Prediction Intervals: Split Normal Mixture from Quality-Driven Deep Ensembles (with Supplementary Material)

Tárik S. Salem
tariks@alumni.ntnu.no

Helge Langseth
helgel@ntnu.no

Heri Ramampiaro
heri@ntnu.no

Norwegian University of Science and Technology (NTNU)

Abstract

Prediction intervals are a machine- and human-interpretable way to represent predictive uncertainty in a regression analysis. In this paper, we present a method for generating prediction intervals along with point estimates from an ensemble of neural networks. We propose a multi-objective loss function fusing quality measures related to prediction intervals and point estimates, and a penalty function, which enforces semantic integrity of the results and stabilizes the training process of the neural networks. The ensembled prediction intervals are aggregated as a split normal mixture accounting for possible multimodality and asymmetricity of the posterior predictive distribution, and resulting in prediction intervals that capture aleatoric and epistemic uncertainty. Our results show that both our quality-driven loss function and our aggregation method contribute to well-calibrated prediction intervals and point estimates.

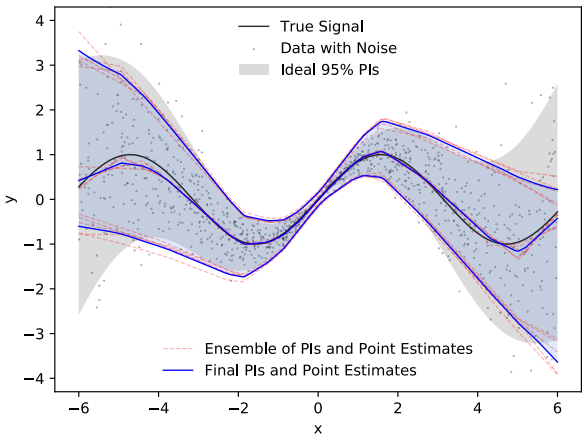


Figure 1: A toy example demonstrating our method for generating PIs from a split normal mixture of quality-driven deep ensembles. The final PIs are accounting for aleatoric and epistemic uncertainty. The synthetic dataset is a sinusoid with Gaussian noise and sparsified samples. The magnitude of Gaussian noise and sparsification is increasing with distance from the center ($x = 0$).

1 INTRODUCTION

Quantifying predictive uncertainty of machine learning models is crucial in applications, e.g., mission-critical systems, where it is essential to know when the model is not able to provide accurate predictions. In decision support systems, providing the human with an additional information about the uncertainty of a prediction may decrease the response time and increase the accuracy of an action. It can also positively contribute in building trust and understanding towards the machine learning system and its correct facilitation. In this work, we focus on the quantification of predictive uncertainty for the regression task, specifically in the form of prediction intervals which have probabilistic interpretation and which are interpretable for both humans and machines.

Prediction interval (PI) is an estimate representing predictive uncertainty in the form of two values between which a future observation will fall with a certain probability. Well-calibrated/high-quality prediction intervals are as narrow as possible while attaining the desired coverage probability.

While neural networks (NNs) are powerful function approximators, they are poor at representing predictive uncertainty. Previously, methods adopting a Bayesian approach in the context of neural networks (e.g., Graves, 2011; Blundell et al., 2015; Hernandez-Lobato and Adams, 2015; Krueger et al., 2017; Louizos and Welling, 2017; Pawlowski et al., 2017; Wu et al., 2019; Izmailov et al., 2019) represented the state-of-the-art for providing

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

predictive uncertainty estimates. See Yao et al. (2019) for an overview. Recently, a number of non-Bayesian yet probabilistic approaches provide competitive predictive uncertainty estimates (Lakshminarayanan et al., 2017; Pearce et al., 2018; Tagasovska and Lopez-Paz, 2019). In this paper, we focus on and extend the latter branch of research.

Our work builds on the findings of Pearce et al. (2018) which is inspired by works of Khosravi et al. (2011) and Lakshminarayanan et al. (2017). Pearce et al. (2018) presented a method based on aggregating ensembled PIs from a set of NNs optimized with respect to a so-called quality-driven loss function (see Section 2.4).

Although the aforementioned work has achieved promising results of well-calibrated PIs, it has several limitations that need to be addressed. In our continuation, we primarily address three main limitations of the state-of-the-art method: (1) The inability to generate point estimates, (2) the theoretically weakly justified method for aggregation of ensembled PIs, and (3) the fragile training process. With this in mind, the main contributions of this work can be summarized as follows:

- We retrofit the quality-driven loss function:
  - With a point estimate loss (particularly MSE or mean squared error) to be able to draw point estimates and PIs from the same generative distribution.
  - With a penalty function adding a constraint and thus enforcing the integrity of the results, i.e. avoiding crossing of PI boundaries or point estimates out of PI bounds, and consequently stabilizing/strengthening the training process.
- We propose a new aggregation method for ensembles of PIs with point estimates. It is fitting a split normal mixture (Wallis, 2014) providing tighter and theoretically well-founded aggregates of PIs.
- We propose an analytical approach to parameter initialization for the parameter fitting process of a split normal probability density function, thus increasing the success of fitting and accelerating it compared to random initialization.

In addition to the above, we provide important insights and guidelines for hyper-parameter search contributing to reproducibility and reliable model fitting, and we suggest directions for future research.

2 BACKGROUND

In this section, we formally introduce the predictive uncertainty and prediction intervals, we consider methods for quantifying predictive uncertainty in NNs, especially in the form of prediction intervals, and we provide insights into the work of Pearce et al. (2018) which we build on.

2.1 Predictive Uncertainty

The sources of predictive uncertainty can be categorized into aleatoric and epistemic uncertainty (Kiureghian and Ditlevsen, 2009).

The aleatoric or aleatory or data uncertainty is also known as the irreducible uncertainty, i.e. it can not be reduced either through model or data. It arises from an inherent and irreducible data noise. The aleatoric uncertainty can be captured by learning the conditional distribution between the target and the input variables. The aleatoric uncertainty can further be characterized as homoskedastic (homoscedastic) if the irreducible noise is constant across random variables or as heteroskedastic (heteroscedastic) if contrary.

In contrast to aleatoric uncertainty, the epistemic uncertainty is reducible, and it can be further decomposed into model uncertainty and distributional uncertainty. Model uncertainty can be caused by model bias or parameter uncertainty due to insufficient data. Distributional uncertainty can be caused by the mismatch between training and test set (Malinin and Gales, 2018), and it is often described as a part of the model uncertainty.

Generally, the aim is to reduce epistemic uncertainties. However, constraints emerging from model or data usually do not allow it.

