3rd Symposium on Advances in Approximate Bayesian Inference, 2020 1–27

A Novel Regression Loss for Non-Parametric Uncertainty Optimization

Anonymous Authors
Anonymous Institution

Abstract

Quantification of uncertainty is one of the most promising approaches to establish *safe* machine learning. Despite its importance, it is far from being generally solved, especially for neural networks. One of the most commonly used approaches so far is Monte Carlo dropout, which is computationally cheap and easy to apply in practice. However, it can underestimate the uncertainty. We propose a new objective, referred to as second-moment loss (SML), to address this issue. While the full network is encouraged to model the mean, the dropout networks are explicitly used to optimize the model variance. We intensively study the performance of the new objective on various UCI regression datasets. Comparing to the state-of-the-art of deep ensembles, SML leads to comparable prediction accuracies and uncertainty estimates while only requiring a single model. Under distribution shift, we observe moderate improvements. As a side result, we introduce an intuitive Wasserstein distance-based uncertainty measure that is non-saturating and thus allows to resolve quality differences between any two uncertainty estimates.

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 (Pomerleau, 1989; Bojarski et al., 2016) 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 (i.i.d. data sampling), but also under continuous domain shifts, think e.g. of adults on e-scooters or growing numbers of mobile health sensors. 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 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 uncertainty approaches have been established, stable quantification of uncertainty is still an open problem. Many recent machine learning applications are e.g. equipped with Monte Carlo (MC) dropout (Gal and Ghahramani, 2016) that offers conceptual simplicity and scalability. However, it tends to underestimate uncertainties thus bearing disadvantages compared to more recent approaches such as deep ensembles (Lakshminarayanan et al., 2017). We propose an alternative uncertainty mechanism. It builds on dropout sub-networks and explicitly optimizes variances (see Fig. 1 for an illustrative example). Technically, this is realized by a simple additive loss term, the *second-moment*

© A. Authors.

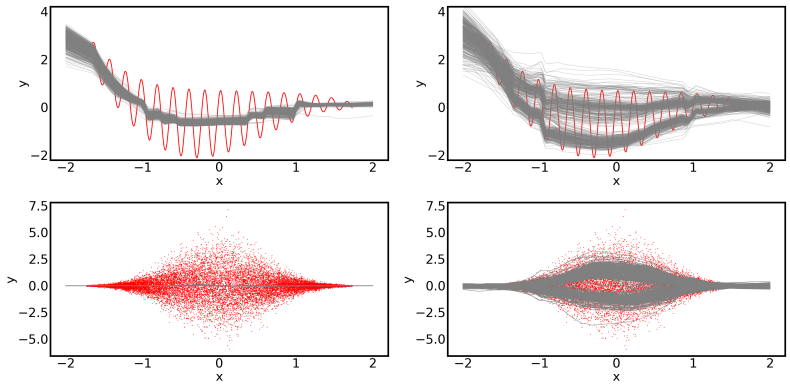

Figure 1: Sampling-based uncertainty mechanisms on toy datasets. The second-moment loss (right) induces uncertainties that capture aleatoric uncertainty. This is in contrast to MC dropout (left). Ground truth data is shown in red. Each grey line represents the outputs of one of 200 sub-networks that are obtained by applying dropout-based sampling to the trained full network.

loss. To address the above outlined requirements for safety-critical systems, we evaluate our approach systematically w.r.t. continuous data shifts.

In detail, our contribution is the introduction of a novel regression loss for better calibrated uncertainties applicable to dropout networks, reaching state-of-the-art performance in an empirical study and improving on it when considering data shifts.

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) and its variants (see e.g. Gal et al. (2017); Kendall and Gal (2017); Postels et al. (2019)) are prominent representatives of the first group. They offer 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 Blundell et al. (2015) and Ritter et al. (2018). Note that dropout training is also used—independent from an uncertainty context—for better model generalization (Srivastava et al., 2014).

Ensembles of neural networks, so-called deep ensembles (Lakshminarayanan et al., 2017), pose another popular approach to uncertainty modelling. Comparative studies of uncertainty mechanisms (Snoek et al., 2019; Gustafsson et al., 2020) highlight their advantageous uncertainty quality, making deep ensembles a state-of-the-art method. Fort et al. (2019) argue that deep ensembles capture multi-modality of loss landscapes and thus yield potentially more diverse sets of solutions.

The third group are parametric modelling approaches that extend point estimations by adding a model output that is interpreted as variance or covariance (Nix and Weigend, 1994; Heskes, 1997). Typically, these approaches optimize a (Gaussian) negative log-likelihood

(NLL, Nix and Weigend (1994)). A more recent representative of this group is, e.g., Kendall and Gal (2017), for a review see Khosravi et al. (2011). A closely related model class is deep kernel learning combining NNs and Gaussian processes (GPs) in various ways (see e.g. (Wilson et al., 2016; Iwata and Ghahramani, 2017; Garnelo et al., 2018; Qiu et al., 2019)).

The quality of uncertainties is typically evaluated using negative log-likelihood (Blei et al., 2006; Walker et al., 2016; Gal and Ghahramani, 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. Moreover, it is common to study how useful uncertainty estimates are for solving auxiliary tasks like out-of-distribution classification (Lakshminarayanan et al., 2017) or robustness w.r.t. adversarial attacks.

3. Second-moment loss

Monte Carlo (MC) dropout was proposed as a computationally cheap approximation of performing Bayesian inference in neural networks (Gal and Ghahramani, 2016). Given a neural network $f_\theta : \mathbb{R}^d \to \mathbb{R}^m$ with parameters θ, MC dropout samples sub-networks $f_{\tilde{\theta}}$ by randomly dropping nodes from the main model f_θ. During MC dropout inference the prediction is given by the mean estimate over the predictions of a given sample of sub-networks, while the uncertainty associated with this prediction can be estimated, e.g., in terms of the sample variance. During MC dropout training the objective function, in our case the mean squared error (MSE), is applied to the sub-networks separately. Due to this training procedure, all sub-network predictions are shifted towards the same training targets, which can result in overconfident predictions, i.e. in an underestimation of prediction uncertainty.[1]

Based on this observation, we propose to use the sub-networks $f_{\tilde{\theta}}$ in a different way: they are explicitly *not* encouraged to fit the data mean directly. This is the task of the full network f_θ. The sub-networks $f_{\tilde{\theta}}$ instead model aleatoric uncertainty and prediction residuals if the prediction of the full network f_θ is incorrect. Thus, we deliberately assign different 'jobs' to the main network f_θ on the one hand and its sub-networks on the other hand. Formalizing this idea into an optimization objective yields

$$L = L_{\text{regr}} + L_{\text{sml}} = \frac{1}{M} \sum_{i=1}^{M} \Big[\underbrace{(f_\theta(x_i) - y_i)^2}_{\text{regression loss}} + \beta \underbrace{\left(|f_{\tilde{\theta}}(x_i) - f_\theta(x_i)| - |f_\theta(x_i) - y_i| \right)^2}_{\text{second-moment loss}} \Big] \quad , \quad (1)$$

where the sum runs over a mini-batch of size $M < N$ taken from the set of observed samples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $x_i \in \mathbb{R}^d$ denotes the input, $y_i \in \mathbb{R}^m$ the ground-truth label, and $\beta > 0$ is a hyper-parameter that weights both terms. The first term, L_{regr}, is the MSE w.r.t. the full network f_θ. The second term, L_{sml}, seeks to optimize[2] the sub-networks $f_{\tilde{\theta}}$. It aims at finding sub-networks such that the distance $|f_{\tilde{\theta}} - f_\theta|$ matches the aleatoric uncertainty

