In addition to the toy and 1D regression experiments, SqueezeDet is selected as a representative of large-scale object detection networks. We find the above mentioned properties of Wasserstein dropout to carry over to Wasserstein-SqueezeDet, namely the enhanced uncertainty quality and its increased stability under different types of data shifts. At the same time observed performance losses are minimal. Overall, our experiments

on SqueezeDet show that W-dropout scales to larger networks relevant for practical applications.

When intending to employ uncertainty estimation as a safeguard against model errors, distributional properties of the normalized residuals gain importance. To address such properties we introduce the ETL as a measure for rare and critical cases where uncertainty is strongly underestimated. While we find that W-dropout leads to more Gaussian residuals compared to our benchmarks we still observe remaining deviations. A priori, it is not clear whether the aleatoric uncertainty in complex data is Gaussian or whether such rare cases could be better described with more heavy-tailed distributions. If this is the case, the question arises of whether dropout mechanisms are flexible enough to model distributions outside the Gaussian regime, which we investigated in Sicking et al. (2020).

Taking a step back, the idea of exchanging the distributions allows to apply our framework to a variety of tasks beyond regression and makes the migration from single point modelling to full distributions a rather general concept. Replacing, e.g., Gaussians with Dirichlet distributions makes an application to classification conceivable, where Malinin and Gales (2018) employ parametric (Dirichlet) distributions to quantify uncertainty. Conceptually, our findings suggest that distribution modeling based on *sampling* generalizes better compared to parameterized counterparts. An observation that might find applications far outside the scope of uncertainty quantification.

Appendix A Extension to the empirical study

This part accompanies our paper “*Wasserstein Dropout*” and provides further in-depth information. Large parts of the empirical evaluation on toy data and standard regression datasets can be found in Sect. A, including details on the datasets, more granular evaluations and additional toy data experiments. Details on the object detection datasets and supplementary evaluations of SqueezeDet are located in Sect. A.5. As W-dropout exhibits the hyper-parameters p (drop rate) and L (sample size), we test various values in Sect. B, finding no strong correlation between result and parameter choices. We close with a discussion on the relation between uncertainty measures and their respective sensitivity in Sect. C.

Complementing the evaluation sketched in the body of the paper, Sect. 4, we provide more details on the training setup and benchmark approaches in the following subsection. Further information on the toy dataset experiments can be found in Sect. A.2. The same holds for the 1D regression experiments in Sect. A.3, which we extend by evaluations on dataset level that were skipped in the main text. A close look at the predicted uncertainties (per method) on these datasets is given via scatter plots in Sect. A.4. Details on OD dataset preprocessing and SqueezeDet results are found in the last subsection.

A.1 Experimental setup

The experimental setup used for the toy data and 1D regression experiments is presented in two parts: first, technical details of the benchmark approaches we compare with and second, a description of the neural networks and training procedures we employ.

For MC dropout, we choose the regularization coefficient λ by grid search on the set $\lambda \in \{0, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ and find $\lambda = 10^{-6}$ to provide the best overall results for the 1D regression datasets. A variant of MC dropout that optimizes its layer-specific drop rates during training is Concrete dropout (CON-MC): all its initial drop rates are

set to $p_{\text{initial}} = 0.1$. The hyper-parameters $wr = l^2/(\tau N)$ and $dr = 2/N$ are determined by the number of training datapoints N , prior length scale $l = 10^{-3}$ and $\tau(N) \in [10^{-3}, 2]$ that decreases monotonically with N . For PU and PU-EV networks, we ensure positivity constraints using softplus (Glorot et al., 2011) and optimize Gaussian NLL and t-distribution NLL, respectively. The regularization coefficient of PU-EV is set to $\lambda = 10^{-2}$, determined by a grid search considering the parameter range $\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.5\}$. For SWAG, we start to estimate the low-rank Gaussian proxy (rank $r = 20$) for the NN weight distribution after training for $n/2$ epochs, with n being the total number of training epochs.

We categorize the toy and 1D regression datasets as follows: small datasets {toy-hf, yacht, diabetes, boston, energy, concrete, wine-red}, large datasets {toy-noise, abalone, kin8nm, power, naval, california, superconduct, protein} and very large datasets {year}. For small datasets, NNs are trained for 1000 epochs using mini-batches of size 100. All results are 10-fold cross validated. For large datasets, we train for 150 epochs and apply 5-fold cross validation. We keep this large-dataset setting for the very large ‘year’ dataset but increase mini-batch size to 500.

All experiments are conducted on Core Intel (R) Xeon (R) Gold 6126 CPUs and NVidia Tesla V100 GPUs. Conducting the described experiments with cross validation on one CPU takes 20 h for toy data, 130 h for 1D regression datasets and approximately 100 h for object regression on the GPU.

A.2 Toy datasets: systematic evaluation and further experiments

The toy-noise and toy-hf datasets are sampled from $f_{\text{noise}}(x) \sim \mathcal{N}(0, \exp(-0.02 x^2))$ for $x \in [-15, 15]$ and $f_{\text{hf}}(x) = 0.25 x^2 - 0.01 x^3 + 40 \exp(-(x+1)^2/200) \sin(3x)$ for $x \in [-15, 20]$, respectively. Standard normalization is applied to input and output values. Detailed evaluations of the considered uncertainty methods on these datasets are given in Table 7.

To illustrate the capabilities and limitations of MC dropout regarding the modeling of aleatoric uncertainty, we consider the toy-noise dataset again and systematically vary MC’s regularization parameter λ (see Fig. 10, λ decreases from left to right). As MC dropout’s uncertainty estimates contain an additive constant term proportional to λ , tuning this parameter allows to model the *average* aleatoric uncertainty (the ideal λ in Fig. 10 is between $\lambda = 10^{-6}$ and $\lambda = 10^{-5}$). Input dependencies of noise (heteroscedasticity) can however not be incorporated, i.e. even an optimized λ causes systematic over- and underestimations of the data uncertainty in many cases. This is in contrast to W-dropout.

Having shown that W-dropout can approximate input-dependent data uncertainty appropriately (see Fig. 1), we now analyze its ability to match ground truth uncertainties σ_{true} more systematically. Therefore, we fit a ‘noisy line’ toy dataset that is given by (x_i, y_i) with $x_i \sim \mathcal{U}(-1, 1)$ and $y_i \sim \mathcal{N}(0, \sigma_{\text{true}})$. The ground truth standard deviations take the values $\sigma_{\text{true}} = 0, 0.1, 0.2, 0.5, 1, 2, 5, 10$. Fig. 11 emphasizes that W-dropout provides accurate uncertainty estimates for both small and large noise levels. Minor x-dependent fluctuations (see ‘whiskers’ in Fig. 11) decrease monotonically with σ_{true} .

A.3 Standard regression datasets: systematic evaluation

An overview on the 1D regression datasets providing basic statistics and information on preprocessing is given in Table 8. Evaluations of RMSE, NLL, ECE and WS on dataset level can be found in Table 9. Moreover, we extend our evaluation by a deterministic

Table 7 Regression performance and uncertainty quality of networks with different uncertainty mechanisms. All scores are calculated on the test set of toy-hf and toy-noise, respectively

Measure	Dataset	Swag	de	pu	pu-ev	pu-de
RMSE (↓)	toy-hf	0.696	0.660	0.691	0.691	0.690
NLL (↓)	toy-hf	85.331	52.444	−0.098	1.855	−0.100
ECE (↓)	toy-hf	1.472	1.584	0.548	0.500	0.524
WS (↓)	toy-hf	9.043	7.413	0.233	0.242	0.243
RMSE (↓)	toy-noise	1.006	1.006	1.006	1.006	1.006
NLL (↓)	toy-noise	6934.498	1.14×10^4	−0.374	1.555	−0.374
ECE (↓)	toy-noise	1.541	1.642	0.062	0.098	0.084
WS (↓)	toy-noise	63.760	83.590	0.028	0.064	0.048
Measure	Dataset	pu-mc	con-mc	mc	w-drop	
RMSE (↓)	toy-hf	0.694	0.701	0.696	0.678	
NLL (↓)	toy-hf	−0.083	17.616	13.370	−0.055	
ECE (↓)	toy-hf	0.544	1.380	1.352	0.428	
WS (↓)	toy-hf	0.233	4.356	3.830	0.222	
RMSE (↓)	toy-noise	1.007	0.995	1.006	1.013	
NLL (↓)	toy-noise	−0.370	6.57×10^4	1.723	−0.330	
ECE (↓)	toy-noise	0.066	1.730	0.645	0.107	
WS (↓)	toy-noise	0.030	5.03×10^4	0.693	0.054	

network which we obtain as one of the members of the DE ensemble. For better overview, we reproduce Fig. 6 (top row) with this member added, see Fig. 12. The small performance deterioration of this member compared to the full DE ensemble can be attributed to the averaging over the outcomes of the five ensemble components which suppresses stochastic fluctuations.