Please note that predictive uncertainty is a broad concept, and the literature is inconsistent in the terminology. We follow the taxonomy found in (Kiureghian and Ditlevsen, 2009; Pearce et al., 2018; Malinin and Gales, 2018).

2.2 Prediction Intervals

Given an input $\mathbf{x}^{(i)}$, a prediction interval $[\hat{y}_L^{(i)}, \hat{y}_U^{(i)}]$ of a sample i captures the future observation (target variable) $y^{(i)}$ with the probability equal or greater than $\gamma \in [0, 1]$ (eq. 1). The value of γ is commonly set to 0.95 or 0.99. Common in the literature is an alternative notation with α.

$$\Pr\left(\hat{y}_L^{(i)} \leq y^{(i)} \leq \hat{y}_U^{(i)}\right) \geq \gamma = (1 - \alpha) \qquad (1)$$

Given n samples, the quality of the generated prediction intervals is assessed by measuring the prediction interval

coverage probability (PICP)

$$PICP = \frac{c}{n}$$

where

$$c = \sum_{i=1}^{n} k_i \tag{2}$$

for

$$k_i = \begin{cases} 1 & \text{if } \hat{y}_L^{(i)} \leq y^{(i)} \leq \hat{y}_U^{(i)}, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

and by measuring the mean prediction interval width (MPIW)

$$MPIW = \frac{1}{n} \sum_{i=1}^{n} \hat{y}_U^{(i)} - \hat{y}_L^{(i)}$$

or its normalized version NMPIW

$$NMPIW = \frac{MPIW}{r} \tag{4}$$

where $r = max(y) - min(y)$. Please note that MPIW assumes $\hat{y}_U^{(i)} \geq \hat{y}_L^{(i)}$, i.e. no crossing of PI bounds. Ignoring this fact can lead to misleading results.

It is desired to achieve $PICP \geq \gamma$ while having MPIW as small as possible.

2.3 Related Work

Following is a concise exploration of methods based on NNs capable of quantifying predictive uncertainty in a regression analysis. Methods adopting the Bayesian approach (such as BNN or Bayesian neural networks) are a prominent group (Graves, 2011; Blundell et al., 2015; Hernandez-Lobato and Adams, 2015; Krueger et al., 2017; Louizos and Welling, 2017; Pawlowski et al., 2017; Wu et al., 2019; Izmailov et al., 2019; etc.) and represented for long the state-of-the-art to estimate predictive uncertainty. We explore alternative non-Bayesian yet probabilistic methods representing competitive and possibly a state-of-the-art alternative.

Interpreting Monte Carlo dropout (MC-dropout) at test time as approximate Bayesian inference (Gal and Ghahramani, 2016) has been a widely used method to quantify predictive uncertainty, mainly due to its scalability and simplicity. Interpretation of MC-dropout as an ensemble model combination motivated consecutive research on ensemble-based methods which have been shown to be superior in generating predictive uncertainty to MC-dropout or even Bayesian methods (Lakshminarayanan et al., 2017). Recently, it has been shown that deep ensembles with random initialization may explore different modes in function space, and therefore perform well in exploring model uncertainty. That is in contrast to subspace sampling methods (e.g. MC-dropout, weight averaging) which may generate a set of diverse functions but still in the vicinity of the starting point and thus generating insufficiently diverse predictions (Fort et al., 2019). Also recently, the robustness of uncertainty quantification methods under dataset shift was investigated, and although all compared methods have been deceptible to increasing dataset shifts, deep ensembles have shown the greatest robustness (Ovadia et al., 2019).

More recently, an alternative line of work employs a loss function to optimize for generating specifically prediction intervals. A method based on quantile regression is utilizing the so-called pinball loss function (Koenker, 2005) to optimize a single NN for the desired PIs and so-called Orthonormal Certificates to account for epistemic uncertainty (Tagasovska and Lopez-Paz, 2019). A method called LUBE (Khosravi et al., 2011) constructs prediction intervals using quality-driven loss function utilizing quality metrics PICP and NMPIW. LUBE applies simulated annealing for training a NN with respect to the quality-driven loss function. Inspired by Khosravi et al. (2011) and Lakshminarayanan et al. (2017), a slightly modified quality-driven loss function optimized for gradient descent was proposed (Pearce et al., 2018) and extended with ensembles to account for epistemic uncertainty. Although presenting favorable results, both methods deliver only prediction intervals without point estimates.

2.4 Quality-Driven Ensembles by Pearce et al. (2018)

Our work builds on (Pearce et al., 2018) henceforth referred to as the original quality-driven ensembles or shortly original QDE (in results annotated as SEM-QD). First, the quality-driven loss function is described (annotated as QD). Second, the aggregation method is described (annotated as SEM).

2.4.1 Loss Function

Parameters of each NN in an ensemble of size m are optimized with respect to the loss function $\mathcal{L}_{QD}$ (eq. 5). The loss function operates on mini-batches of size n.

$$\mathcal{L}_{QD} = \mathcal{L}_{MPIW} + \lambda \frac{n}{\alpha(1-\alpha)} \mathcal{L}_{PICP} \tag{5}$$

$\mathcal{L}_{MPIW}$ defined in eq. (6) is optimizing the width of the PIs capturing an observation. Due to variables c and k_i from eq. (2) and (3), $\mathcal{L}_{MPIW}$ only includes those samples for which the observation is inside the PI.

$$\mathcal{L}_{MPIW} = \frac{1}{c} \sum_{i=1}^{n} (\hat{y}_U^{(i)} - \hat{y}_L^{(i)}) \cdot k_i \tag{6}$$

$\mathcal{L}_{PICP}$ defined in eq. (7) is optimizing the coverage probability of the PIs penalizing only if the PICP is below the desired γ.

$$\mathcal{L}_{PICP} = \max(0, (1-\alpha) - PICP)^2 \tag{7}$$

Expanded, we get

$$\mathcal{L}_{QD} = \frac{1}{c}\sum_{i=1}^{n}(\hat{y}_U^{(i)} - \hat{y}_L^{(i)}) \cdot k_i + \lambda \frac{n}{\alpha(1-\alpha)} \max(0, (1-\alpha) - PICP)^2.$$

The hyper-parameter (Lagrangian) λ controls the importance of $\mathcal{L}_{PICP}$ with respect to $\mathcal{L}_{MPIW}$. The intuition behind the fraction $\frac{n}{\alpha(1-\alpha)}$ is that it should reflect the confidence of $\mathcal{L}_{PICP}$ with respect to n and α.