1. An intuitive explanation is as follows: Let f_θ be a NN with one-dimensional output. For MC dropout with the MSE loss we get $\left\langle (f_{\tilde{\theta}}(x) - y)^2 \right\rangle = (\langle f_{\tilde{\theta}}(x) \rangle - y)^2 + \sigma^2(f_{\tilde{\theta}}(x))$. Therefore, it simultaneously minimizes the squared error between sub-network mean and target and the variance $\sigma^2(f_{\tilde{\theta}}(x)) = \langle f_{\tilde{\theta}}^2(x) \rangle - \langle f_{\tilde{\theta}}(x) \rangle^2$ over the sub-networks.
2. To avoid unintended optimization of full f_θ in direction of $f_{\tilde{\theta}}$, we only back-propagate through $f_{\tilde{\theta}}$ in L_{sml}.

or the prediction residual which is quantified by $|f_\theta(x_i) - y_i|$.[3] This leads to a significant increase in the variance of the sub-networks, i.e. the second moment of $f_{\tilde{\theta}}$, compared to standard MC dropout, which is why we name L_{sml} the *second-moment loss* (SML).[4] The standard deviations σ_{total} of the predictions of the sub-networks w.r.t. the prediction of the mean network induced by the SML have two components: the spread σ_{drop} of the sub-networks and an offset $\left|f_\theta - \langle f_{\tilde{\theta}} \rangle\right|$ between the full network and the sub-network mean that our loss might cause, concretely, $\sigma_{\text{total}} = \sigma_{\text{drop}} + |f_\theta - \langle f_{\tilde{\theta}} \rangle|$. While $|f_\theta - \langle f_{\tilde{\theta}} \rangle|$ is reminiscent of residual matching, σ_{drop} seems to be more closely related to modelling uncertainties. We show in appendix A.2 that σ_{drop} accounts on average for more than 80% of σ_{total} in our experiments.

Note that while we investigate the proposed objective in terms of dropout sub-networks in this paper, our arguments as well as the actual approach are generally applicable to other models that allow to formulate sub-networks given some kind of mean model. Besides the regression tasks considered here our approach could be useful for other objectives which use or benefit from an underlying distribution, e.g. uncertainty quantification in classification.

4. Experiments

We study uncertainty quality on UCI regression datasets, where we extend the dataset selection in Gal and Ghahramani (2016) by adding three further datasets: 'diabetes', 'california', and 'superconduct'. Apart from i.i.d. 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 train data. Appendix B.2 elaborates on our ways of splitting the data, namely *pca-based* splits in input space (using the first principal component) and *label-based* splits. We assess uncertainty performance in terms of the expected calibration error (ECE) and Wasserstein distance (WS) and regression performance using root-mean-squared error (RMSE) and negative log-likelihood (NLL). All measures are described in detail in appendix B.1, where you can also find more details on the network, the implementation of the methods and the training procedure. For brevity of exposition, we limit our discussion here largely to the ECE. An evaluation of the other measures can be found in appendix B.2. All presented results are 5- or 10-fold cross validated.

Fig. 2 provides ECEs for 13 UCI datasets that are sorted by dataset size on the x-axis. The top panel shows train- (green) and test-set (blue) ECEs, the bottom panel test-set ECEs under two pca-based data shifts (yellow-green, orange) and two label-based data splits (red, light red), for inter- and extrapolation respectively. Uncertainty methods are encoded via plot markers, e.g. PU-DE as 'star' and SML-trained networks ('ours') as 'square'. We summarize these dataset-specific results on the right hand side of the figure (light grey background). The columns 'mean' and 'median' of this summary show that on

3. As our choice of L_{sml} removes all directional information of the residual, possible (optimal) solutions for the $f_{\tilde{\theta}}$ are not uniquely determined. For a one-dimensional example based on aleatoric uncertainty see appendix A.1.
4. For brevity, we also refer to the entire loss objective L as second-moment loss during evaluation.

training sets, ECEs are smallest for PU, followed by PU-DE and the SML network. On test data, however, PU, PU-DE and the SML network share the first place. Looking at the stability w.r.t. data shift, i.e. the ability to extra- or interpolate to "unseen" data, PU loses in performance while PU-DE and SML reach the smallest calibration errors in three out of four cases, compare the lower panel in Fig. 2.

Summarizing these evaluations, we find SML to be as strong as the state-of-the-art method of PU-DEs while using only a single network compared to an ensemble of 5 networks. We moreover observe advantages for SML under PCA- and label-based data shifts. Three datasets lead to overestimated uncertainties for the SML, see discussion in appendix B.2. A visual tool to further inspect uncertainty quality are residual-uncertainty scatter plots as shown in appendix B.3. For a reflection on NLL and comparisons of the different uncertainty measures see appendix B.2.

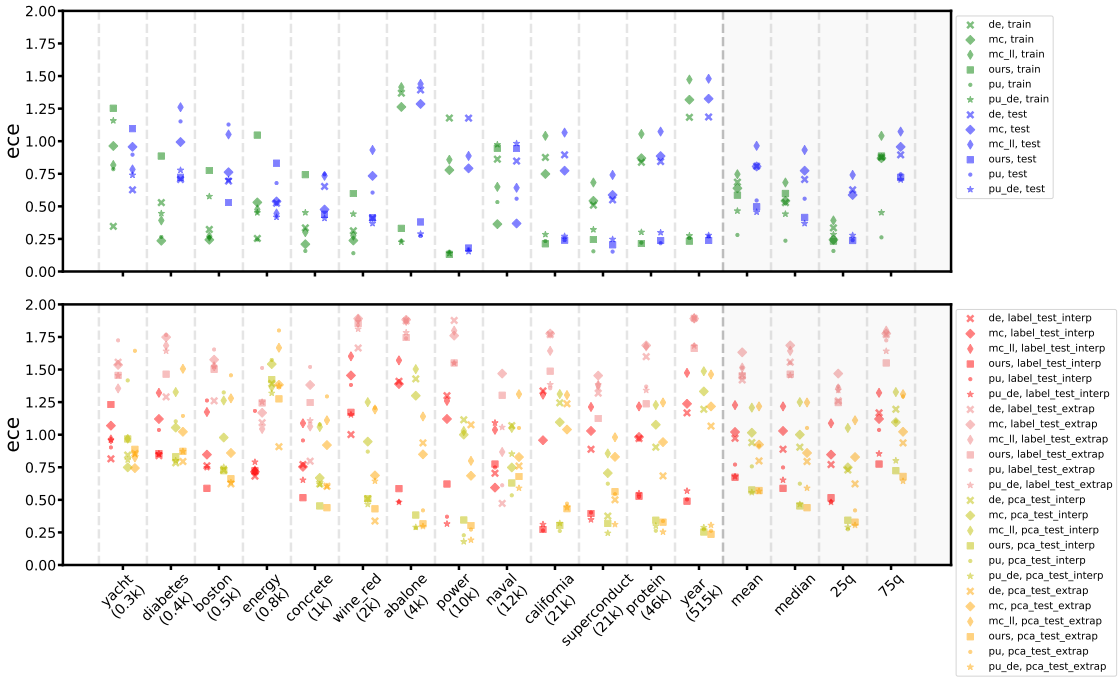

Figure 2: Expected calibration errors (ECEs) for 13 UCI regression datasets under i.i.d. conditions (top) and under data shift (bottom). Uncertainty methods are encoded via plot marker, data splits via color. Each plot point corresponds to a cross-validated trained network. Summarizing statistics (rhs) are indicated by a light grey background.