A.4 Residual-uncertainty scatter plots

Visual inspection of uncertainties can be helpful to understand their qualitative behavior. We scatter model residuals $\mu_i - y_i$ (respective x-axis in Fig. 14) against model uncertainties σ_i (resp. y-axis in Fig. 14). For a *hypothetical ideal* uncertainty mechanism, we expect $(y_i - \mu_i) \sim \mathcal{N}(0, \sigma_i)$, i.e., model residuals following the predictive uncertainty distribution. More concretely, 68.3% of all $(y_i - \mu_i)$ would lie within the respective interval $[-\sigma_i, \sigma_i]$ and 99.7% of all $(y_i - \mu_i)$ within $[-3\sigma_i, 3\sigma_i]$. Fig. 13 visualizes this hypothetical ideal. It is generated as follows: We draw 3, 000 standard deviations $\sigma_i \sim \mathcal{U}(0, 2)$ and sample residuals r_i from the respective normal distributions, $r_i \sim \mathcal{N}(0, \sigma_i)$. The pairs (r_i, σ_i) are visualized. By construction, uncertainty estimates now ideally match residuals in a distributional sense.

Geometrically, the described Gaussian properties imply that 99.7% of all scatter points, e.g., in Fig. 14, should lie above the blue 3σ lines and 68.3% of them above the yellow 1σ lines. For toy-noise, abalone and superconduct (first, third and fourth row in Fig. 14), PU, PU-DE and W-dropout qualitatively fulfill this requirement while MC, MC-LL and DE tend to underestimate uncertainties. This finding is in accordance with our systematic evaluation. The naval dataset (second row in Fig. 14) poses an exception in this regard as

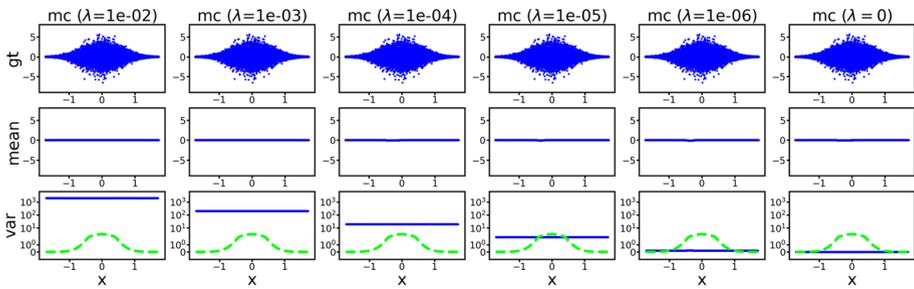


Fig. 10 MC dropout and aleatoric uncertainty. The regularization parameter λ of MC dropout allows to model the average (homoscedastic) noise level of a dataset. As the regularizer is not input-dependent, it does not capture the x -dependency of the noise level, i.e. the heteroscedasticity of the dataset, see third row

Fig. 11 Standard deviation σ_{w_drop} of W-dropout (y-axis) when fitted to a toy dataset with ground truth standard deviation σ_{gt} (x-axis, see text for details). The bisecting line is shown in gray. While σ_{w_drop} exhibits fluctuations (black ‘whiskers’ at 10% and 90% quantile), it provides on average accurate estimates of the ground truth uncertainty. Both mean value (blue cross) and median value (orange bar) of σ_{w_drop} are close to the bisector (Color figure online)

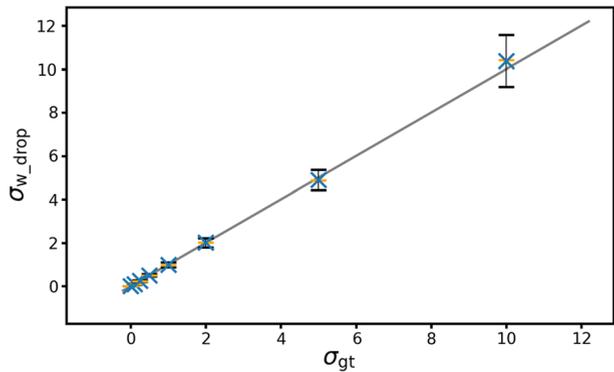


Table 8 Details on 1D regression datasets. Ground truth (gt) is partially preprocessed to match the 1D regression setup

Dataset	# features	# data points	Source	Remarks
Yacht	6	308	UCI	
Diabetes	7	442	StatLib	
Boston	13	506	StatLib	
Energy	8	768	UCI	Only the “cooling load” label is used.
Concrete	8	1030	UCI	
Wine-red	11	1599	UCI	
Abalone	7	4176	UCI	The categorical input feature “sex” is omitted.
kin8nm	8	8192	Delve	
Power	4	9568	UCI	
Naval	16	11,934	UCI	Only the “turbine” label is used.
California	8	20,640	StatLib	
Superconduct	81	21,263	UCI	
Protein	9	45,730	UCI	
Year	90	515,345	UCI	

Table 9 Regression performance and uncertainty quality of networks with different uncertainty mechanisms. The scores are calculated on the test sets of 14 standard regression datasets. Please note that the deterministic network does not provide uncertainty information

Measure	Dataset	Deterministic	swag	de	pu	pu-ev	pu-de	pu-mc	con-mc	mc	w-drop
RMSE (↓)	Yacht	0.045	0.055	0.042	0.062	0.051	0.049	0.076	0.103	0.074	0.075
NLL (↓)	Yacht	–	– 3.010	– 3.560	– 2.370	– 0.438	– 3.017	– 2.881	– 2.059	– 2.202	– 2.489
ECE (↓)	Yacht	–	0.757	0.653	0.863	0.890	1.087	1.093	1.010	1.093	1.113
WS (↓)	Yacht	–	0.470	0.379	0.668	1.228	0.544	0.560	0.481	0.536	0.537
RMSE (↓)	Diabetes	0.971	0.990	0.920	0.816	0.832	0.790	0.773	0.792	0.843	0.886
NLL (↓)	Diabetes	–	61.217	7.629	1712.520	11.505	7.423	2.302	3.427	6.014	2.356
ECE (↓)	Diabetes	–	1.456	0.942	1.353	1.370	0.945	0.854	0.975	1.108	0.909
WS (↓)	Diabetes	–	7.704	1.717	19.552	19.182	1.936	1.230	1.630	2.229	1.265
RMSE (↓)	Boston	0.345	0.352	0.322	0.339	0.349	0.329	0.287	0.317	0.310	0.318
NLL (↓)	Boston	–	10.739	6.116	553.188	4.003	3.590	0.050	– 0.371	– 0.040	– 0.287
ECE (↓)	Boston	–	1.170	0.882	1.294	1.240	0.876	0.644	0.636	0.688	0.624
WS (↓)	Boston	–	2.987	1.593	9.233	5.288	1.490	0.658	0.493	0.674	0.581
RMSE (↓)	Energy	0.097	0.077	0.118	0.166	0.106	0.079	0.107	0.080	0.075	–
NLL (↓)	Energy	–	– 1.048	0.628	5.882	1.751	– 2.007	– 2.056	– 1.703	– 1.815	– 2.047
ECE (↓)	Energy	–	0.567	0.602	1.037	0.819	0.502	0.705	0.637	0.819	0.648
WS (↓)	Energy	–	0.672	0.723	2.183	1.508	0.374	0.379	0.332	0.424	0.339
RMSE (↓)	Concrete	0.258	0.258	0.241	0.263	0.260	0.244	0.237	0.270	0.235	0.248
NLL (↓)	Concrete	–	2.726	4.899	34.402	2.496	0.208	– 0.978	– 0.672	– 0.890	– 0.868
ECE (↓)	Concrete	–	0.672	0.649	0.871	0.763	0.528	0.384	0.395	0.431	0.400
WS (↓)	Concrete	–	1.279	1.402	3.217	2.605	0.773	0.230	0.256	0.267	0.253
RMSE (↓)	Wine-red	0.981	0.931	0.835	0.824	0.817	0.771	0.783	0.748	0.784	0.807
NLL (↓)	Wine-red	–	10.343	1.352	1.41×10^5	10.201	3.578	11.142	2.485	1.962	0.830
ECE (↓)	Wine-red	–	0.946	0.488	0.765	0.816	0.456	0.541	0.677	0.664	0.549
WS (↓)	Wine-Red	–	2.487	0.717	35.052	55.450	0.813	0.914	1.154	1.014	0.536
RMSE (↓)	Abalone	0.673	0.798	0.701	0.653	0.657	0.654	0.636	0.636	0.684	0.718
NLL (↓)	Abalone	–	13.289	31.002	2.05×10^8	2.486	– 0.073	– 0.111	9.206	1.017	0.256

Table 9 (continued)