2.4.2 Aggregation Method

Given an ensemble of m NN models fitted with respect to $\mathcal{L}_{QD}$ (5), we acquire prediction intervals $[\hat{y}_L^{(ij)}, \hat{y}_U^{(ij)}]$ for sample $i \in \{1 \; .. \; n\}$ and model $j \in \{1 \; .. \; m\}$. The following calculations are taken to acquire the final predictions intervals $[\tilde{y}_L^{(i)}, \tilde{y}_U^{(i)}]$ that should account also for the epistemic uncertainty.

$$\mu_L^{(i)} = \frac{1}{m}\sum_{j=1}^{m}\hat{y}_L^{(ij)},$$

$$\mu_U^{(i)} = \frac{1}{m}\sum_{j=1}^{m}\hat{y}_U^{(ij)},$$

$$\sigma_L^{(i)^2} = \frac{1}{m-1}\sum_{j=1}^{m}\left(\hat{y}_L^{(ij)} - \mu_L^{(i)}\right)^2$$

$$\sigma_U^{(i)^2} = \frac{1}{m-1}\sum_{j=1}^{m}\left(\hat{y}_U^{(ij)} - \mu_U^{(i)}\right)^2$$

These quantities are then combined to generate the final PIs. In the paper (Pearce et al., 2018), the aggregation is done as follows:

$$\tilde{y}_L^{(i)} = \mu_L^{(i)} - 1.96 \cdot \sigma_L^{(i)}, \tag{8}$$

$$\tilde{y}_U^{(i)} = \mu_U^{(i)} + 1.96 \cdot \sigma_U^{(i)}. \tag{9}$$

The actual implementation differs in using a standard error of the mean (SEM) $\sigma_{\bar{y}_L}^{(i)}$ (10) and $\sigma_{\bar{y}_U}^{(i)}$ (11) instead of in the paper presented standard deviation $\sigma_L^{(i)}$ (8) and $\sigma_U^{(i)}$ (9) respectively; notice the introduction of the scalar $1/\sqrt{m}$ in both equations.

$$\tilde{y}_L^{(i)} = \mu_L^{(i)} - 1.96 \cdot \sigma_L^{(i)} \cdot \frac{\mathbf{1}}{\sqrt{\mathbf{m}}} \tag{10}$$

$$\tilde{y}_U^{(i)} = \mu_U^{(i)} + 1.96 \cdot \sigma_U^{(i)} \cdot \frac{\mathbf{1}}{\sqrt{\mathbf{m}}} \tag{11}$$

Aggregating PIs using the latter equations (10) and (11) results in narrower PIs. Both aggregation methods lack any theoretical justification: The lower and upper PI boundaries are aggregated independently, i.e. without mutual consideration, and we consider this as a flaw of the method. In fact, under a simple assumption of normal posterior distribution that is correctly captured by m NN models but with shifted PIs with correct γ coverage probability (PICP), it can be shown that both original aggregation methods in the above equations yield PIs resulting in PICP greater than γ, and thus greater MPIW. In the results, the latter aggregation function (eq. 10 and 11) was evaluated (annotated as SEM).

3 METHOD

In this section, we describe our method annotated as SNM-QD+ that provides a point prediction along with prediction interval as output. First, an ensemble of neural network models is trained with an extended quality-driven loss function annotated as QD+ (eq. 12). Second, the results from each model in the ensemble are aggregated to provide a final result (incorporating the epistemic uncertainty) by a fitting split normal density functions (Wallis, 2014) from each NN's point estimate and PI, and aggregating them into a split normal mixture (annotated as SNM) from which the final PI of coverage probability γ is calculated.

3.1 Quality-Driven Loss Function

By having a single model for prediction intervals and point estimates, we achieve a coherency of the results and so avoid the case of two disjoint models learning different function approximations for prediction intervals and point estimates. In other words, providing prediction intervals and point estimates as a result of two disjoint models may not capture the predictive uncertainty of the point estimate model.

We therefore propose a new loss function

$$\begin{aligned}\mathcal{L}_{QD+} = &(1-\lambda_1)(1-\lambda_2)\cdot\mathcal{L}_{MPIW} \\ &+ \lambda_1(1-\lambda_2)\cdot\mathcal{L}_{PICP} \\ &+ \lambda_2\cdot\mathcal{L}_{MSE} \\ &+ \xi\cdot\mathcal{L}_P\end{aligned} \tag{12}$$

where point estimates $\hat{y}^{(i)}$ are optimized by

$$\mathcal{L}_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}^{(i)} - y^{(i)}\right)^2 \tag{13}$$

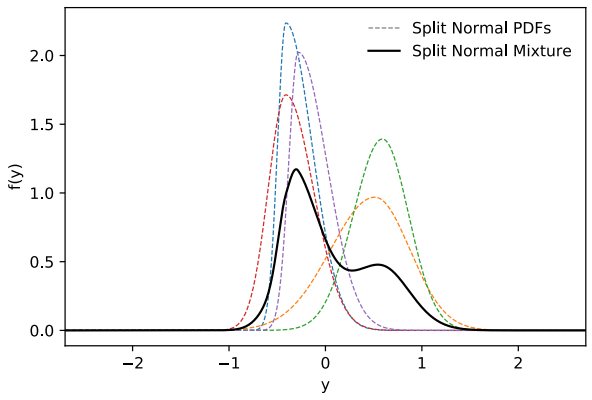


Figure 2: An example illustrating split normal PDFs (dashed) fitted from an ensemble ($m = 5$) of PIs and point estimates, and subsequently aggregated into a split normal mixture (black) from which the final PI is calculated.

and the penalty function

$$\mathcal{L}_P = \frac{1}{n}\sum_{i=1}^{n}\left[\max(0, \hat{y}_L^{(i)} - \hat{y}^{(i)}) + \max(0, \hat{y}^{(i)} - \hat{y}_U^{(i)})\right] \tag{14}$$

is adding a constraint to enforce their integrity.

We retrofit a slightly simplified version of the loss function by Pearce et al. (2018). We introduce an auxiliary loss $\mathcal{L}_{MSE}$ driving the point estimates, the mean squared error (MSE) in our particular use case. However, our empirical results showed rather difficult training process (exhibited already by the original loss function $\mathcal{L}_{QD}$) and issues with the integrity of the generated output, i.e. interval crossing and point estimates out of PI bounds. Therefore a constraint violation penalty function $\mathcal{L}_P$ was added which mitigated both issues significantly.