5. Conclusion

We approach dropout-based uncertainty quantification from a new direction: sub-networks are explicitly not encouraged to model the data mean, they capture aleatoric uncertainties and potential fitting residuals of the full network instead. Technically, this is realized by

an additional loss term that accompanies the standard regression objective: the *second-moment loss*. Our loss enables stable training. Training complexity and runtime behavior at inference are comparable to MC dropout. Task performances and uncertainty qualities of these models are on par with (parametric) deep ensembles, the widely used state-of-the-art for uncertainty quantification. However, unlike deep ensembles, we use single networks. In practice, this might allow to reduce training effort significantly compared to deep ensembles, especially for application-scale networks. Moreover, a single network requires only a fraction of the storage of a deep ensemble, making models with competitive uncertainties more accessible for mobile or embedded applications.

An extensive study of uncertainties under data shift revealed advantages of SML-trained models compared to deep ensembles: while both methods *on average* provide comparable results, we find a higher stability across a variety of datasets and data shifts for the SML. Technically, we attribute this gain in stability to our sub-network-based approach: like MC dropout, we integrate uncertainty estimates into the very structure of the network, rendering it more robust towards unseen inputs than a parameter estimate.

Moreover, the second-moment loss can serve as a drop-in replacement for MC dropout on regression tasks. For already trained MC dropout models, post-training with the second-moment loss might suffice to improve uncertainty quality. As an outlook, our first such post-training experiments are encouraging. Another interesting variant is the combination of SML with last-layer dropout (MC-LL) as it enables sampling-free inference (Postels et al., 2019). Preliminary experiments show clearly improved uncertainty qualities compared to standard MC-LL. A potentially interesting avenue for near real-time applications.

The simple additive structure of the second-moment loss makes it applicable to a variety of optimization objectives. For classification, we might be able to construct a non-parametric counterpart to prior networks (Malinin and Gales, 2018). Taking a step back, we demonstrated an easily feasible approach to influence and train sub-network distributions. This could be a promising avenue, for distribution matching but also for theoretical investigations.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

David M Blei, Michael I Jordan, et al. Variational inference for Dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Andrew YK Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. 'in-between'uncertainty in Bayesian neural networks. *arXiv preprint arXiv:1906.11537*, 2019.

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

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

Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Advances in Neural Information Processing Systems*, pages 3581–3590, 2017.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. JMLR Workshop and Conference Proceedings.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable Bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 318–319, 2020.

Tom Heskes. Practical confidence and prediction intervals. In *Advances in Neural Information Processing Systems*, pages 176–182, 1997.

Tomoharu Iwata and Zoubin Ghahramani. Improving output uncertainty estimation and generalization in deep learning via neural network Gaussian processes. *arXiv preprint arXiv:1707.05922*, 2017.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5574–5584, 2017.

Abbas Khosravi, Saeid Nahavandi, Doug Creighton, and Amir F Atiya. Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Transactions on Neural Networks*, 22(9):1341–1356, 2011.

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

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *35th International Conference on Machine Learning, ICML 2018*, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

Siqi Liu, Sidong Liu, Weidong Cai, Sonia Pujol, Ron Kikinis, and Dagan Feng. Early diagnosis of alzheimer's disease with deep learning. In *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pages 1015–1018. IEEE, 2014.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, pages 7047–7058, 2018.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 2015, page 2901, 2015.

David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of IEEE International Conference on Neural Networks 1994*, volume 1, pages 55–60. IEEE, 1994.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.

Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari. Sampling-free epistemic uncertainty estimation using approximated variance propagation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2931–2940, 2019.

Xin Qiu, Elliot Meyerson, and Risto Miikkulainen. Quantifying point-prediction uncertainty in neural networks via residual estimation with an i/o kernel. *arXiv preprint arXiv:1906.00588*, 2019.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable Laplace approximation for neural networks. *ICLR*, 2018.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, page 59, USA, 1998.

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

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

Michael A Stephens. EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730–737, 1974.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016.

Supplementary Material

This part accompanies our paper "*A Novel Regression Loss for Non-Parametric Uncertainty Optimization*" and provides further in-depth information. In Section A we provide both theoretical and numerical insight into the resulting uncertainties of our loss modification. Large parts of the empirical evaluation can be found in section B, including details on the setup, data splits as well as further uncertainty measures. As the second-moment loss couples to the usual MSE regression loss via a hyper-parameter β we test various values in section C, finding no strong correlation between result and parameter. We close with a discussion on the relations between uncertainty measures and their respective sensitivity in section D.

Appendix A. Mechanics of the second-moment loss

We analytically study the optimization landscape evoked by the second-moment loss in A.1. This analysis provides building blocks to better understand the composition of the SML-uncertainties as detailed in the remainder of this section.

A.1. Analytical properties of the second-moment loss

In the following, we look closer at the behaviour of the second-moment loss with respect to aleatoric uncertainty. For this, we assume that the residuals, compare eq. (1), are given by a Gaussian distribution with, for simplicity, $\mu_{\text{Res.}} = 0$ and $\sigma_{\text{Res.}} = 1$. We want to determine the resulting loss for the L_{sml} term in eq. (1) governing the uncertainty estimation of the model. It depends on the underling distribution of the effective MC dropout distribution, which me model as $\mathcal{N}(\mu_{\text{drop}}, \sigma_{\text{drop}})$ such that:

$$L_{\text{sml}} = \int_{-\infty}^{\infty} \mathrm{d}y_1 \mathrm{d}y_2 \left(|y_1| - |y_2|\right)^2 p_1(y_1)\, p_2(y_2)\,, \tag{2}$$

where p_1 and p_2 are the Gaussian distributions discussed above. After some calculation this yields:

$$L_{\text{sml}} = -\frac{4}{\pi}\sigma_{\text{drop}} \exp\left(-\frac{1}{2}\frac{\mu_{\text{drop}}^2}{\sigma_{\text{drop}}^2}\right) - \sqrt{\frac{8}{\pi}}\,\mu_{\text{drop}}\ \text{Erf}\left(\frac{\mu_{\text{drop}}}{\sqrt{2}\,\sigma_{\text{drop}}}\right) + \sigma_{\text{drop}}^2 + \mu_{\text{drop}}^2 + 1\,, \tag{3}$$

which is visualized in Fig. 3. The two global minima can be found for $\sigma_{\text{drop}} = 0$ and $\mu_{\text{drop}} = \pm\sqrt{2/\pi}$. However, as we model a randomized residual y_1 these minima do not reach zero. We find that it is favourable to move μ_{drop} away from the network prediction of $\mu_{\text{Res.}} = 0$, the mean of the underlying data distribution. But, this is only the case as long as the inherent uncertainty in the dropout distribution can be brought below $\sigma_{\text{drop}} < 2/\pi$, which is still smaller than the uncertainty of $\sigma_{\text{Res.}} = 1$ assumed within the training data distribution. Otherwise, it is more favourable to have $\mu_{\text{drop}} = \mu_{\text{Res.}} = 0$. Decomposing the uncertainty for the UCI datasets in section A.2 showed mixed behaviour with indications for bi-modal shifts in μ_{drop} as well as improved values of σ_{drop}.

We already showed the effect of this bi-modality in Fig. 1 at the beginning of the paper, where various sub-networks where sampled. Clearly visible is a stronger variation

between the networks compared to MC, but also a concentration around the two possible minima. While this Fig. provides a good visual estimate of σ_{drop} the total uncertainty σ_{total} would additionally contain the systematic shift $|f_\theta - \langle f_{\tilde{\theta}} \rangle|$. Given the roughly symmetric distribution of the sub-networks we can expect it to be comparatively small.