Measure	Dataset	Deterministic	swag	de	pu	pu-ev	pu-de	pu-mc	con-mc	mc	w-drop
ECE (↓)	Abalone	–	1.181	1.276	0.252	0.250	0.238	0.270	1.082	0.496	0.402
WS (↓)	Abalone	–	3.149	4.624	3.07×10^7	286.448	0.157	0.141	2.630	0.726	0.462
RMSE (↓)	kin8nm	0.269	0.259	0.246	0.272	0.276	0.253	0.269	0.313	0.266	0.261
NLL (↓)	kin8nm	–	–0.393	1.905	–0.142	1.274	–0.866	–0.631	–0.612	–0.678	–0.855
ECE (↓)	kin8nm	–	0.453	0.677	0.462	0.362	0.205	0.561	0.223	0.491	0.151
WS (↓)	kin8nm	–	0.552	1.202	0.629	0.405	0.185	0.334	0.203	0.306	0.080
RMSE (↓)	Power	0.224	0.228	0.219	0.225	0.227	0.221	0.228	0.234	0.226	0.226
NLL (↓)	Power	–	1.682	12.696	–0.921	0.870	–1.024	–1.007	–0.470	–0.720	–0.788
ECE (↓)	Power	–	0.755	1.075	0.154	0.176	0.135	0.172	0.415	0.656	0.264
WS (↓)	Power	–	1.306	2.943	0.127	0.193	0.071	0.098	0.520	0.393	0.318
RMSE (↓)	Naval	0.087	0.250	0.030	0.163	0.194	0.165	0.169	0.345	0.118	0.100
NLL (↓)	Naval	–	–0.198	–2.815	–2.358	0.327	–1.405	–1.464	–0.522	–0.701	–1.479
ECE (↓)	Naval	–	1.230	0.898	0.677	0.887	1.235	0.760	0.378	1.233	0.888
WS (↓)	Naval	–	0.615	0.487	0.357	0.931	0.606	0.454	0.297	0.632	0.483
RMSE (↓)	California	0.453	0.444	0.430	0.674	0.514	0.475	0.549	0.456	0.436	0.448
NLL (↓)	California	–	0.813	7.305	–0.494	1.266	–0.612	–0.560	1.441	–0.236	–0.213
ECE (↓)	California	–	0.469	0.829	0.248	0.212	0.251	0.314	0.625	0.650	0.251
WS (↓)	California	–	0.745	1.965	0.181	306.628	0.162	0.194	0.989	0.357	0.312
RMSE (↓)	Superconduct	0.309	0.305	0.290	0.340	0.341	0.326	0.346	0.345	0.318	0.310
NLL (↓)	Superconduct	–	–0.242	3.045	–0.558	0.622	–1.340	–1.169	–0.065	–0.341	–0.954
ECE (↓)	Superconduct	–	0.346	0.515	0.151	0.181	0.204	0.243	0.381	0.916	0.183
WS (↓)	Superconduct	–	0.451	1.126	0.204	0.250	0.129	0.141	0.569	0.487	0.161
RMSE (↓)	Protein	0.612	0.615	0.575	0.723	0.696	0.701	0.666	0.654	0.610	0.610
NLL (↓)	Protein	–	0.560	4.314	0.117	1.515	–0.132	–0.080	3.149	0.141	0.073
ECE (↓)	Protein	–	0.440	0.649	0.225	0.567	0.288	0.325	0.809	0.565	0.265

Table 9 (continued)

Measure	Dataset	Deterministic	swag	de	pu	pu-ev	pu-de	pu-mc	con-mc	mc	w-drop
WS (↓)	Protein	–	0.592	1.379	0.156	8.242	0.182	0.183	1.483	0.336	0.245
RMSE (↓)	Year	0.825	0.800	0.765	0.789	0.803	0.775	0.785	0.785	0.786	0.812
NLL (↓)	Year	–	11.250	12.064	0.169	1.861	–0.023	0.940	7.952	1.148	0.477
ECE (↓)	Year	–	1.151	1.003	0.249	0.230	0.261	1.043	1.126	1.219	0.369
WS (↓)	Year	–	2.848	2.543	0.187	0.163	0.164	0.539	2.461	0.625	0.454

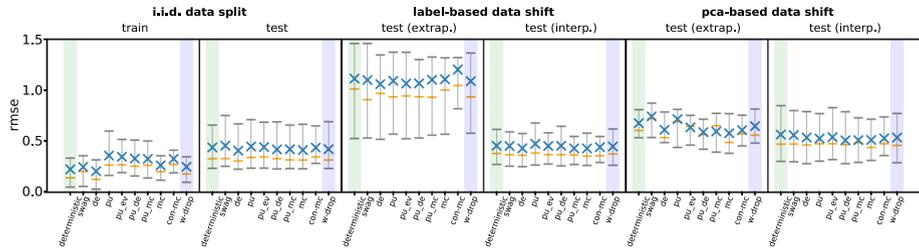


Fig. 12 Root-mean-square errors (RMSEs (\downarrow)) of different network types under i.i.d. conditions (first and second panel) and under various kinds of data shift (third to sixth panel). W-dropout (light blue background) is compared to 9 benchmark approaches including a deterministic model (light green background). Each blue cross is the mean over 14 1D regression datasets. Orange line markers indicate median values. The gray vertical bars reach from the 25% quantile (bottom horizontal line) to the 75% quantile (top horizontal line) (Color figure online)

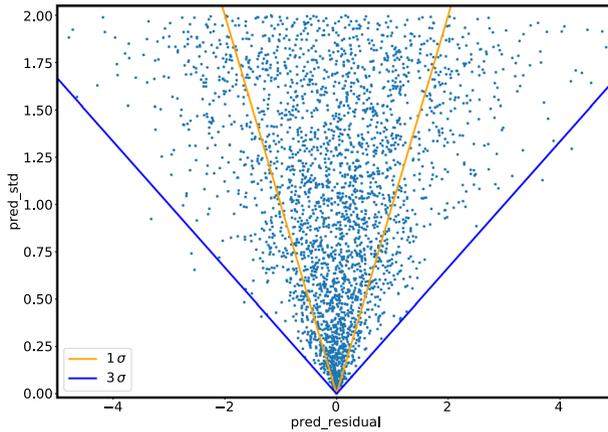


Fig. 13 Prediction residuals (x-axis) and predictive uncertainty (y-axis) for a *hypothetical* ideal uncertainty mechanism. The Gaussian errors are matched by Gaussian uncertainty predictions at the exact same scale. 68.3% of all uncertainty estimates (plot points) lie above the orange 1σ -lines and 99.7% of them above the blue 3σ -lines (Color figure online)

all uncertainty methods lead to comparably convincing uncertainty estimates. The small test RMSEs of all methods on naval indicate relatively small aleatoric uncertainties and model residuals. Epistemic uncertainty might thus be a key driving factor and coherently MC, MC-LL and DE perform well.

A.5 Object detection: systematic evaluation

We report basic information on the object detection (OD) datasets and their harmonization in the first paragraph of this subsection. Supplementary evaluations of SqueezeDet can be found subsequently in the second paragraph.

Details on OD datasets

The six OD datasets we consider are diverse in multiple dimensions as they capture traffic scenes from three continents (Asia, Europe and North America) and cover a broad set of

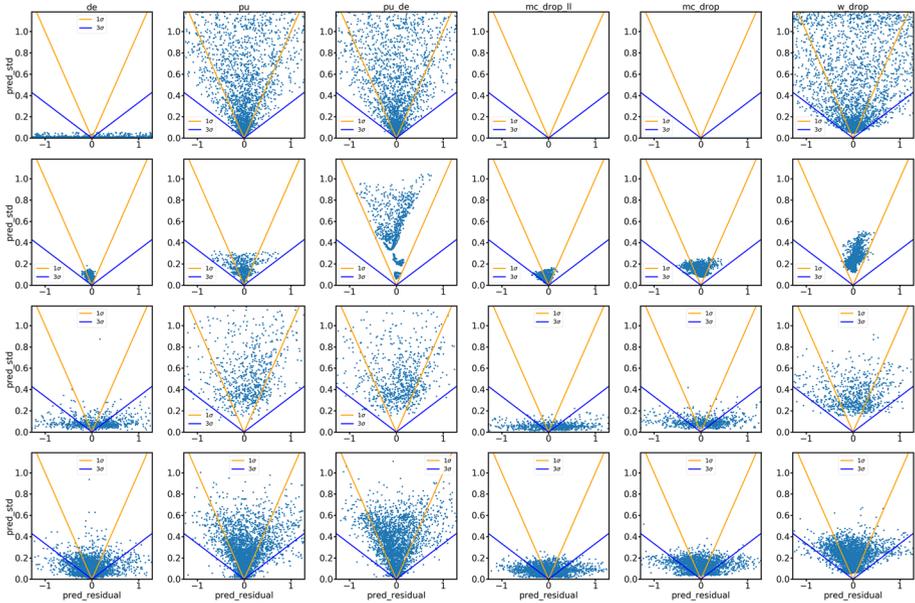


Fig. 14 Prediction residuals (respective x-axis) and predictive uncertainty (respective y-axis) for different uncertainty mechanisms (columns) and datasets (rows). Each light blue dot in each plot corresponds to one test data point. Realistic uncertainty estimates should lie mostly above the blue 3σ -lines. The datasets toy-noise, naval, abalone and superconduct are shown, from top to bottom (Color figure online)

Table 10 General information on the object detection datasets

Dataset	Place of data collection	Type	Train/test split
KITTI	Metropolitan area of Karlsruhe	Real	Semi-custom (sequence-based)
SynScapes	Simulation (only urban scenes)	Synthetic	Custom (random split)
A2D2	Highways and cities in Germany	Real	Custom (sequence-based)
Nightowls	Several cities across Europe	Real	Pre-defined
NuImages	Boston and 3 diverse areas of Singapur	Real	Pre-defined
BDD100k	New York, San Francisco Bay, Berkeley	Real	Pre-defined

scenarios ranging from cities and metropolitan areas over country roads to highways (see Table 10). They moreover differ in the average number of objects per image (see Table 4) that reaches its highest values for the simulation-based SynScapes dataset.⁹ Finally, both random and sequence-based train-test splits are considered. This variety is moreover reflected in the numerous object classes the different datasets provide. Their mappings to three main categories (‘pedestrian’, ‘cyclist’, ‘vehicle’) can be found in Table 11. Rare or

⁹ For A2D2, 2D bounding boxes are inferred from semantic segmentation ground truth.