The hyper-parameter $\lambda_1 \in (0, 1)$ controls the mutual influence between losses $\mathcal{L}_{MPIW}$ (eq. 6) and $\mathcal{L}_{PICP}$ (eq. 7). The hyper-parameter $\lambda_2 \in (0, 1)$ controls the influence of the $\mathcal{L}_{MSE}$ (eq. 13) in relation to the aforementioned $\mathcal{L}_{MPIW}$ and $\mathcal{L}_{PICP}$. The hyper-parameter ξ of the penalty function $\mathcal{L}_P$ (eq. 14) controls the degree of penalization in case the constraint is violated.

3.2 Split Normal Aggregation Method

The prediction intervals and point estimates retrieved from the ensemble need to be aggregated into a final prediction interval, capturing both the aleatoric and epistemic uncertainty, and a final point estimate.

To this end, we assume that the posterior predictive distribution from a single model is a split normal distribution (Wallis, 2014). The split normal distribution or two-piece normal distribution is a result of joining halves of normal distributions with the same mode but different variances. The split normal probability density function (PDF) is defined as

$$f_{\mathcal{SN}}(x; \mu, \sigma_1, \sigma_2) = \begin{cases} A \exp\left(-\frac{(x-\mu)^2}{2\sigma_1^2}\right) & \text{if } x < \mu, \\ A \exp\left(-\frac{(x-\mu)^2}{2\sigma_2^2}\right) & \text{otherwise,} \end{cases}$$

where $A = \sqrt{2/\pi}\,(\sigma_1 + \sigma_2)^{-1}$.

Given the lower and upper PI bounds $[\hat{y}_L^{(ij)}, \hat{y}_U^{(ij)}]$ and the point estimates $\hat{y}^{(ij)}$ from an ensemble, we can fit parameters of a split normal distribution for each ensemble result by optimizing the loss

$$\begin{aligned}\mathcal{L}_{\mathcal{SN}} = &[F_{\mathcal{SN}}(\hat{y}_L^{(ij)}; \cdot) - \alpha/2]^2 \\ &+ [F_{\mathcal{SN}}(\hat{y}_U^{(ij)}; \cdot) - (1 - \alpha/2)]^2\end{aligned} \tag{15}$$

where the split normal cumulative density function (CDF) is defined as

$$F_{\mathcal{SN}}(x; \mu, \sigma_1, \sigma_2) = \begin{cases} \frac{\sigma_1 + \mathrm{erf}\left(\frac{x-\mu}{\sqrt{2}\sigma_1}\right)\sigma_1}{\sigma_1+\sigma_2} & \text{if } x < \mu, \\ \frac{\sigma_1 + \mathrm{erf}\left(\frac{x-\mu}{\sqrt{2}\sigma_2}\right)\sigma_2}{\sigma_1+\sigma_2} & \text{otherwise.} \end{cases}$$

Through a gradient descent (GD) optimization of eq. (15) we estimate the parameters $\sigma_1^{(ij)}$ and $\sigma_2^{(ij)}$ with respect to $[\hat{y}_L^{(ij)}, \hat{y}_U^{(ij)}]$ (variable x), $\hat{y}^{(ij)}$ (variable μ) and α.

When fitting the parameters $\sigma_1^{(ij)}$ and $\sigma_2^{(ij)}$ with eq. (15) as the objective, initialization of $(\sigma_1^{(ij)}, \sigma_2^{(ij)})$ plays an important role for finding the optimal solution. A random initialization does not always yield the optimal solution. Therefore, following the hypothesis that the parameters of a split normal distribution are close to those of normal distribution, we initialize the $\sigma_1^{(ij)}$ and $\sigma_2^{(ij)}$ as follows:

$$\sigma_1^{(ij)} = \frac{\hat{y}_L^{(ij)} - \hat{y}^{(ij)}}{\sqrt{2}\,\mathrm{erf}^{-1}(2p_L - 1)},$$

$$\sigma_2^{(ij)} = \frac{\hat{y}_U^{(ij)} - \hat{y}^{(ij)}}{\sqrt{2}\,\mathrm{erf}^{-1}(2p_U - 1)},$$

where $p_L = \alpha/2$ and $p_U = 1 - \alpha/2$. The above is derived from the inverse CDF (quantile function) of a normal distribution:

$$F_{\mathcal{N}}^{-1}(p) = \mu + \sigma\sqrt{2}\,\mathrm{erf}^{-1}(2p - 1)$$

with $p \in (0, 1)$.

In our experiments, this has been proven as superior compared to random initialization.

As a result of the above steps, we acquire an ensemble of m split normal PDFs $\{f_{\mathcal{SN}}(x;.)\}_{j=1}^{m}$ (for a single sample i) from which we create a mixture PDF

$$f^{(i)}(x) = \frac{1}{m}\sum_{j=1}^{m} f_{\mathcal{SN}}\left(x;\theta^{(ij)}\right).$$

See Figure 2 for illustration. The final PIs $[\tilde{y}_L^{(i)}, \tilde{y}_U^{(i)}]$ are calculated from the mixture distribution $f^{(i)}(x)$ by numerically solving

$$\tilde{y}_L^{(i)} = F^{-1^{(i)}}(\alpha/2) \quad \text{and} \quad \tilde{y}_U^{(i)} = F^{-1^{(i)}}(1-\alpha/2)$$

where $F^{-1^{(i)}}(p)$ is the inverse CDF of the mixture $f^{(i)}(x)$.

The final point estimates are calculated as an equally weighted combination (mean) of point estimates from an ensemble.

$$\tilde{y}^{(i)} = \frac{1}{m}\sum_{j=1}^{m} \hat{y}^{(ij)}$$

4 EXPERIMENTS

Our proposed loss function QD+ (Section 3.1) and aggregation method SNM (Section 3.2) are compared with the original QDE (Pearce et al., 2018), specifically with the loss function QD (Section 2.4.1) and aggregation method SEM (Section 2.4.2). We also compare against an ensemble of NNs optimizing the parametrization of a normal distribution (Lakshminarayanan et al., 2017), referred as a mean-variance estimator (MVE). Additionally, we compare point estimates generated by our method SNM-QD+ with an ensemble of NNs optimizing MSE (eq. 13) for point estimates solely.

We follow a similar experimental setup initially set by Hernandez-Lobato and Adams (2015) and also followed in related works (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Pearce et al., 2018; Tagasovska and Lopez-Paz, 2019).

4.1 Datasets

Ten open-access benchmark datasets from the UCI dataset repository are used (Dua and Graff, 2017). For each dataset we have created 20 shuffled versions across which we have evaluated the methods, i.e. 20 trials. Exceptions are the Year dataset with 1 trial (without shuffling) and the Protein dataset with 5 trials. The datasets are standardized (to zero mean and unit variance) based on the training set. The input of evaluation measures is standardized based on the full dataset for comparability across different trials.