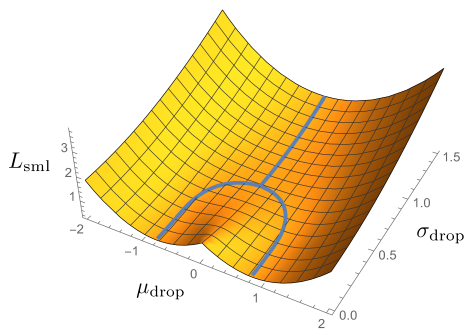

Figure 3: Shown is the value of the loss component L_2 as given by eq. (3) over μ_{drop} and σ_{drop} describing the implicit dropout ensemble. The blue line shows the position of the minima of L_2 for fixed values of σ_{drop}. Clearly visible are the global minima at $\sigma_{\text{drop}} = 0$ and the bifurcation at $\sigma_{\text{drop}} = 2/\pi$.

A.2. Composition of the uncertainty estimate

The uncertainty estimate of the second-moment loss is comprised of two parts: $\sigma_{\text{total}} = \sigma_{\text{drop}} + |f_\theta - \langle f_{\tilde{\theta}} \rangle|$. Fig. 4 reveals that σ_{drop} contributes to more than 80% of σ_{total} for the three presented datasets and for all applied data splits. A highly similar behavior can be observed for all other datasets. The analytical consideration in appendix A.1 suggests that for cases without aleatoric uncertainty the SML provides no incentive for $|f_\theta - \langle f_{\tilde{\theta}} \rangle| > 0$. The same holds true in the presence of aleatoric uncertainty as long as σ_{drop} is comparably large. For aleatoric uncertainty and small σ_{drop} larger $|f_\theta - \langle f_{\tilde{\theta}} \rangle|$ are favorable. However, as our loss is radial symmetric, all directions are equivalent and initialization and randomness determine the direction of the spread $|f_\theta - \langle f_{\tilde{\theta}} \rangle|$ for each individual sub-network. This symmetry leads again to a small averaged $|f_\theta - \langle f_{\tilde{\theta}} \rangle|$. σ_{drop} on the contrary describes the width of a bi-modal set of sub-networks in these cases.

A.3. Detailed analysis of the two loss components

A deeper look into the structure of the second-moment loss is possible if we investigate its behaviour component-wise. To clarify the results presented in Fig. 5, we recall the loss structure as

$$L = L_1 + L_2 = \sum_{i=1}^{M} \left[a_i^2 + \beta \left(|b_i| - |a_i| \right)^2 \right] \tag{4}$$

with $a_i = f_{\boldsymbol{\theta}}(x_i) - y_i$ and $b_i = f_{\tilde{\theta}}(x_i) - f_\theta(x_i)$. Histograms of the a_i (Fig. 5, first column) enable a detailed view on network performance. The uncertainty quality of the networks can be judged by studying the L_2 loss term more closely, namely by visualizing histograms of

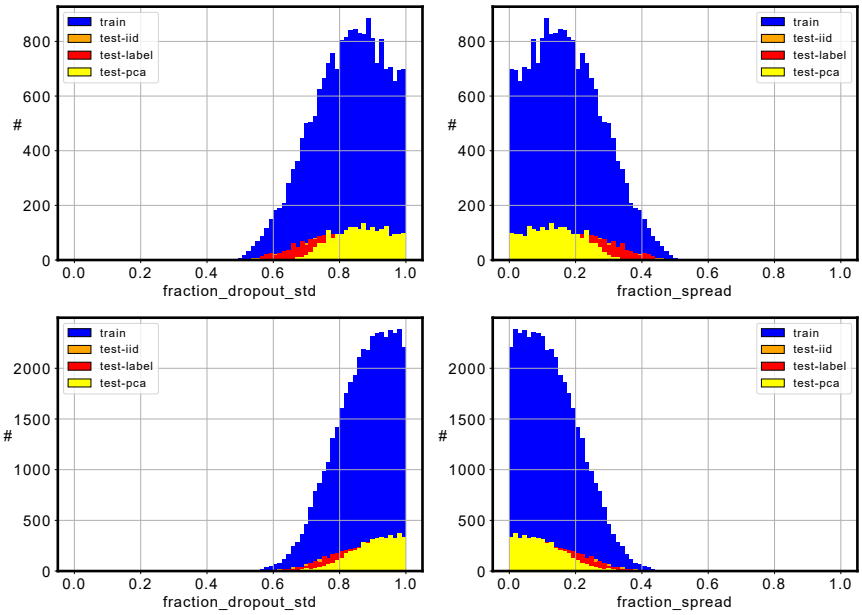

Figure 4: The second-moment loss induces uncertainties $\sigma_{\text{total}} = \sigma_{\text{drop}} + |f_\theta - \langle f_{\tilde{\theta}} \rangle|$. The relative contribution of both components ("fraction_dropout_std", "fraction_spread") is shown for two exemplary datasets (top: superconduct, bottom: protein) and i.i.d. (train: blue, test: orange) as well as non-i.i.d. data splits (test-label: red, test-pca: yellow).

$|b_i| - |a_i|$ (fourth column). The second and third column zoom into L_2 and show histograms of the b_i and scatter plots of (b_i, a_i), respectively. Only test datasets are visualized and as we applied $90:10$ train-test splits, this explains the low resolution of some histograms in the first column. All quantities involving b_i require the sampling of sub-networks. We draw 200 sub-networks. This sampling procedure explains the higher plot resolutions in columns two to four.

Qualitatively, we observe that both the a_i's and b_i's are centered around zero which hints at successful optimization of regression performance and of uncertainty quality. Details on how the optimization is technically realistic, can be gained from the scatter plots. They show two qualitative shapes: a 'line' (first row) and a 'blob' (second and third row). For an in-detail discussion of the uni- and bi-modality of the second-moment loss landscape see A.1. A 'line' shape reflects that all sub-networks occupy the same minimum given a bi-modal case. Following appendix A.1, a 'blob' indicates a uni-modal case that might be evoked by large standard deviations σ_{drop}.

Appendix B. Extension to the empirical study

Accompanying to the evaluation sketched in the body of the paper, section 4, we provide more details on the setup, used benchmarks and measures in the following sub-section. Further information on the experiments are given in section B.2, which we extend by the