Table 11 Harmonization of the object detection datasets. The various object classes of the six object detection datasets (rows) are grouped into the three main categories “vehicle”, “pedestrian” and “cyclist” (columns). Some classes are too rare or irregular and are thus discarded

Dataset	Vehicle	Cyclist	Pedestrian	Discarded
KITTI	Car	Cyclist	Pedestrian	Van, truck, tram, person-sitting, misc, dontcare
SynScapes	Car, motorbike, truck, bus	Cyclist	Pedestrian	Train
A2D2	Car, truck	Cyclist	Pedestrian	–
Nightowls	Motorbike	Cyclist	Pedestrian	Ignore-area
NuImages	Car, motorbike, truck, vehicle-other	Cyclist	Pedestrian	Movable-object
BDD100k	Car, motorbike, truck, bus, trailer, vehicle-other	Cyclist	Pedestrian, other-person	Train, rider, traffic-light, traffic-sign

irregular classes are removed. For KITTI, we moreover discard ‘van’, ‘truck’ and ‘person-sitting’, following the original SqueezeDet paper. To analyze uncertainty quality on distorted images, blurred and noisy versions of the test datasets are created. Figure 16 shows these transformations for two exemplary images from BDD100k (top row) and SynScapes (bottom row), respectively.

Comparison with the deterministic SqueezeDet

A comparison of the results of the deterministic SqueezeDet, MC-SqueezeDet and W-SqueezeDet for the KITTI test dataset can be found in Table 13. There, we additionally consider the mean average precision (mAP) score of these networks as well as the closely related measures of precision and recall.¹⁰ In contrast to the regression results on the 1D standard datasets, we observe more pronounced deviations between the different types of SqueezeDet and, in particular, note that the deterministic SqueezeDet yields OD scores that are 5–15% better than the ones of MC- and W-SqueezeDet. This finding equally concerns mAP and the regression scores of mIoU and RMSE. The OD capabilities of the two probabilistic networks are comparable, see also Fig. 15. It shows that performance losses are less caused by our specific version of dropout but rather generally by using dropout-based techniques (Table 12).

This performance gap between deterministic and probabilistic models can be understood when recalling that the anchor-based object proposals of SqueezeDet are piped through a multi-step post-processing to obtain the “final” detections. Dropout-enhanced models, in particular, require an additional step of clustering the stochastic proposals which may in some cases cause incorrect cluster assignments. Moreover, fewer resulting proposals can be matched with ground truth objects (see the precision values in Table 13). This might be attributed to the stochastic nature of the bounding boxes which leads to slightly increased errors and therefore, and in combination with the IoU threshold, to less matched proposals. Early experiments suggested that changes to the post-processing routines can contribute to mitigating large parts of these performance losses. As demonstrated in Fig. 15, we believe

¹⁰ In object detection, average precision (AP) can be understood as the area under the precision-recall curve that is obtained when sorting all predicted bounding boxes (for a given dataset) by their confidence scores.

Table 12 Regression performance and uncertainty quality of SqueezeDet-type networks on KITTI train/test data. W-SqueezeDet is compared with MC-SqueezeDet

Measure	MC-SqzDet	W-SqzDet	MC-SqzDet	W-SqzDet
	Train		Test	
mIoU (\uparrow)	0.705	0.691	0.695	0.694
RMSE (\downarrow)	8.769	9.832	14.666	14.505
NLL _x (\downarrow)	14.793	2.808	34.827	6.941
NLL _y	6.135	2.170	13.364	3.808
NLL _w	6.916	3.305	36.384	8.579
NLL _h	6.146	2.796	18.241	5.908
ECE _x (\downarrow)	0.560	0.148	0.748	0.330
ECE _y	0.659	0.180	0.835	0.419
ECE _w	0.523	0.147	0.888	0.520
ECE _h	0.716	0.296	0.83	0.465
WS _x (\downarrow)	1.729	0.283	3.06	0.830
WS _y	1.370	0.299	2.260	0.680
WS _w	1.145	0.243	3.485	1.203
WS _h	1.442	0.437	2.517	0.888
ETL _{0.99,x}	34.443	9.316	55.772	21.310
ETL _{0.99,y}	18.202	7.677	26.675	11.772
ETL _{0.99,w}	19.914	10.835	53.408	23.202
ETL _{0.99,h}	16.872	7.584	32.547	16.608

Bold values indicate the best value that was reached for a given evaluation measure

Table 13 Regression performance and uncertainty quality of SqueezeDet-type networks on KITTI test data. W-SqueezeDet (W-SqzDet) is compared with MC-SqueezeDet (MC-SqzDet) and the deterministic network (SqzDet). Extending our set of measures, we additionally report mean average precision (mAP) as well as (class-averaged) recall and precision

Measure	SqzDet	MC-SqzDet	W-SqzDet
mAP (\uparrow)	0.692	0.618	0.619
recall (\uparrow)	0.758	0.686	0.688
precision (\uparrow)	0.324	0.296	0.277
mIoU (\uparrow)	0.730	0.695	0.694
RMSE (\downarrow)	13.058	14.666	14.505
NLL (\downarrow)	–	25.704	6.309
ECE (\downarrow)	–	0.825	0.433
WS (\downarrow)	–	2.831	0.900
ETL _{0.99}	–	42.101	18.223

Bold values indicate the best value that was reached for a given evaluation measure

that the questions of performance and W-dropout are detached and therefore relegate a deeper exploration of proposal matching to future work.

Further results on SqueezeDet

Coordinate-wise regression results and uncertainty scores for MC-SqueezeDet and W-SqueezeDet on KITTI are shown in Table 12. While we observe noteworthy differences between coordinates, the relative ordering of MC-SqueezeDet and W-SqueezeDet for a given measure remains the same (Fig. 17).

Analyzing in-data and out-of-data NLL and WS values for all six datasets (see Fig. 17), we find results that qualitatively resemble those on ECE in Fig. 9. W-SqueezeDet

Fig. 15 Recall-precision curves of the deterministic SqueezeDet (blue), MC-SqueezeDet (orange) and W-SqueezeDet (green) for the object class “vehicle” in the KITTI test dataset. The AP values of the respective networks (for class “vehicle”) are given by the areas under the curves (Color figure online)

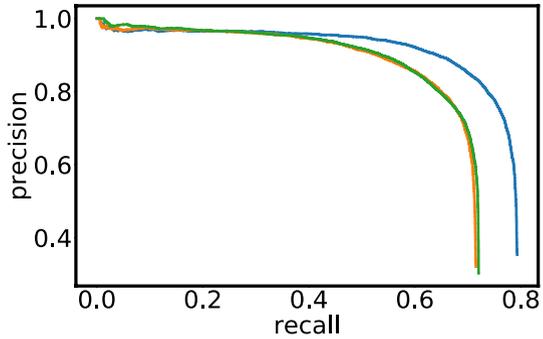


Fig. 16 Two exemplary object detection images from BDD100k (top row, real-world image) and SynScapes (bottom row, synthetic image), respectively. For each original image (left column), two corrupted versions are generated: a blurred one (middle column) and a noisy one (right column), see text for details

outperforms MC-SqueezeDet on the respective i.i.d. test set and also under data shift. For both uncertainty approaches, some NLL values are affected by outliers.

Finally, Fig. 18 visualizes how various regression and uncertainty (test) scores evolve during model training on the BDD100k dataset. MC-SqueezeDet (dashed) and W-SqueezeDet (solid) ‘converge’ with comparable speed (no changes to test RMSE and mIoU after 100,000 training steps) and reach similar final performances. W-SqueezeDet’s explicit optimization of uncertainty estimates yields larger standard deviations (center panel) and smaller values for NLL, ECE, WS and ETL compared to MC-SqueezeDet (center right panel, bottom row). For the unbounded scores NLL, WS and ETL, W-SqueezeDet exhibits higher stability during training.

Appendix B Stability w.r.t. hyper-parameters p and L

W-dropout possesses two hyper-parameters: the neuron drop rate p and the sample size L used to calculate the empirical estimates $\mu_{\bar{\theta}}(x_i)$ and $\sigma_{\bar{\theta}}(x_i)$. Here, we analyze the impact of these parameters on the quality of accordingly trained models.