4.2 Hyper-Parameters

We optimize for 95% prediction intervals, i.e. $\alpha = 0.05$.

To not add any competitive advantage to our method, we keep the NN sizes identical across different models. Note that this may be a disadvantage for our model, given the complexity of the loss function QD+. This potentially sub-optimal setting should be kept in mind as our comparison's objective is feasibility when compared to models specialized either to generate point estimates or PIs (and not both at the same time).

The ensemble size is $m = 5$. The size of mini-batches is $n = 100$ ($n = 1000$ for the Year dataset). All NNs have 2 hidden layers with 50 units (100 units for Protein and Year datasets) and ReLU activation functions.

For the hyper-parameter search (HPS), we do not follow the legacy of Hernandez-Lobato and Adams (2015). The HPS is performed only on a single concrete shuffled version of a dataset with excluded 10% test set, i.e. training set remains. Given the training set, the hyper-parameters (HPs) are validated on 5 shuffled 90% and 10% splits (i.e. 81% and 9% of the complete dataset) for training and validation, respectively. This deviates opposed to the originally suggested single 80% and 20% split (i.e. 72% and 18% of the complete dataset) without cross-validation. The change of the hyper-parameter search setup was motivated by the difficulty to find good HPs using the original setup, especially in smaller datasets (such as Boston, Concrete, Energy, Wine and Yacht).

A random HPS was performed on the following hyper-parameters (depending on the model): learning rate, decay rate, λ_1, λ_2, epochs. ξ is set to 10 for all experiments. The remaining settings are consistent with those of Pearce et al. (2018).

The applied heuristic for selecting the best hyper-parameters was as follows: If mean PICP is equal to $\gamma \pm 0.01$, then HPs with the lowest MPIW and MSE were selected. Alternatively, also heuristic considering mean and standard deviation of PICP proved good results. If neither of the criteria were fulfilled, the best achieved PICP was selected. Also, the training process on the validation set was analyzed for steady convergence and overfitting. Note that the loss cannot be used to select HPs because λ_1, λ_2 and ξ scale parts of the loss function; hence the loss is associated with particular values of λ_1, λ_2 and ξ.

Due to a smaller number of HPS trials (max. 300), we did not always find well-performing HPs. Therefore, some experiments (QD+ for the datasets Energy, Kin8nm, Naval, Protein and Year) were manually fine-tuned. Adjusting hyper-parameters (λ_1 and λ_2, learning rate, decay rate

Table 1: Ultimately, PIs are compared between SNM-QD+, SEM-QD and MVE, and point estimates are compared between SNM-QD+, MSE and MVE in Table 2. Additionally, aggregation methods are compared in SNM-QD+ vs. SEM-QD+. All methods are ensembles of 2-layer NNs. The values stand for mean ± standard error of the mean. The best results (with standard error of the mean taken in consideration) are shown in bold.

Dataset	PICP				MPIW			
	SNM-QD+	SEM-QD+	SEM-QD	MVE	SNM-QD+	SEM-QD+	SEM-QD	MVE
Boston	0.95 ± 0.01	0.97 ± 0.01	$\mathbf{0.95 \pm 0.01}$	0.88 ± 0.01	1.58 ± 0.06	1.82 ± 0.07	$\mathbf{1.52 \pm 0.06}$	0.89 ± 0.01
Concrete	$\mathbf{0.94 \pm 0.01}$	0.96 ± 0.01	0.97 ± 0.00	0.96 ± 0.00	$\mathbf{0.99 \pm 0.04}$	1.14 ± 0.05	1.36 ± 0.02	1.14 ± 0.02
Energy	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.01	$\mathbf{0.98 \pm 0.01}$	0.29 ± 0.01	0.33 ± 0.02	0.48 ± 0.03	$\mathbf{0.16 \pm 0.00}$
Kin8nm	0.97 ± 0.00	0.98 ± 0.00	0.99 ± 0.00	$\mathbf{0.97 \pm 0.00}$	1.07 ± 0.01	1.21 ± 0.01	1.29 ± 0.01	$\mathbf{1.04 \pm 0.01}$
Naval	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	$\mathbf{1.00 \pm 0.00}$	0.09 ± 0.00	0.10 ± 0.00	0.36 ± 0.02	$\mathbf{0.05 \pm 0.00}$
Power	$\mathbf{0.95 \pm 0.00}$	0.96 ± 0.00	0.96 ± 0.00	0.96 ± 0.00	$\mathbf{0.80 \pm 0.00}$	0.85 ± 0.00	0.87 ± 0.00	0.87 ± 0.01
Protein	$\mathbf{0.95 \pm 0.00}$	0.97 ± 0.00	0.95 ± 0.00	0.98 ± 0.00	$\mathbf{2.12 \pm 0.01}$	2.31 ± 0.01	2.24 ± 0.01	2.82 ± 0.10
Wine	$\mathbf{0.94 \pm 0.01}$	0.95 ± 0.01	0.92 ± 0.01	0.94 ± 0.00	$\mathbf{2.62 \pm 0.06}$	2.92 ± 0.07	2.06 ± 0.03	2.94 ± 0.02
Yacht	$\mathbf{0.94 \pm 0.01}$	0.96 ± 0.01	1.00 ± 0.00	0.99 ± 0.00	$\mathbf{0.12 \pm 0.00}$	0.13 ± 0.00	0.25 ± 0.02	0.40 ± 0.04
Year	$0.94 \pm$ NA	$0.96 \pm$ NA	$\mathbf{0.95 \pm}$ **NA**	$0.96 \pm$ NA	$2.34 \pm$ NA	$2.54 \pm$ NA	$\mathbf{2.29 \pm}$ **NA**	$2.75 \pm$ NA

Table 2: Extension of Table 1 evaluating point estimates of SNM-QD+, MSE and MVE.

Dataset	MSE		
	MSE	SNM-QD+	MVE
Boston	$\mathbf{0.112 \pm 0.013}$	0.115 ± 0.013	0.127 ± 0.019
Concrete	0.056 ± 0.003	$\mathbf{0.053 \pm 0.003}$	0.077 ± 0.004
Energy	0.001 ± 0.000	$\mathbf{0.001 \pm 0.000}$	0.001 ± 0.000
Kin8nm	0.060 ± 0.001	$\mathbf{0.059 \pm 0.001}$	0.062 ± 0.001
Naval	$\mathbf{4.8}$e-$\mathbf{5 \pm 0.000}$	1.3e-4 $\pm$ 0.000	1.2e-4 $\pm$ 0.000
Power	$\mathbf{0.042 \pm 0.001}$	0.050 ± 0.001	0.048 ± 0.001
Protein	$\mathbf{0.310 \pm 0.002}$	0.361 ± 0.004	0.445 ± 0.023
Wine	$\mathbf{0.597 \pm 0.015}$	0.616 ± 0.018	0.621 ± 0.015
Yacht	0.002 ± 0.000	$\mathbf{0.001 \pm 0.000}$	0.003 ± 0.000
Year	$0.637 \pm$ NA	$\mathbf{0.636 \pm}$ **NA**	$0.639 \pm$ NA

and number of epochs) has proven to be intuitive by analyzing PICP, MPIW and MSE on a validation set. We resorted to manual fine-tuning only if a model was considerably under-performing compared to the original QDE or compared to MVE. There was no attempt to optimize to the model's full potential.