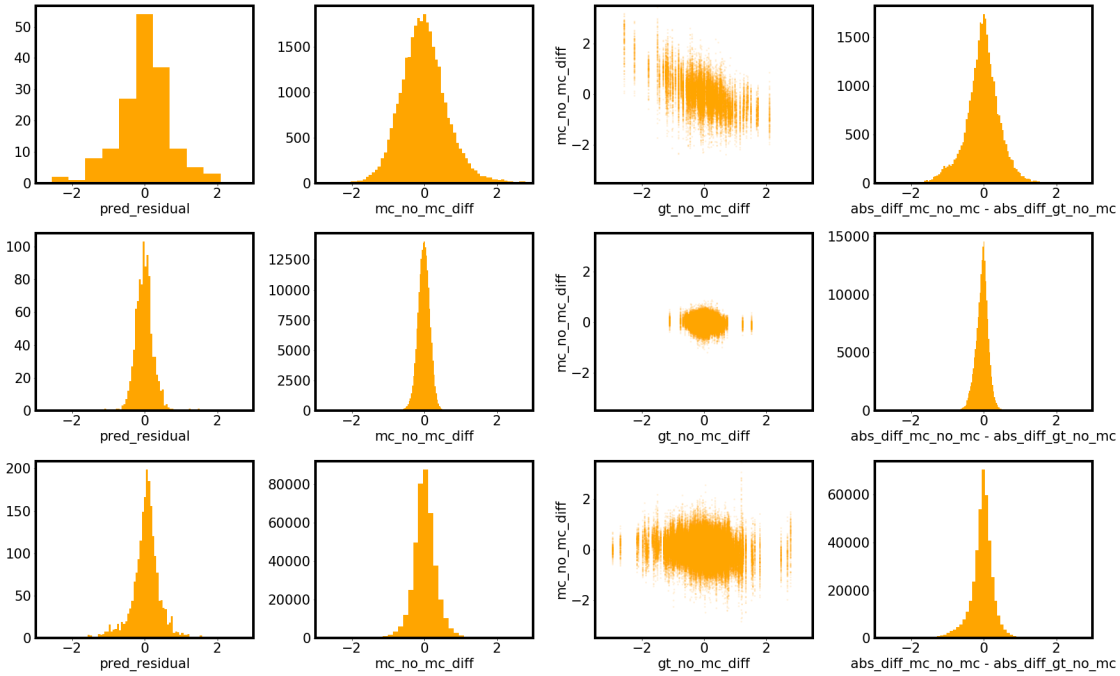

Figure 5: Visualisation of the components (columns) of the second-moment loss for selected test datasets (rows). The prediction residual $f_{\boldsymbol{\theta}}(x_i) - y_i$ (first column), model spread $f_{\tilde{\theta}}(x_i) - f_{\theta}(x_i)$ (second column), a scatter plot of both quantities (third column) and $|f_{\tilde{\theta}}(x_i) - f_{\theta}(x_i)| - |f_{\boldsymbol{\theta}}(x_i) - y_i|$ (fourth column) are shown. The chosen datasets from top to bottom are: wine-red, power and california.

measures skipped in the main text, and include a description on the used label splits. We close with a look at the predicted uncertainties (per method) via scatter plots in section B.3.

B.1. Experimental setup

The experimental setup used for the experiments is presented in three parts: the benchmark approaches we compare with, the evaluation measures we apply to quantify uncertainty, and a description of the neural networks and training procedures we employ.

Benchmark approaches We compare dropout networks trained with the SML to archetypes of uncertainty modelling, namely approximate Bayesian techniques, parametric uncertainty, and ensembling approaches. From the first group, we pick MC dropout (abbreviated as **MC**) and its variant last-layer MC dropout (**MC-LL**). While these dropout approaches integrate uncertainty estimation into the very structure of the 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) and are trained by likelihood maximization. This approach is denoted as **PU**

Non-parametric Regression Uncertainty

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). Moreover, we consider ensembles of non-parametric standard networks. We refer to the latter ones as **DEs** while we call those using PU **PU-DEs**. All considered types of networks provide estimates (μ_i, σ_i) where σ_i is obtained either analytically (PU), by sampling (MC, MC-LL, SML) or as an ensemble aggregate (DE, PU-DE).

Evaluation measures In all experiments we evaluate both regression performance and uncertainty quality. Regression performance is quantified by the root-mean-square error (**RMSE**), $\sqrt{(1/N \sum_i (\mu_i - y_i)^2}$ (Bishop, 2006). Another established metric in the uncertainty community is the (Gaussian) negative log-likelihood (**NLL**), $1/N \sum_i \big(\log \sigma_i + (\mu_i - y_i)^2/(2\sigma_i^2) + c\big)$, a hybrid between performance and uncertainty measure (Gneiting and Raftery, 2007), see appendix D.2 for a discussion.[5] The expected calibration error (**ECE**, Kuleshov et al. (2018)) in contrast is not biased towards well-performing models and in that sense is a pure uncertainty measure. It reads $\text{ECE} = \sum_{j=1}^{B} |\tilde{p}_j - 1/B|$ 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. The normalized prediction residuals r_i are defined as $r_i = (\mu_i - y_i)/\sigma_i$. Additionally, we propose to consider the *Wasserstein distance of normalized prediction residuals* (**WS**). The 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 ML. For ideally calibrated uncertainties, we expect $y_i \sim \mathcal{N}(\mu_i, \sigma_i)$ and therefore $r_i \sim \mathcal{N}(0, 1)$. Thus we use $d_{\text{WS}}(\{r_i\}_i, \mathcal{N}(0, 1))$ to measure deviations from this ideal behavior. As ECE, this is a pure uncertainty measure. However, it does not use binning and can therefore resolves deviations on all scales. For example, two strongly ill-calibrated uncertainties ($r_1, r_2 \gg 1$, $r_1 < r_2$) would result in (almost) identical ECE values while WS would resolve this difference in magnitude.

Technical details All investigated neural networks have the same architecture, 2 hidden layers of width 50, and ReLu activations (Glorot et al., 2011). For all dropout-based methods (MC, MC-LL, SML) we set the drop rate to $p = 0.1$. Like MC, SML-trained networks apply Bernoulli dropout to all hidden activations. In the case of MC-LL the dropout is only applied to the last hidden layer. For ensemble methods (DE, DE-PU) we employ 5 networks. For PE networks, we normalize the σ value using softplus (Glorot et al., 2011) and optimzie the NLL instead of the MSE. For the optimization of all NNs we use the ADAM-optimizer (Kingma and Ba, 2014) with a learning rate of 0.001. For 'california', the learning rate is reduced to 0.0001 as training of PU and PU-DE is unstable using the standard setup. Additionally, we apply standard normalization to the input and output features of all datasets to enable better comparability.

Number of epochs trained and amount of cross validation differs by the training-set size. We categorize the datasets as follows: small datasets {yacht, diabetes, boston, energy, concrete, wine-red}, large datasets {abalone, power, naval, california, superconduct, protein} and very large datasets {year}. For small datasets, NNs are trained for $1,000$ epochs using mini-batches of size 100. All results are 10-fold cross validated. For large datasets, we train

5. Throughout the paper, we ignore the constant $c = \log \sqrt{2\pi}$ of the NLL.

14

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. Conducting the described experiments with cross validation on one CPU takes $80\,h$.

For SML it turns out that as long as $0<\beta<1$, the actual value of β has only a limited influence on the optimization result, see appendix C for details. Larger β-values can however favour uncertainty optimization at an expanse of task performance. Throughout the body of the paper we use a conservative value of $\beta = 0.5$.

B.2. RMSEs, NLLs and systematic evaluation

This sub-section provides further details on our experiments covering: an overview on the datasets and splits used for the data-shift studies, further uncertainty measure evaluations (RMSE, NLL, WS), and close with a discussion of the weaker SML results.

Datasets and data splits For the regression data, Table 1 provides details on dataset references, preprocessing and basic statistics. Extrapolation and interpolation data-shifts are, technically, introduced by applying non-i.i.d. (independent and identically distributed) 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. 6, 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. 6, left). The actual *PCA-split* is then based on projections of the data points onto this first PCA-component.[6] Splitting data along such an 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*. We introduce this distinction as extrapolation is expected to be considerably more difficult than 'bridging' between feature combinations that were seen during training.

Regression quality First, we consider regression performance (see top panel of Fig. 7). Averaging the RMSE values over the considered 13 datasets ('mean' column) yields almost identical results for all uncertainty methods. The only exceptions pose PU and PU-DE with larger train data 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. Next, we investigate model performance under data shift, visualized in the bottom panel of Fig. 7. Again, regression quality is comparable between all methods. As expected, performances under data shift are worse compared to those on i.i.d. test sets.