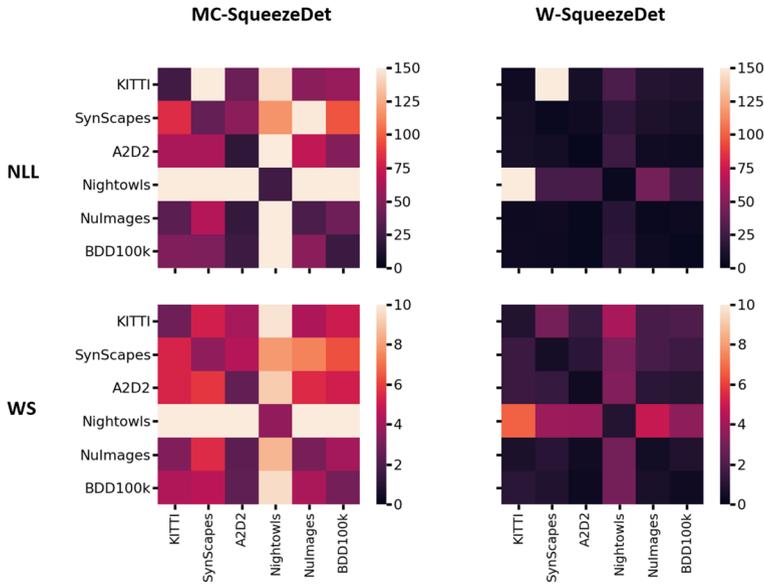


Fig. 17 In-data and out-of-data evaluation of MC-SqueezeDet (lhs) and W-SqueezeDet (rhs) on six OD datasets. We consider the negative log-likelihood (NLL, top row) and the Wasserstein measure (WS, bottom row). For each heatmap entry, the row label refers to the training dataset and the column label to the test dataset. Thus, diagonal matrix elements are in-data evaluations, non-diagonal elements are OOD analyses

For $p = 0.05, 0.1, 0.2, 0.3, 0.4, 0.5$, we observe only relatively small differences in both RMSE (see top panel of Fig. 19) and ECE (see bottom panel of Fig. 19). On train data, RMSE slightly deteriorates with increasing p , i.e., with decreasing complexity of the sub-networks. For ECE, we find minor improvements with growing drop rate which might be explained by the fact that the L sub-networks in a given optimization step overlap less for higher p -values, thus allowing them to approximate the actual data distribution more closely. We choose $p = 0.1$ as the complexity of the resulting sub-networks is only mildly reduced compared to the deterministic full network.

Studying the impact of sample size $L = 4, 5, 8, 10, 20$, we find RMSE (see top panel of Fig. 20) to be largely stable w.r.t. this parameter. For ECE (see bottom panel of Fig. 20, train scores grow with L , indicating a certain over-estimation of the present aleatoric uncertainties. This artefact is not generalized to test data though, where we observe broadly similar mean values and 75% quantiles. Under data shift, certain fluctuations of ECE occur as sample size L changes, however there is no clear trend. We thus choose the rather small $L = 5$ to keep the computational overhead down.

Appendix C In-depth investigation of uncertainty measures

In the following, we employ the Kolmogorov-Smirnov distance as a supplementary uncertainty score and compare it with expected calibration error (ECE) and Wasserstein distance (WS). Finally, limitations of negative log-likelihood (NLL) for uncertainty quantification are discussed.

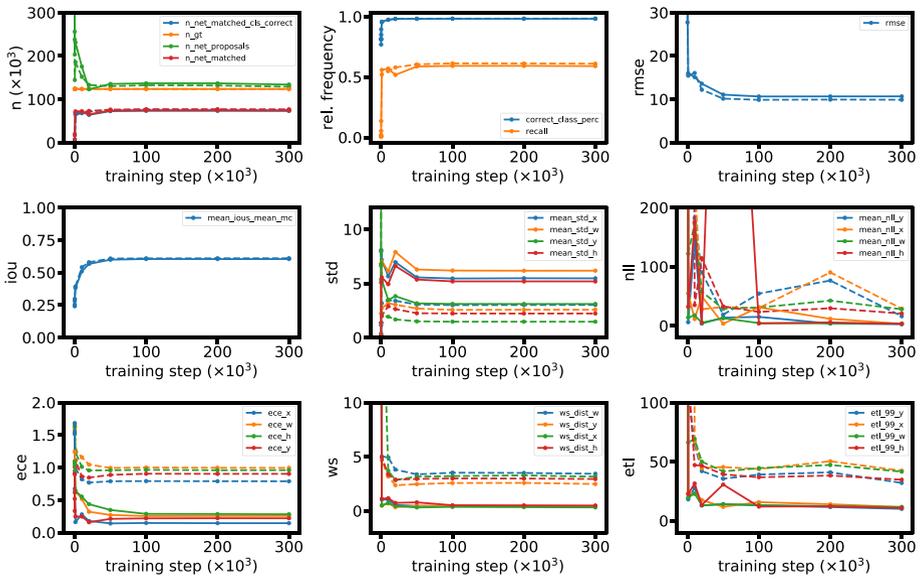


Fig. 18 Various test statistics of W-SqueezeDet (solid lines) and MC-SqueezeDet (dashed lines) during model optimization on the BDD100k dataset. We consider performance scores (recall, RMSE, IoU, see first and second row) and uncertainty measures (NLL, ECE, WS, ETL, see second and third row). W-SqueezeDet yields comparable task performance while providing clearly better uncertainty estimates

C.1 Dependencies between uncertainty measures

Extending the analysis of empirically observed dependencies between WS and ECE in Fig. 3, we additionally consider Kolmogorov-Smirnov (KS) distances (Stephens, 1974) in Fig. 21 (middle and bottom panel). These KS-distances are calculated between samples of normalized residuals and a standard Gaussian. Different from the Wasserstein distance, the KS-distance is not transport-based but determined by the largest distance between the empirical CDFs of the two samples. It is therefore bounded to [0, 1] and unable to resolve differences between two samples that both strongly deviate from a standard Gaussian. Again, we find the dependencies between these measures to clearly deviate from ideal correlation.

The data splits in Figs. 3 and 21 are color-coded as follows: train is green, test is blue, PCA-interpolate is green-yellow, PCA-extrapolate is orange-yellow, label-interpolate is red and label-extrapolate is light red. The mapping between uncertainty methods and plot markers reads: SWAG is ‘triangle’, MC is ‘diamond’, MC-LL is ‘thin diamond’, DE is ‘cross’, PU is ‘point’, PU-DE is ‘star’, PU-MC is ‘circle’, PU-EV is ‘pentagon’ and W-dropout is ‘plus’. The data base of this visualization are the 14 standard regression datasets. Some Wasserstein distances lie above the x-axis cut-off and are thus not visualized.

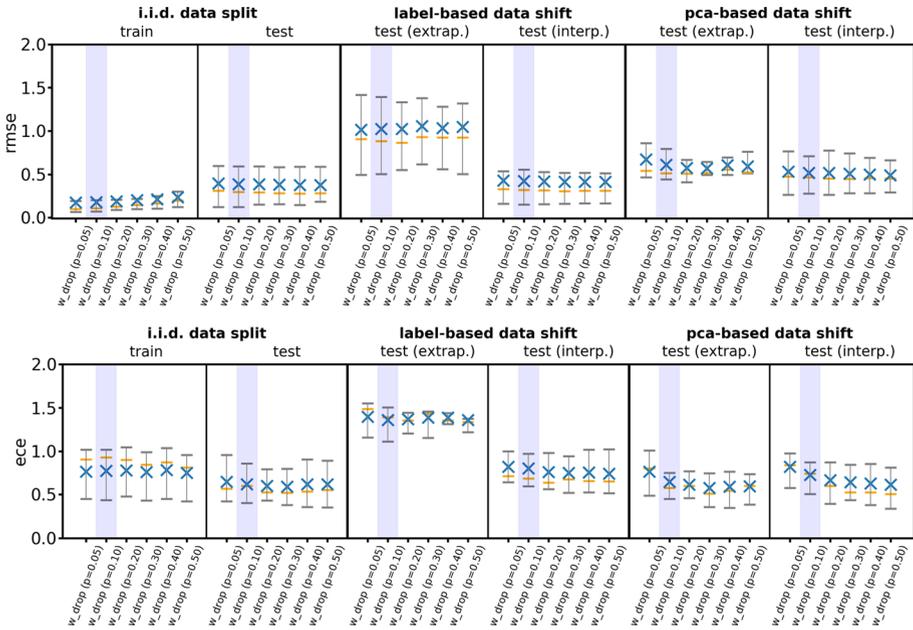


Fig. 19 Dependence of Wasserstein dropout on drop rate p . Root-mean-square errors (RMSEs) (\downarrow , top row) and expected calibration errors (ECEs) (\downarrow , bottom row) are shown for neuron drop rates of $p = 0.05, 0.1, 0.2, 0.3, 0.4, 0.5$ under i.i.d. conditions (first and second panel in each row) and under various kinds of data shift (third to sixth panel in each row, see text for details). W-dropout with $p = 0.1$ (used for evaluations on toy and 1D regression data) is highlighted by a light blue background. Each blue cross is the mean over 10 standard regression datasets. Orange line markers indicate median values. The gray vertical bars reach from the 25% quantile (bottom horizontal line) to the 75% quantile (top horizontal line) (Color figure online)