4.3 Evaluation

The input of evaluation measures is standardized based on the full dataset for comparability across different trials. The following measures are used to assess and compare models (QD+, QD and MVE) and aggregation methods (SNM and SEM): PICP, MPIW and mean squared error (MSE).

4.4 Results

Tables 1 and 2 show the results of comparing the two models QD+ and QD, combined with the aggregation methods SNM and SEM. We have also included a model that generates only point estimates (denoted MSE), and MVE generating both PIs and point estimates. Consider SNM-QD+ vs. SEM-QD for a comparison between our method and the original QDE method (Pearce et al., 2018), respectively. SNM-QD+ vs. SEM-QD+ gives a comparison between our split normal aggregation method and the original aggregation method, respectively. SNM-QD+ vs. MVE provides a comparison of methods generating both PIs and point estimates. Finally, Table 2 shows the evaluation of the point estimates between SNM-QD+, MSE and MVE. The best results (with standard error of the mean taken in consideration) are presented using bold font. Supplement A provides the evaluation of non-aggregated PIs.

Overall, the results demonstrate that a complex multi-objective quality-driven loss function (QD+) can deliver well-calibrated PIs when the split normal mixture (SNM) is employed as the aggregation method. The proposed method SNM-QD+ performs best with respect to PIs and competitively with respect to point estimates. We argue that the quality of the point estimates can be improved by increasing the capacity of NNs or by hyper-parameter fine-tuning.

The results clearly show that the split normal aggregation method (SNM) yields well-calibrated PIs, while the original aggregation method (SEM) generally tends to generate wider PIs. Contrary to SEM, SNM aggregation method does not treat PI boundaries independently. It fits flexible enough (asymmetric) distributions within a versatile mixture distribution that is, as the results show, a sufficient approximation of the posterior predictive distribution. Note that the SNM aggregation method is not necessarily tied to the QD+ loss, but can be applied in other situations as well. A potential drawback of SNM is the apparent computational overhead of fitting the split normal mixture when the number of samples is large. However, this learning task is parallelizable and distributable; hence the wall-clock time spent on the task is negligible.

The training process of QD+ is, contrary to the original QD, quite robust. To give a particular example with the

Protein dataset, training 2-layer NNs with the original QDE required approximately one retry per run due to interval crossing or high loss value (stuck in local minimum). No retries were required when using QD+. Details are given in Supplement B. Furthermore, the original QD is sensitive to parameter initialization. Overall, the penalty function $\mathcal{L}_P$ (eq. 14) in QD+ results in significant improvements of the stability of the training process and strengthens the integrity of the output (mitigates the undesired interval crossing and point estimates out of the PI bounds). Supplement C provides a sensitivity analysis together with an ablation study clearly demonstrating the importance of the penalty function.

Since we compare our results with Pearce et al. (2018) and Lakshminarayanan et al. (2017), our work is also comparable with Bayesian approach by Hernandez-Lobato and Adams (2015), and partly also with Tagasovska and Lopez-Paz (2019).

4.5 Implementation

Our implementation is shared[1] and results fully reproducible. To embrace reproducibility, we use the Python package Sacred (Greff et al., 2017). PyTorch (Paszke et al., 2019) is used as the main NN framework and the fitting of a split normal mixture is implemented using JAX (Bradbury et al., 2018).

5 CONCLUSIONS

The main finding of this paper is that models delivering point estimates together with prediction intervals are competitive to models providing only one or the other. Motivated by the results of the quality-driven deep ensembles (Pearce et al., 2018) as an alternative to Bayesian methods, we endeavored to address the three key limitations: (1) The inability to generate point estimates; (2) the weakly justified method for aggregation of prediction interval ensembles; (3) the fragile training process.

We propose a new quality-driven loss function generating both prediction intervals and point estimates, and we dramatically increase the robustness of the training process by integrating a penalty function.

A unique and well-founded method fitting a split normal mixture as the aggregate of ensembled neural network output generates well-calibrated prediction intervals accounting for aleatoric and epistemic uncertainty. Moreover, an analytical approach for initializing parameters of a split normal probability density function is proposed that leads to acceleration and dramatically increased success of the fitting process.

[1] `https://github.com/tarik/pi-snm-qde`

With this work, we extend the practitioners toolset for quantifying predictive uncertainty in the regression task.

6 FUTURE WORK

We end the paper with the following suggestions for future research or possible improvements:

- We have shown that a single model architecture can be used to generate both point estimates and prediction intervals. We therefore envision a two-step learning process, where one first optimizes the model to generate point estimates and consequently learns to generate prediction intervals. To avoid catastrophic forgetting, it seems appropriate to treat this as a transfer learning problem. The key benefit of this operation is that it will allow us to reuse (pre-trained) NN models that currently only generate point estimates.

- It seems reasonable to use the NMPIW measure (eq. 4) in place of MPIW in the definition of the loss function $\mathcal{L}_{MPIW}$ (eq. 6). While it is already relatively intuitive to find HPs manually, and the HPS can be automated efficiently, we hypothesise that using NMPIW may lead to better-scaled hyper-parameters and possibly also to smaller variations across different datasets. The current work uses MPIW to ease the comparison between QD and QD+.

- The current approach learns to output a reasonable PI that is later used to fit a split normal PDF. This could alternatively be simplified so that the output of the neural network was the parameterization of the split normal PDF (or a mixture of split normal PDFs) directly. This approach is relevant for use-cases where a full predictive distribution is needed and would require a training objective defined for such situations instead of QD+. Again, the implemented approach was chosen to extend the non-Bayesian branch of research and ease comparison with the work of Pearce et al. (2018).

References

Alex Graves. Practical Variational Inference for Neural Networks. In *Advances in Neural Information Processing Systems 24*, pages 2348–2356. 2011.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Network. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *PMLR*, pages 1613–1622, 2015.