Negative log-likelihoods For NLL, results are less balanced compared to RMSE (see Fig. 8). PU-DE and the SML-trained network reach comparably small average values, followed by MC and DE. The average NLL values of MC-LL and PU are above the upper

6. Note that these projections are only considered for data splitting, they are not used for model training.

Table 1: Details on UCI regression datasets. Ground truth (gt) is partially pre-processed to match the 1D regression setup.

dataset	# features	# datapoints	reference	remarks
yacht	6	308	UCI	
diabetes	7	442	sklearn	
boston	13	506	sklearn	
energy	8	768	UCI	only "cooling load" gt used
concrete	8	1030	UCI	
wine-red	11	1599	UCI	
abalone	7	4176	UCI	1st feature is ignored
power	4	9568	UCI	
naval	16	11934	UCI	using only "turbine" gt
california	8	20640	sklearn	
superconduct	81	21263	UCI	
protein	9	45730	UCI	
year	90	515345	UCI	

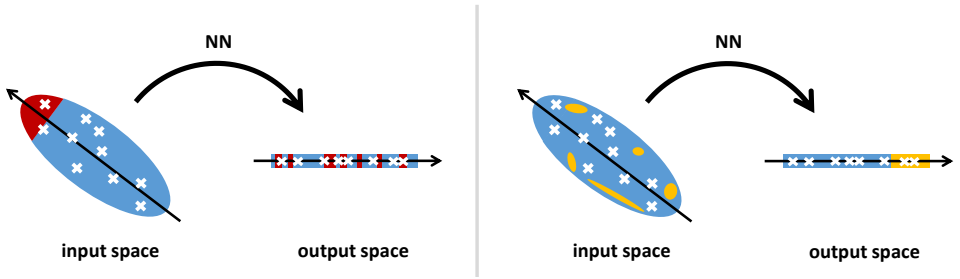

Figure 6: Scheme of two non-i.i.d. splits: a PCA-based split in input space (left) and label-based split in output space (right). While datasets appear to be convex here, they are (most likely) not in reality.

plot limit indicating a rather weak stability of these methods. On PCA-interpolate and PCA-extrapolate test sets, again PU-DE and SML-trained networks perform best. On label-interpolate and label-extrapolate test sets, SML-trained networks take the first place with a large margin. The mean NLL values of most other approaches are above the upper plot limit. Note that median results (the column next to 'mean') are not as widely spread and PU-DE and SML 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

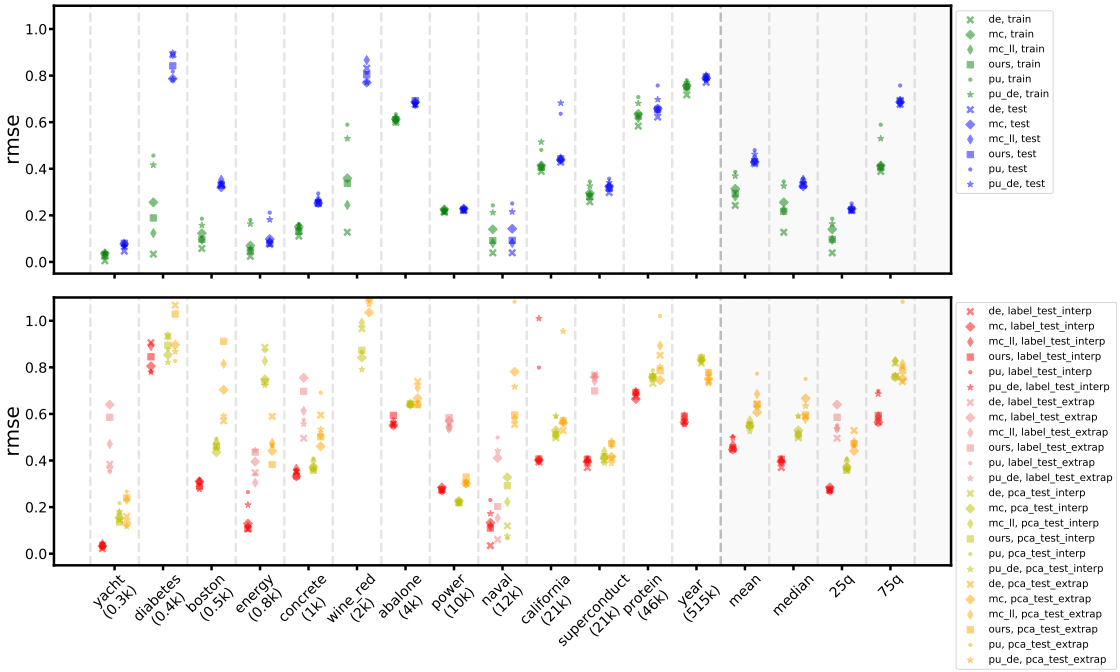

Figure 7: Root-mean-square errors (RMSEs) for 13 UCI regression datasets under i.i.d. conditions (top) and under data shift (bottom). Uncertainty methods are encoded via plot marker, data splits via color. Each plot point corresponds to a cross-validated trained network. Summarizing statistics (rhs) are indicated by a light grey background.

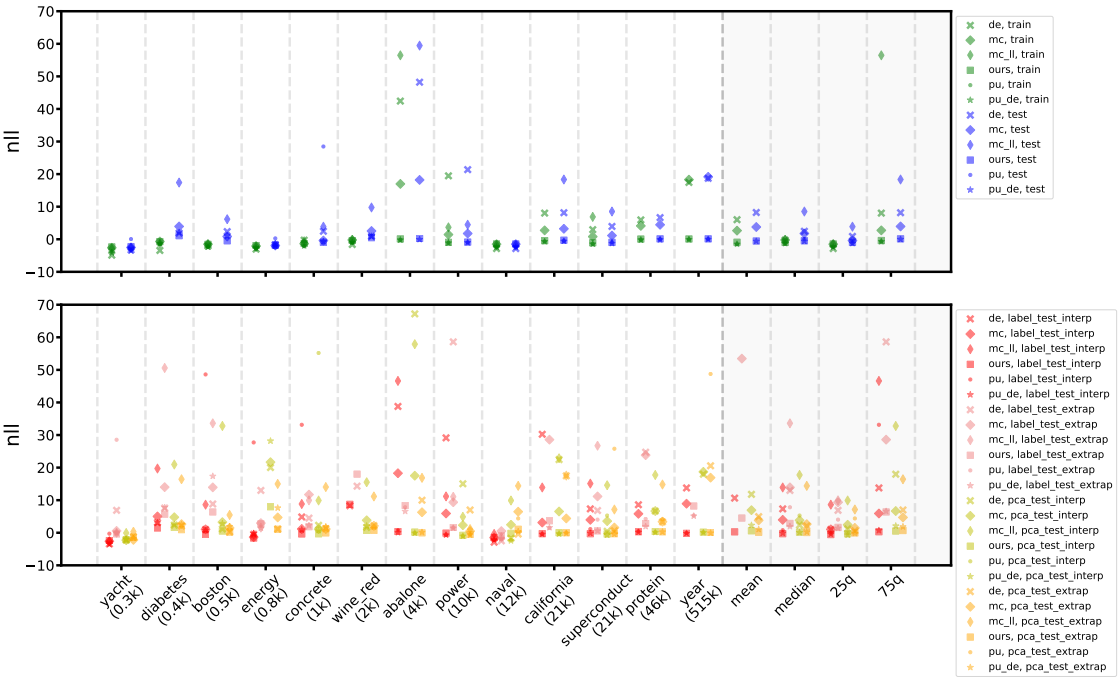

Figure 8: Negative log-likelihoods (NLLs) for 13 UCI regression datasets under i.i.d. conditions (top) and under data shift (bottom). Uncertainty methods are encoded via plot marker, data splits via color. Each plot point corresponds to a cross-validated trained network. Summarizing statistics (rhs) are indicated by a light grey background.

as a crucial requirement for an uncertainty method. SML-based models yield the highest stability in that sense w.r.t. NLL.