C.2 Discussion of NLL as a measure of uncertainty

Typically, DNNs using uncertainty are often evaluated in terms of their negative log-likelihood (NLL). This property is affected not only by the uncertainty, but also by the DNNs performance. Additionally, it is difficult to interpret, sometimes leading to counterintuitive results, which we want to elaborate on here. As a first example, take the likelihood of two datasets $x_1 = \{0\}$ and $x_2 = \{0.5\}$, each consisting of a single point, with respect to a normal distribution $\mathcal{N}(0, 1)$. Naturally, we find x_1 to be located at the maximum of the considered normal distribution and deem it the more likely candidate. But, if we extend these datasets to more than single points, i.e., $\tilde{x}_1 = \{0, 0.1, 0, -0.1, 0\}$ and $\tilde{x}_2 = \{0.5, -0.4, 0, -1.9, -0.7\}$, it becomes obvious that \tilde{x}_2 is much more likely to follow the intended Gaussian distribution. Nonetheless, $NLL(\tilde{x}_2) \approx 1.4 > 0.9 \approx NLL(\tilde{x}_1)$, where

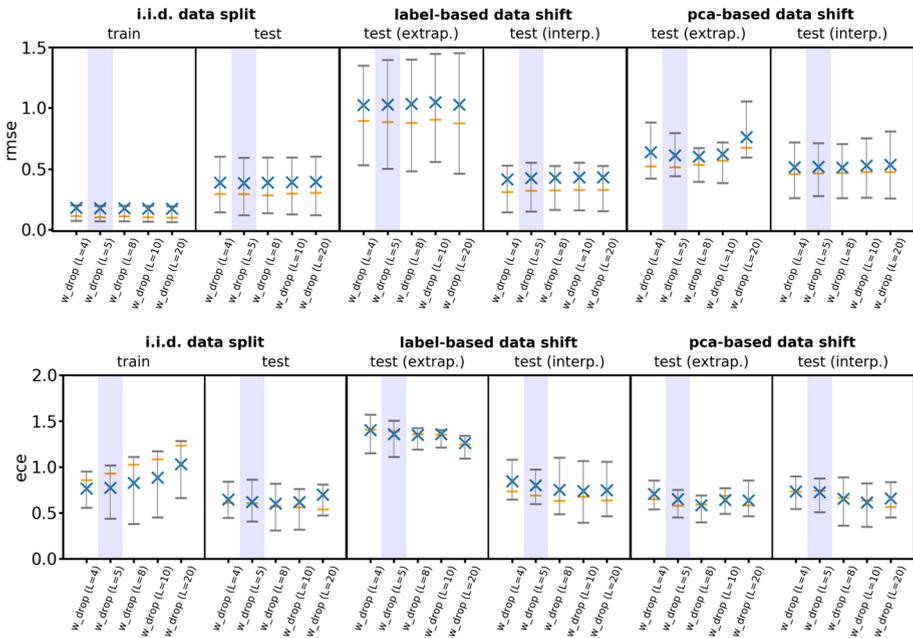


Fig. 20 Dependence of Wasserstein dropout on sample size L . Root-mean-square errors (RMSEs) (\downarrow , top row) and expected calibration errors (ECEs) (\downarrow , bottom row) are shown for sample sizes of $L = 4, 5, 8, 10, 20$ under i.i.d. conditions (first and second panel in each row) and under various kinds of data shift (third to sixth panel in each row, see text for details). W-dropout with $L = 5$ (used throughout the rest of the paper) is highlighted by a light blue background. Each blue cross is the mean over 10 standard regression datasets. Orange line markers indicate median values. The gray vertical bars reach from the 25% quantile (bottom horizontal line) to the 75% quantile (top horizontal line) (Color figure online)

$$\text{NLL}(y) := \log \sqrt{2\pi\sigma^2} + \frac{1}{N} \sum_{i=1}^N \frac{(y_i - \mu)^2}{2\sigma^2}. \tag{C1}$$

This may be seen as a direct consequence of the point-wise definition of NLL, which does not consider the distribution of the elements in \hat{x}_i . From this observation also follows that a model with high prediction accuracy will have a lower NLL score as a worse performing one if uncertainties are predicted in the same way. Independent of whether those reflected the “true” uncertainty in either case. This issue can be further substantiated on a second example. Consider two other datasets z_1, z_2 drawn i.i.d. from Gaussian distributions $\mathcal{N}(0, \sigma_i)$ with two differing values $\sigma_1 < \sigma_2$. If we determine the NLL of each with respect to its own distribution the offset term in Eq. (C1) leads to $\text{NLL}(z_2) = \text{NLL}(z_1) + \log(\sigma_2/\sigma_1)$ with $\log(\sigma_2/\sigma_1) > 0$. Although both accurately reflect their own distributions, or uncertainties so to speak, the narrower z_1 is more “likely”. This offset makes it difficult to assess reported NLL values for systems with heteroscedastic uncertainty. While smaller is typically “better”, it is highly data- (and prediction-)dependent which value is good in the sense of a reasonable correlation between performance and uncertainty.

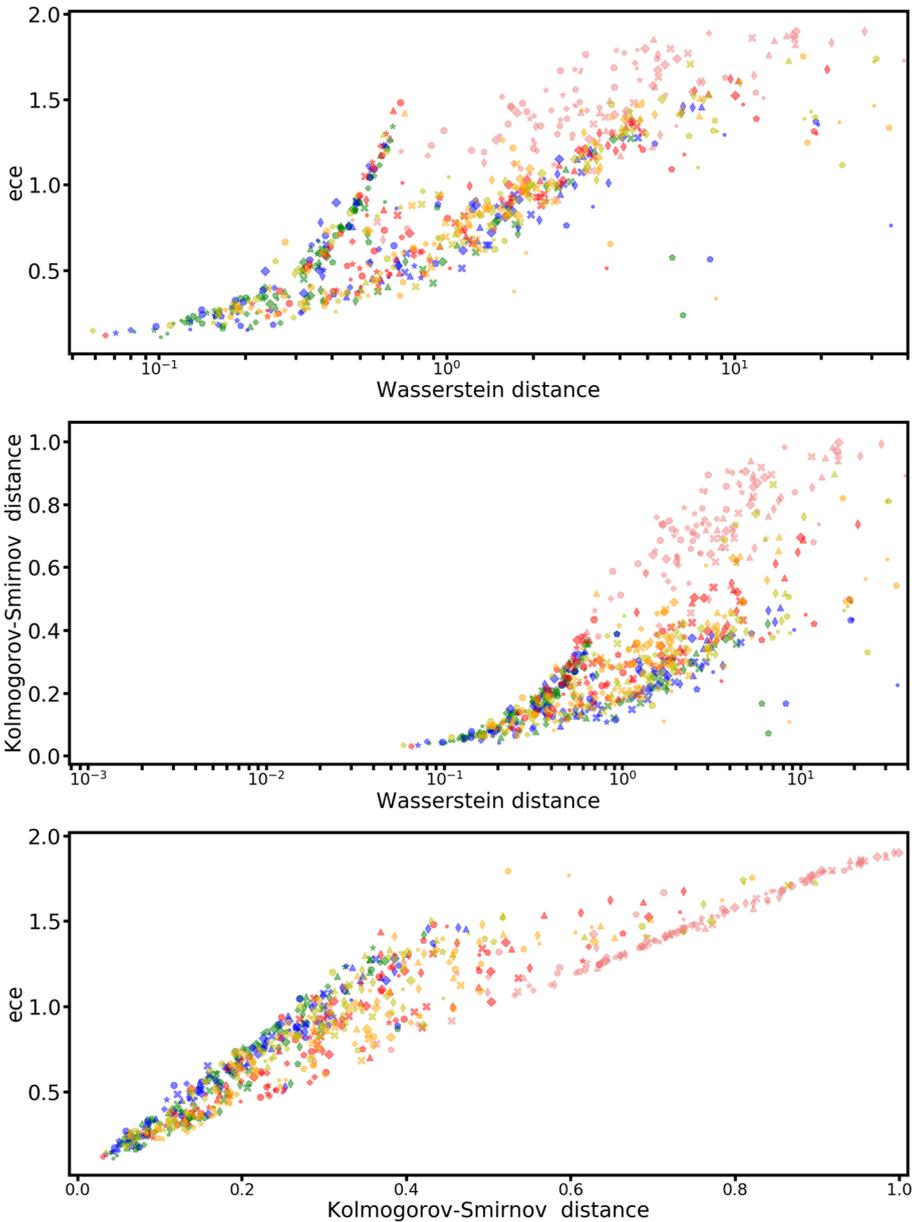


Fig. 21 Dependencies between the three uncertainty measures ECE, Wasserstein distance and Kolmogorov-Smirnov distance. Uncertainty methods are encoded via plot markers, data splits via color. Datasets are not encoded and cannot be distinguished (see text for more details). Each plot point corresponds to a cross-validated trained network. The clearly visible deviations from ideal correlations point at the potential of these uncertainty measures to complement one another

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by J. Sicking and M. Pintz. The first draft of the manuscript was written by J. Sicking, M. Akila and M. Pintz and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. The research of J. Sicking and M. Akila was funded by the German Federal Ministry for Economic Affairs and Energy within the project “KI Absicherung – Safe AI for Automated Driving”. Said authors would like to thank the consortium for the successful cooperation. The work of T. Wirtz was funded by the German Federal Ministry of Education and Research, ML2R - No. 01S18038B. S. Wrobel contributed as part of the Fraunhofer Center for Machine Learning within the Fraunhofer Cluster for Cognitive Internet Technologies. The work of A. Fischer was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC-2092 CaSA – 390781972.