Jose Miguel Hernandez-Lobato and Ryan Adams. Probabilistic Backpropagation for Scalable Learning of

Bayesian Neural Networks. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *PMLR*, pages 1861–1869, 2015.

David Krueger, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron Courville. Bayesian Hypernetworks, 2017, arXiv:1710.04759.

Christos Louizos and Max Welling. Multiplicative Normalizing Flows for Variational Bayesian Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *PMLR*, pages 2218–2227, 2017.

Nick Pawlowski, Andrew Brock, Matthew C. H. Lee, Martin Rajchl, and Ben Glocker. Implicit Weight Uncertainty in Neural Networks, 2017, arXiv:1711.01297.

Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E. Turner, Jose Miguel Hernandez-Lobato, and Alexander L. Gaunt. Deterministic Variational Inference for Robust Bayesian Neural Networks. In *International Conference on Learning Representations*, 2019.

Pavel Izmailov, Wesley Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Subspace Inference for Bayesian Deep Learning. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019.

Jiayu Yao, Weiwei Pan, Soumya Ghosh, and Finale Doshi-Velez. Quality of Uncertainty Quantification for Bayesian Neural Network Inference, 2019, arXiv:1906.09686.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems 30*, pages 6402–6413. 2017.

Tim Pearce, Alexandra Brintrup, Mohamed Zaki, and Andy Neely. High-Quality Prediction Intervals for Deep Learning: A Distribution-Free, Ensembled Approach. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *PMLR*, pages 4075–4084, 2018.

Natasa Tagasovska and David Lopez-Paz. Single-Model Uncertainties for Deep Learning. In *Advances in Neural Information Processing Systems 32*, pages 6414–6425. 2019.

A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya. Lower Upper Bound Estimation Method for Construction of Neural Network-Based Prediction Intervals. *IEEE Transactions on Neural Networks*, 22(3):337–346, 2011.

Kenneth F. Wallis. The Two-Piece Normal, Binormal, or Double Gaussian Distribution: Its Origin and Rediscoveries. *Statistical Science*, 29(1):106–112, 2014.

Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? Does it matter? *Structural Safety*, 31(2): 105–112, 2009.

Andrey Malinin and Mark Gales. Predictive Uncertainty Estimation via Prior Networks. In *Advances in Neural Information Processing Systems 31*, pages 7047–7058. 2018.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *PMLR*, pages 1050–1059, 2016.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep Ensembles: A Loss Landscape Perspective, 2019, arXiv:1912.02757.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems 32*, pages 13991–14002. 2019.

Roger Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge University Press, 2005.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017.

Klaus Greff, Aaron Klein, Martin Chovanec, Frank Hutter, and Jürgen Schmidhuber. The Sacred Infrastructure for Computational Research. In *Proceedings of the 16th Python in Science Conference*, pages 49–56, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8026–8037. 2019.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. JAX: Composable Transformations of Python+NumPy Programs, 2018. URL `http://github.com/google/jax`.

SUPPLEMENTARY MATERIAL

A Additional Results

Related to Table 1, Table A shows the evaluation of non-aggregated PIs (not accounting for epistemic uncertainty) for the QD+ and QD models.

Table A: Evaluation of non-aggregated PIs (2-layer NNs).

Dataset	PICP		MPIW	
	QD+	QD	QD+	QD
Boston	0.85 ± 0.01	0.88 ± 0.01	1.22 ± 0.04	1.12 ± 0.02
Concrete	0.86 ± 0.01	0.89 ± 0.00	0.79 ± 0.03	1.03 ± 0.01
Energy	0.90 ± 0.01	0.93 ± 0.00	0.21 ± 0.01	0.27 ± 0.01
Kin8nm	0.91 ± 0.00	0.92 ± 0.00	0.95 ± 0.01	1.01 ± 0.01
Naval	0.98 ± 0.00	0.94 ± 0.00	0.08 ± 0.00	0.25 ± 0.00
Power	0.94 ± 0.00	0.93 ± 0.00	0.79 ± 0.00	0.75 ± 0.00
Protein	0.93 ± 0.00	0.93 ± 0.00	2.03 ± 0.01	2.11 ± 0.01
Wine	0.91 ± 0.00	0.91 ± 0.00	2.31 ± 0.03	1.81 ± 0.01
Yacht	0.76 ± 0.01	0.89 ± 0.01	0.07 ± 0.00	0.15 ± 0.01
Year	0.93 ± 0.00	0.93 ± 0.00	2.27 ± 0.00	2.07 ± 0.02

In Table B, we show the reproduced results of the original QDE method following the exact experimental setting as implemented in Pearce et al. (2018). These results are included to offer a comparison between the 1-layer and 2-layer models (Table B and Table 1, respectively). It also serves as a verification of our re-implementation of the method by Pearce et al. (2018).

Table B: Original QDE with 1-layer NNs as a reference.

Dataset	PICP		MPIW	
	SEM-QD	QD	SEM-QD	QD
Boston	0.90 ± 0.01	0.80 ± 0.01	1.01 ± 0.01	0.81 ± 0.01
Concrete	0.92 ± 0.01	0.84 ± 0.00	1.01 ± 0.01	0.83 ± 0.00
Energy	0.97 ± 0.00	0.91 ± 0.00	0.45 ± 0.01	0.38 ± 0.01
Kin8nm	0.96 ± 0.00	0.89 ± 0.00	1.24 ± 0.00	1.02 ± 0.00
Naval	0.99 ± 0.00	0.96 ± 0.00	0.25 ± 0.02	0.17 ± 0.00
Power	0.95 ± 0.00	0.94 ± 0.00	0.85 ± 0.00	0.81 ± 0.00
Protein	0.95 ± 0.00	0.94 ± 0.00	2.26 ± 0.00	2.17 ± 0.00
Wine	0.93 ± 0.01	0.91 ± 0.00	2.31 ± 0.02	2.04 ± 0.01
Yacht	0.95 ± 0.01	0.86 ± 0.01	0.15 ± 0.00	0.10 ± 0.00
Year	$0.95 \pm$ NA	0.93 ± 0.00	$2.41 \pm$ NA	2.22 ± 0.01

Table C considers a simple baseline with point estimates constructed as the mean of PIs from SEM-QD (annotated SEM-QD*). This simple baseline is underperforming compared to the results in Table 2.

B Robustness

The robustness of training process of QD+ vs. the fragility of training process of QD is shown in Table D. The contribution of the penalty function $\mathcal{L}_P$ to the robustness is shown in Figure A of the following section.

Table C: Extension of Table 2 provides a simple baseline with point estimates constructed as the mean of PIs from SEM-QD (annotated SEM-QD*). Also, the sizes $|\mathcal{D}|$ and input dimensions d of the datasets are presented.