Wasserstein distances Studying Wasserstein distances, we again observe equally strong results for PU-DE and SML on train and test data (see column 'mean' in top panel of Fig. 9). PU in contrast possesses a large generalization gap thus yielding weak test set performances. MC, MC-LL, and DE behave consistently weak on train and test sets with MC-LL even falling out of plot range. Under data shift (bottom panel of Fig. 9), the picture remains similar. PU-DE and SML are in the lead and comparably strong with the exception of PU-DE on label-interpolate and label-extrapolate test data ('mean' column). As for NLL, we find these mean values of PU-DE to be significantly above the respective median values indicating again weaknesses in the stability of parametric ensembles.

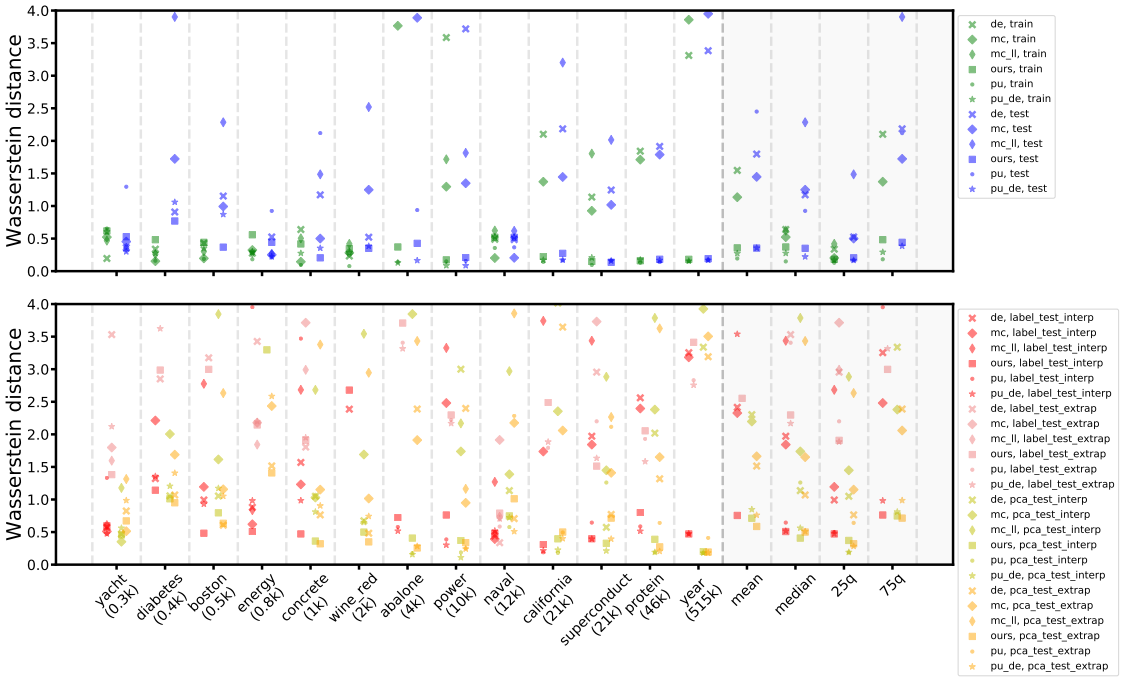

Figure 9: Wasserstein distances for 13 UCI regression datasets under i.i.d. conditions (top) and under data shift (bottom). Uncertainty methods are encoded via plot marker, data splits via color. Summarizing statistics (rhs) are indicated by a light grey background.

Slight overestimation of small uncertainties for SML The second-moment loss yields weak results on 'yacht', 'energy' and 'naval', the three easiest datasets if measured by test set RMSE, compare Fig. 8. On these datasets neither aleatoric uncertainty nor modelling residuals play a mayor role. In such cases, the second-moment loss seems to cause slightly overshooting uncertainty estimates (compare edges of Fig. 1 for a visual clue), likely due to its sub-network 'repulsion'. Back-propagating not only through $f_{\tilde{\theta}}$ but also through the full network f_θ in L_{sml} might mitigate this effect. In practice, slight overestimations

of small uncertainties might be acceptable. In contrast, our method performs consistently strong on all more challenging datasets ('california', 'superconduct', 'protein', 'year'). A beneficial characteristic for virtually any real-world task.

B.3. Residual-uncertainty scatter plots

Visual inspection of uncertainties can be helpful to understand their qualitative behaviour. We scatter model residuals $\mu_i - y_i$ (respective x-axis in Fig. 11) against model uncertainties σ_i (resp. y-axis in Fig. 11). 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. 10 visualizes this hypothetical ideal. Geometrically, the described Gaussian properties imply that 99.7% of all scatter points, e.g.,

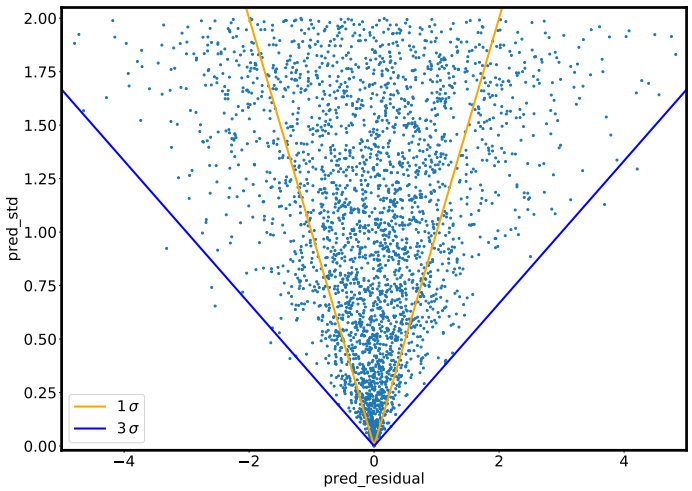

Figure 10: Prediction residuals (x-axis) and predictive uncertainty (y-axis) for a *hypothetical* ideal uncertainty mechanism. The Gaussian errors are matched by Gaussian uncertainty predictions at the exact same scale. 68.3% of all uncertainty estimates (plot points) lie above the orange 1σ-lines and 99.7% of them above the blue 3σ-lines.

in Fig. 11 should lie above the blue 3σ lines and 68.3% of them above the yellow 1σ lines. For 'abalone' test data (third row of Fig. 11), PU and SML qualitatively fulfil this requirement while MC and DE tend to underestimate uncertainties. This finding is in accordance with our systematic evaluation. For abalone and superconduct, we qualitatively find PU, PU-DE and SML-trained networks to provide more realistic uncertainties compared to MC, MC-LL and DE (see Fig. 11). The naval dataset poses an exception in this regard as all uncertainty methods lead to comparably convincing uncertainty estimates. The small test RMSEs of all methods on naval (see Fig. 7) indicate relatively small aleatoric uncertainties and model

Table 2: Regression performance and uncertainty quality of networks with different uncertainty mechanisms. The scores are calculated on the test sets of 13 UCI datasets.