Availability of data and materials All datasets used in this work and the SqueezeDet network architecture are publicly available. For more information, see the respective references in the text.

Code availability The code base for Wasserstein dropout can be found on <https://github.com/fraunhofer-iaisi/wasserstein-dropout>.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Consent for publication All authors agree with the content of this work and gave their explicit consent to this submission.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Amini, A., Schwarting, W., Soleimany, A., et al. (2020). Deep evidential regression. In H. Larochelle, M. Ranzato, R. Hadsell, et al. (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 14927–14937). New York: Curran Associates, Inc.
- Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In D. Precup, & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning, proceedings of machine learning research, PMLR* (Vol. 70, pp. 214–223). <https://proceedings.mlr.press/v70/arjovsky17a.html>.
- Bhattacharyya, A., Fritz, M., & Schiele, B. (2018). Long-term on-board prediction of people in traffic scenes under uncertainty. In *2018 IEEE conference on computer vision and pattern recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018* (pp. 4194–4202). Computer Vision Foundation/IEEE Computer Society. <https://doi.org/10.1109/CVPR.2018.00441>. http://openaccess.thecvf.com/content_cvpr_2018/html/Bhattacharyya_Long-Term_On-Board_Prediction_CVPR_2018_paper.html.
- Bishop, C. M. (2006). *Pattern recognition and machine learning (information science and statistics)*. Berlin: Springer.
- Blei, D. M., & Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, *1*(1), 121–143. <https://doi.org/10.1214/06-BA104>
- Blundell, C., Cornebise, J., Kavukcuoglu, K., et al. (2015). Weight uncertainty in neural network. In F. Bach, & D. Blei (Eds.), *Proceedings of the 32nd international conference on machine learning, proceedings*

- of machine learning research, PMLR, Lille, France (Vol. 37, pp. 1613–1622). <https://proceedings.mlr.press/v37/blundell15.html>.
- Bojarski, M., Del Testa, D., Dworakowski, D., et al. (2016). *End to end learning for self-driving cars*. arXiv preprint [arXiv:1604.07316](https://arxiv.org/abs/1604.07316).
- Caesar, H., Bankiti, V., Lang, A. H., et al. (2020). nuScenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020* (pp. 11,618–11,628). Computer Vision Foundation/IEEE. <https://doi.org/10.1109/CVPR42600.2020.01164>. https://openaccess.thecvf.com/content_CVPR_2020/html/Caesar_nuScenes_A_Multimodal_Dataset_for_Autonomous_Driving_CVPR_2020_paper.html.
- Campello, R. J. G. B., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In J. Pei, V. S. Tseng, L. Cao, et al. (Eds.), *Advances in knowledge discovery and data mining, 17th Pacific-Asia conference, PAKDD 2013, Gold Coast, Australia, April 14–17, 2013, proceedings, part II, lecture notes in computer science* (Vol. 7819, pp. 160–172). Springer. https://doi.org/10.1007/978-3-642-37456-2_14.
- Dekking, F. M., Kraaikamp, C., Lopuhaä, H. P., et al. (2005). *A modern introduction to probability and statistics: Understanding why and how*. Berlin: Springer.
- Duam, D., & Graff, C. (2017). *UCI machine learning repository*. <http://archive.ics.uci.edu/ml>.
- Fan, X., Zhang, S., Tanwisuth, K., et al. (2021). Contextual dropout: An efficient sample-dependent dropout module. In *9th international conference on learning representations, ICLR 2021, virtual event, Austria, May 3–7, 2021*. OpenReview.net. https://openreview.net/forum?id=ct8_a9h1M.
- Feng, D., Harakeh, A., Waslander, S. L., et al. (2020). *A review and comparative study on probabilistic object detection in autonomous driving*. CoRR [arXiv:2011.10671](https://arxiv.org/abs/2011.10671).
- Foong, A. Y., Li, Y., Hernández-Lobato, J. M., et al. (2019). ‘In-between’ uncertainty in Bayesian neural networks. In *ICML workshop on uncertainty and robustness in deep learning*.
- Fort, S., Hu, H., & Lakshminarayanan, B. (2019). Deep ensembles: A loss landscape perspective. In *NeurIPS workshop on Bayesian deep learning*.
- Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In M. F. Balcan, & K. Q. Weinberger (Eds.), *Proceedings of The 33rd international conference on machine learning, proceedings of machine learning research, PMLR, New York, New York, USA* (Vol. 48, pp. 1050–1059). <http://proceedings.mlr.press/v48/gal16.html>.
- Gal, Y., Hron, J., & Kendall, A. (2017). Concrete dropout. In I. Guyon, U. von Luxburg, S. Bengio, et al. (Eds.), *Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017, December 4–9, 2017, Long Beach, CA, USA* (pp. 3581–3590). <https://proceedings.neurips.cc/paper/2017/hash/84ddfb34126fc3a48ee38d7044e87276-Abstract.html>.
- Garnelo, M., Schwarz, J., Rosenbaum, D., et al. (2018). Neural processes. In *ICML workshop on theoretical foundations and applications of deep generative models*.
- Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition, Providence, RI, USA, June 16–21, 2012* (pp. 3354–3361). IEEE Computer Society. <https://doi.org/10.1109/CVPR.2012.6248074>.
- Geyer, J., Kassahun, Y., Mahmudi, M., et al. (2020). A2D2: Audi autonomous driving dataset. <https://www.a2d2.audi>. [arXiv:2004.06320](https://arxiv.org/abs/2004.06320).
- Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In G. Gordon, D. Dunson, & M. Dudík (Eds.), *Proceedings of machine learning research, vol 15. JMLR workshop and conference proceedings, Fort Lauderdale, FL, USA* (pp. 315–323).
- Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477), 359–378. <https://doi.org/10.1198/016214506000001437>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge: MIT Press.
- Gustafsson, F. K., Danelljan, M., & Schön, T. B. (2020). Evaluating scalable Bayesian deep learning methods for robust computer vision. In *2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR workshops 2020, Seattle, WA, USA, June 14–19, 2020* (pp. 1289–1298). Computer Vision Foundation/IEEE. <https://doi.org/10.1109/CVPRW50498.2020.00167>. https://openaccess.thecvf.com/content_CVPRW_2020/html/w20/Gustafsson_Evaluating_Scalable_Bayesian_Deep_Learning_Methods_for_Robust_Computer_Vision_CVPRW_2020_paper.html.
- Hall, D., Dayoub, F., Skinner, J., et al. (2020). Probabilistic object detection: Definition and evaluation. In *IEEE D. WACV 2020, Snowmass Village, CO, USA, March 1–5, 2020* (pp. 1020–1029). IEEE. <https://doi.org/10.1109/WACV45572.2020.9093599>.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Berlin: Springer.