Dataset			MSE
Name	$\lvert\mathcal{D}\rvert$	d	SEM-QD*
Boston	506	13	0.209 ± 0.030
Concrete	1030	8	0.102 ± 0.005
Energy	768	8	0.028 ± 0.008
Kin8nm	8192	8	0.067 ± 0.001
Naval	11934	16	0.012 ± 0.001
Power	9568	4	0.054 ± 0.001
Protein	45730	9	0.666 ± 0.004
Wine	1599	11	0.804 ± 0.021
Yacht	308	6	0.003 ± 0.001
Year	515345	90	$0.686 \pm$ NA

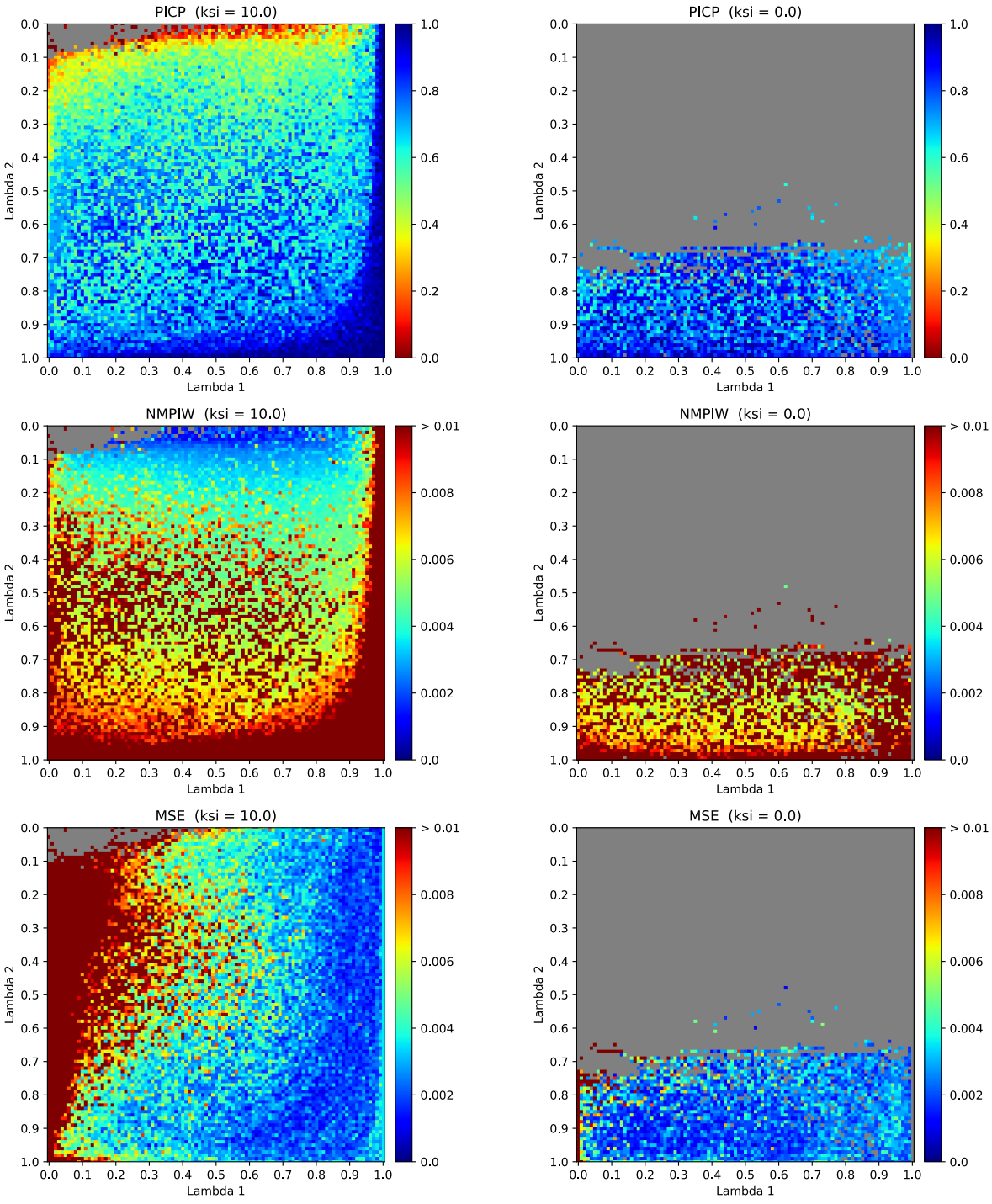
Table D: Extension of Tables 1 and 2 showing the number of failed/repeated training attempts of QD+ and QD. The targeted model count equals to an ensemble size m times number of trials t.

Dataset	Failures/Retries		Target Model Count
	QD+	QD	$m \cdot t$
Boston	0	3	$5 \cdot 20$
Concrete	0	2	$5 \cdot 20$
Energy	0	22	$5 \cdot 20$
Kin8nm	0	2	$5 \cdot 20$
Naval	0	10	$5 \cdot 20$
Power	0	6	$5 \cdot 20$
Protein	0	22	$5 \cdot 5$
Wine	0	29	$5 \cdot 20$
Yacht	0	33	$5 \cdot 20$
Year	0	18	$5 \cdot 1$

C Sensitivity Analysis

Figure A provides a sensitivity analysis with respect to hyper-parameters λ_1, λ_2 and ξ. It visualizes the measures PICP, NMPIW and MSE in relation to changing hyper-parameters of the QD+ model. The analysis was performed on the Yacht dataset. The sensitivity analysis is also an ablation study since the border values, i.e. $\lambda_1, \lambda_2 \in \{0, 1\}$ and $\xi = 0$, disable certain parts of the $\mathcal{L}_{QD+}$ loss function (eq. 12).

Figure Aa would guide us to choose λ_1 around 0.975 and λ_2 around 0.05. Leaving out the penalty function from the loss function (i.e. $\xi = 0$) leads to a non-converging training process for λ_2 below approx. 0.7 and more noisy measures in the remaining hyper-parameter space (Figure Ab). It demonstrates the critical importance of the penalty function $\mathcal{L}_P$ (eq. 14), thus, supporting the claims about the robustness together with the evidence in Table D.



(a) $\xi = 10$ (b) $\xi = 0$

Figure A: Sensitivity analysis of QD+ on Yacht dataset with respect to hyper-parameters λ_1 and λ_2 with (a) and without (b) an active penalty function $\mathcal{L}_P$. Gray color stands for a failed training process, i.e. high loss or outputs violating the semantic integrity (PI crossing or point estimates outside the PI bounds).