- He, Y., Zhu, C., Wang, J., et al. (2019). Bounding box regression with uncertainty for accurate object detection. In *IEEE conference on computer vision and pattern recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019* (pp. 2888–2897). Computer Vision Foundation/IEEE. <https://doi.org/10.1109/CVPR.2019.00300>. http://openaccess.thecvf.com/content_CVPR_2019/html/He_Bounding_Box_Regression_With_Uncertainty_for_Accurate_Object_Detection_CVPR_2019_paper.html.
- Heskes, T. (1996). Practical confidence and prediction intervals. In M. Mozer, M. I. Jordan, T. Petsche (Eds.), *Advances in neural information processing systems 9, NIPS, Denver, CO, USA, December 2–5, 1996* (pp. 176–182). MIT Press. <http://papers.nips.cc/paper/1306-practical-confidence-and-prediction-intervals>.
- Ilg, E., Çiçek, Ö., Galesso, S., et al. (2018). Uncertainty estimates and multi-hypotheses networks for optical flow. In V. Ferrari, M. Hebert, C. Sminchisescu, et al. (Eds.), *Computer vision—ECCV 2018—15th European conference, Munich, Germany, September 8–14, 2018, proceedings, part VII, lecture notes in computer science* (Vol. 11211, pp. 677–693). Springer. https://doi.org/10.1007/978-3-030-01234-2_40.
- Iwata, T., & Ghahramani, Z. (2017). *Improving output uncertainty estimation and generalization in deep learning via neural network Gaussian processes*. arXiv preprint [arXiv:1707.05922](https://arxiv.org/abs/1707.05922).
- Kendall, A., & Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? In I. Guyon, U. von Luxburg, S. Bengio, et al. (Eds.), *Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017, December 4–9, 2017, Long Beach, CA, USA* (pp. 5574–5584). <https://proceedings.neurips.cc/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html>.
- Khosravi, A., Nahavandi, S., Creighton, D. C., et al. (2011). Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Transaction on Neural Networks*, 22(9), 1341–1356. <https://doi.org/10.1109/TNN.2011.2162110>.
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio, & Y. LeCun (Eds.), *3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, conference track proceedings*. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- Kuleshov, V., Fenner, N., & Ermon, S. (2018). Accurate uncertainties for deep learning using calibrated regression. In J. Dy, & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning, proceedings of machine learning research, PMLR* (Vol. 80, pp. 2796–2804). <http://proceedings.mlr.press/v80/kuleshov18a.html>.
- Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. von Luxburg, S. Bengio, et al. (Eds.), *Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017, December 4–9, 2017, Long Beach, CA, USA* (pp. 6402–6413). <https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html>.
- Landau, H. (1967). Sampling, data transmission, and the Nyquist rate. *Proceedings of the IEEE*, 55(10), 1701–1706. <https://doi.org/10.1109/PROC.1967.5962>
- Liu, S., Liu, S., Cai, W., et al. (2014). Early diagnosis of Alzheimer’s disease with deep learning. In *IEEE 11th international symposium on biomedical imaging, ISBI 2014, April 29–May 2, 2014, Beijing, China* (pp. 1015–1018). IEEE. <https://doi.org/10.1109/ISBI.2014.6868045>.
- Maddox, W. J., Izmailov, P., Garipov, T., et al. (2019). A simple baseline for Bayesian uncertainty in deep learning. In H. Wallach, H. Larochelle, A. Beygelzimer, et al. (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2019/file/118921efba23fc329e6560b27861f0c2-Paper.pdf>.
- Malinin, A., & Gales, M. J. F. (2018). Predictive uncertainty estimation via prior networks. In S. Bengio, H. M. Wallach, H. Larochelle, et al. (Eds.), *Advances in neural information processing systems 31: Annual conference on neural information processing systems 2018, NeurIPS 2018, December 3–8, 2018, Montréal, Canada* (pp. 7047–7058). <https://proceedings.neurips.cc/paper/2018/hash/3ea2db5062ceefceaf70a9d9a56a6f4-Abstract.html>.
- Miller, D., Nicholson, L., Dayoub, F., et al. (2018). Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE international conference on robotics and automation, ICRA 2018, Brisbane, Australia, May 21–25, 2018* (pp. 1–7). IEEE. <https://doi.org/10.1109/ICRA.2018.8460700>.
- Naeni, M. P., Cooper, G. F., & Hauskrecht, M. (2015). Obtaining well calibrated probabilities using Bayesian binning. In B. Bonet, & S. Koenig (Eds.), *Proceedings of the twenty-ninth AAAI conference on artificial intelligence, January 25–30, 2015, Austin, Texas, USA* (pp. 2901–2907). AAAI Press. <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9667>.
- Neumann, L., Karg, M., Zhang, S., et al. (2018). Nightowls: A pedestrians at night dataset. In C. V. Jawahar, H. Li, G. Mori, et al (Eds.), *Computer vision—ACCV 2018—14th Asian conference on computer*

- vision, Perth, Australia, December 2–6, 2018, revised selected papers, part I, lecture notes in computer science (Vol. 11361, pp. 691–705). Springer. https://doi.org/10.1007/978-3-030-20887-5_43.
- Nix, D., & Weigend, A. (1994). Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE international conference on neural networks (ICNN'94)* (Vol. 1, pp. 55–60). <https://doi.org/10.1109/ICNN.1994.374138>.
- Pomerleau, D. (1988). ALVINN: An autonomous land vehicle in a neural network. In D. S. Touretzky (Ed.), *Advances in neural information processing systems 1, [NIPS conference, Denver, Colorado, USA, 1988]* (pp. 305–313). Morgan Kaufmann. <http://papers.nips.cc/paper/95-alvinn-an-autonomous-land-vehicle-in-a-neural-network>.
- Postels, J., Ferroni, F., Coskun, H., et al. (2019). Sampling-free epistemic uncertainty estimation using approximated variance propagation. In *2019 IEEE/CVF international conference on computer vision, ICCV 2019, Seoul, Korea (South), October 27–November 2, 2019* (pp. 2931–2940). IEEE. <https://doi.org/10.1109/ICCV.2019.00302>.
- Qiu, X., Meyerson, E., & Mäkiöläinen, R. (2020). Quantifying point-prediction uncertainty in neural networks via residual estimation with an I/O kernel. In *8th international conference on learning representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*. OpenReview.net. <https://openreview.net/forum?id=rkxNh1Stvr>.
- Ritter, H., Botev, A., & Barber, D. (2018). A scalable Laplace approximation for neural networks. In *6th international conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, 2018, conference track proceedings*. OpenReview.net. <https://openreview.net/forum?id=Skdvd2xAZ>.
- Rockafellar, R., & Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7), 1443–1471. [https://doi.org/10.1016/S0378-4266\(02\)00271-6](https://doi.org/10.1016/S0378-4266(02)00271-6).
- Rubner, Y., Tomasi, C., Guibas, L. J. (1998). A metric for distributions with applications to image databases. In *Proceedings of the sixth international conference on computer vision (ICCV-98), Bombay, India, January 4–7, 1998* (pp. 59–66). IEEE Computer Society. <https://doi.org/10.1109/ICCV.1998.710701>.
- Sicking, J., Akila, M., Wirtz, T., et al. (2020). Characteristics of Monte Carlo dropout in wide neural networks. In *ICML 2020 workshop on uncertainty and robustness in deep learning*. [arXiv:2007.05434](https://arxiv.org/abs/2007.05434).
- Snoek, J., Ovadia, Y., Fertig, E., et al. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In H. M. Wallach, H. Larochelle, A. Beygelzimer, et al. (Eds.), *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada* (pp. 13,969–13,980). <https://proceedings.neurips.cc/paper/2019/hash/8558cb408c1d76621371888657d2eb1d-AbstrAct.html>.
- Srivastava, N., Hinton, G., Krizhevsky, A., et al. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56), 1929–1958.
- Stephens, M. A. (1974). EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347), 730–737.
- Villani, C. (2008). *Optimal transport: Old and new* (Vol. 338). Berlin: Springer.
- Walker, J., Doersch, C., Gupta, A., et al. (2016). An uncertain future: Forecasting from static images using variational autoencoders. In *European conference on computer vision* (pp. 835–851). Springer.
- Wilson, A. G., Hu, Z., Salakhutdinov, R., et al. (2016). Deep kernel learning. In A. Gretton, & C. C. Robert (Eds.), *Proceedings of the 19th international conference on artificial intelligence and statistics, AISTATS 2016, Cadiz, Spain, May 9–11, 2016, JMLR workshop and conference proceedings*. JMLR.org (Vol. 51, pp. 370–378). <http://proceedings.mlr.press/v51/wilson16.html>.
- Wirges, S., Reith-Braun, M., Lauer, M., et al. (2019). Capturing object detection uncertainty in multi-layer grid maps. In *2019 IEEE intelligent vehicles symposium, IV 2019, Paris, France, June 9–12, 2019* (pp. 1520–1526). IEEE. <https://doi.org/10.1109/IVS.2019.8814073>.
- Wrenninge, M., & Unger, J. (2018). *Synscapes: A photorealistic synthetic dataset for street scene parsing*. CoRR [arXiv:1810.08705](https://arxiv.org/abs/1810.08705).
- Wu, B., Iandola, F. N., Jin, P. H., et al. (2017). SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *2017 IEEE conference on computer vision and pattern recognition workshops, CVPR workshops 2017, Honolulu, HI, USA, July 21–26, 2017* (pp. 446–454). IEEE Computer Society. <https://doi.org/10.1109/CVPRW.2017.60>.
- Yu, F., Chen, H., Wang, X., et al. (2020). BDD100K: A diverse driving dataset for heterogeneous multi-task learning. In *2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020* (pp. 2633–2642). Computer Vision Foundation/IEEE. <https://doi.org/10.1109/CVPR42600.2020.00271>. https://openaccess.thecvf.com/content_CVPR_2020/html

[Yu_BDD100K_A_Diverse_Driving_Dataset_for_Heterogeneous_Multitask_Learning_CVPR_2020_paper.html](#).

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

Authors and Affiliations

Joachim Sicking¹  · Maram Akila¹ · Maximilian Pintz¹ · Tim Wirtz¹ · Stefan Wrobel^{1,2,3} · Asja Fischer⁴

Maram Akila
maram.akila@iaais.fraunhofer.de

Maximilian Pintz
maximilian.alexander.pintz@iaais.fraunhofer.de

Tim Wirtz
tim.wirtz@iaais.fraunhofer.de

Stefan Wrobel
stefan.wrobel@cs.uni-bonn.de

Asja Fischer
asja.fischer@ruhr-uni-bochum.de

¹ Fraunhofer IAIS, Schloss Birlinghoven 1, 53757 Sankt Augustin, Germany

² Fraunhofer Center for Machine Learning, Schloss Birlinghoven 1, 53757 Sankt Augustin, Germany

³ University of Bonn, Regina-Pacis-Weg 3, 53113 Bonn, Germany

⁴ Ruhr University Bochum, Universitätsstraße 150, 44801 Bochum, Germany