measure	**dataset**	**MC**	**MC-LL**	**Ours**	**PU**	**PU-DE**	**DE**
RMSE (↓)	yacht	0.08	0.07	0.08	0.07	0.07	0.05
NLL (↓)	yacht	−2.53	−2.68	−2.3	0.05	−3.53	−3.26
ECE (↓)	yacht	0.96	0.78	1.10	0.90	0.74	0.63
WS (↓)	yacht	0.45	0.36	0.53	1.30	0.30	0.36
RMSE (↓)	diabetes	0.79	0.89	0.84	0.82	0.78	0.89
NLL (↓)	diabetes	3.93	17.4	1.10	316.53	2.14	1.90
ECE (↓)	diabetes	0.99	1.26	0.72	1.15	0.78	0.71
WS (↓)	diabetes	1.72	3.90	0.77	9.24	1.06	0.91
RMSE (↓)	boston	0.32	0.35	0.33	0.34	0.33	0.33
NLL (↓)	boston	0.85	6.15	−0.48	144.2	1.04	2.35
ECE (↓)	boston	0.76	1.05	0.53	1.13	0.70	0.69
WS (↓)	boston	0.99	2.29	0.37	5.57	0.87	1.15
RMSE (↓)	energy	0.10	0.09	0.08	0.21	0.18	0.08
NLL (↓)	energy	−1.93	−1.7	−1.83	0.26	−2.09	−1.65
ECE (↓)	energy	0.54	0.45	0.83	0.68	0.42	0.52
WS (↓)	energy	0.25	0.27	0.44	0.93	0.22	0.52
RMSE (↓)	concrete	0.25	0.27	0.25	0.29	0.26	0.25
NLL (↓)	concrete	−0.4	3.86	−0.93	28.47	−0.72	2.45
ECE (↓)	concrete	0.48	0.73	0.44	0.75	0.41	0.65
WS (↓)	concrete	0.50	1.49	0.20	2.12	0.35	1.17
RMSE (↓)	wine-red	0.77	0.87	0.80	0.81	0.77	0.83
NLL (↓)	wine-red	2.53	9.76	0.49	14572.96	0.94	0.87
ECE (↓)	wine-red	0.73	0.93	0.41	0.61	0.37	0.41
WS (↓)	wine-red	1.25	2.52	0.35	10.59	0.38	0.52
RMSE (↓)	abalone	0.69	0.68	0.69	0.67	0.68	0.68
NLL (↓)	abalone	18.21	59.45	0.24	610.84	−0.07	48.21
ECE (↓)	abalone	1.29	1.44	0.38	0.27	0.29	1.39
WS (↓)	abalone	3.89	6.85	0.43	0.94	0.16	5.79
RMSE (↓)	naval	0.14	0.08	0.09	0.25	0.22	0.04
NLL (↓)	naval	−1.51	−1.82	−1.45	−2.43	−2.37	−2.86
ECE (↓)	naval	0.37	0.64	0.94	0.56	0.98	0.85
WS (↓)	naval	0.20	0.62	0.52	0.37	0.52	0.48
RMSE (↓)	power	0.23	0.23	0.22	0.23	0.22	0.22
NLL (↓)	power	1.77	4.47	−0.87	−0.97	−1.02	21.37
ECE (↓)	power	0.79	0.89	0.18	0.17	0.15	1.18
WS (↓)	power	1.35	1.81	0.21	0.16	0.09	3.72
RMSE (↓)	california	0.44	0.44	0.44	0.64	0.68	0.43
NLL (↓)	california	3.21	18.34	−0.28	−0.48	−0.58	8.15
ECE (↓)	california	0.77	1.07	0.24	0.24	0.27	0.90
WS (↓)	california	1.45	3.20	0.27	0.17	0.17	2.18
RMSE (↓)	superconduct	0.32	0.32	0.32	0.36	0.34	0.30
NLL (↓)	superconduct	1.13	8.51	−0.96	−0.87	−1.27	3.93
ECE (↓)	superconduct	0.59	0.74	0.20	0.15	0.25	0.55
WS (↓)	superconduct	1.02	2.01	0.14	0.16	0.16	1.25
RMSE (↓)	protein	0.66	0.65	0.66	0.76	0.70	0.62
NLL (↓)	protein	4.45	1.4×10^6	0.12	0.02	−0.11	6.65
ECE (↓)	protein	0.89	1.07	0.24	0.22	0.30	0.84
WS (↓)	protein	1.79	7.9×10^5	0.18	0.14	0.17	1.91
RMSE (↓)	year	0.79	0.80	0.79	0.79	0.78	0.77
NLL (↓)	year	19.15	5.7×10^5	0.12	0.05	−0.01	18.69
ECE (↓)	year	1.33	1.48	0.24	0.27	0.28	1.19
WS (↓)	year	3.95	6.7×10^5	0.19	0.17	0.17	3.38

residuals. Epistemic uncertainty might thus be a key driving factor and coherently MC, MC-LL and DE perform well.

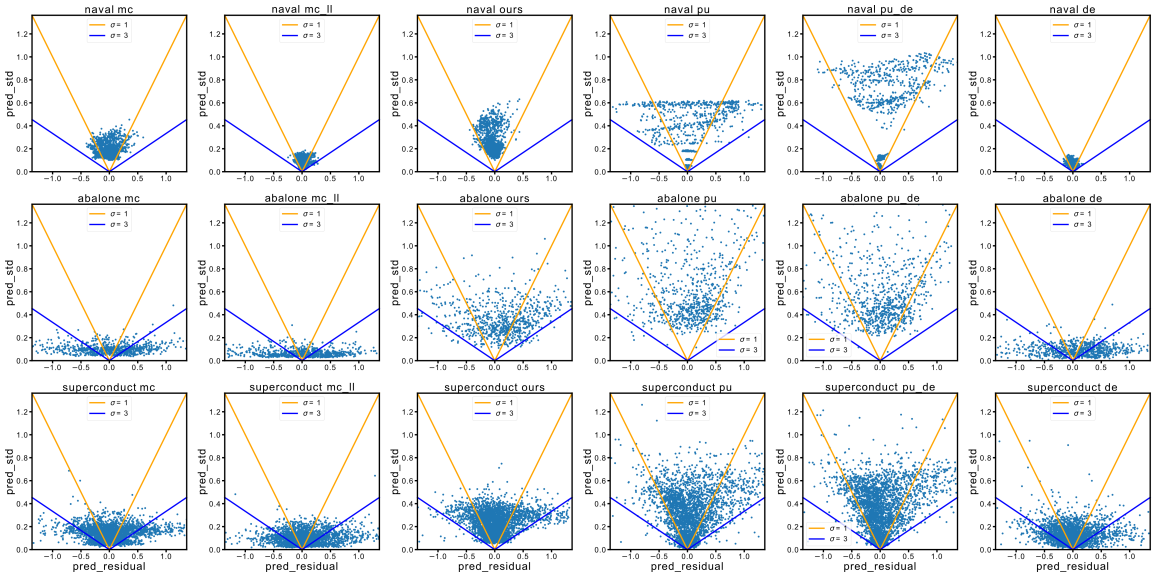

Figure 11: Prediction residuals (respective x-axis) and predictive uncertainty (respective y-axis) for different uncertainty mechanisms (columns) and datasets (rows). Each light blue dot in each plot corresponds to one test data point. Realistic uncertainty estimates should lie mostly above the blue 3σ-lines.The datasets naval, abalone and superconduct are shown, from top to bottom.

The hypothetical ideal residual-uncertainty scatter plot we use in Fig. 10 is generated as follows: We draw 3000 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. But even in this perfect case, Pearson correlation between uncertainty estimates and absolute residuals is only approximately 55%.

Appendix C. Stability w.r.t. hyper-parameter β

Here, we analyze the impact of the SML-parameter β on the uncertainty quality of accordingly trained models. For $\beta = 0.1, 0.25, 0.5, 0.75, 0.9$, we observe only relatively small differences in both ECE (see Fig. 12) and Wasserstein distance (see Fig. 13). $\beta = 0.5$ provides (by a small margin) the best average test set performance in both scores. However, the best-performing β-value for an individual dataset can vary.

Experiments with $\beta \gg 1$ (not shown here) cause non-convergent training in many cases as primarily uncertainty quality is optimized at the expense of task performance. The opposite extreme case is $\beta = 0$, i.e. network optimization without any dropout mechanism. Applying dropout at inference will therefore cause uncontrolled random fluctuations around the network prediction.

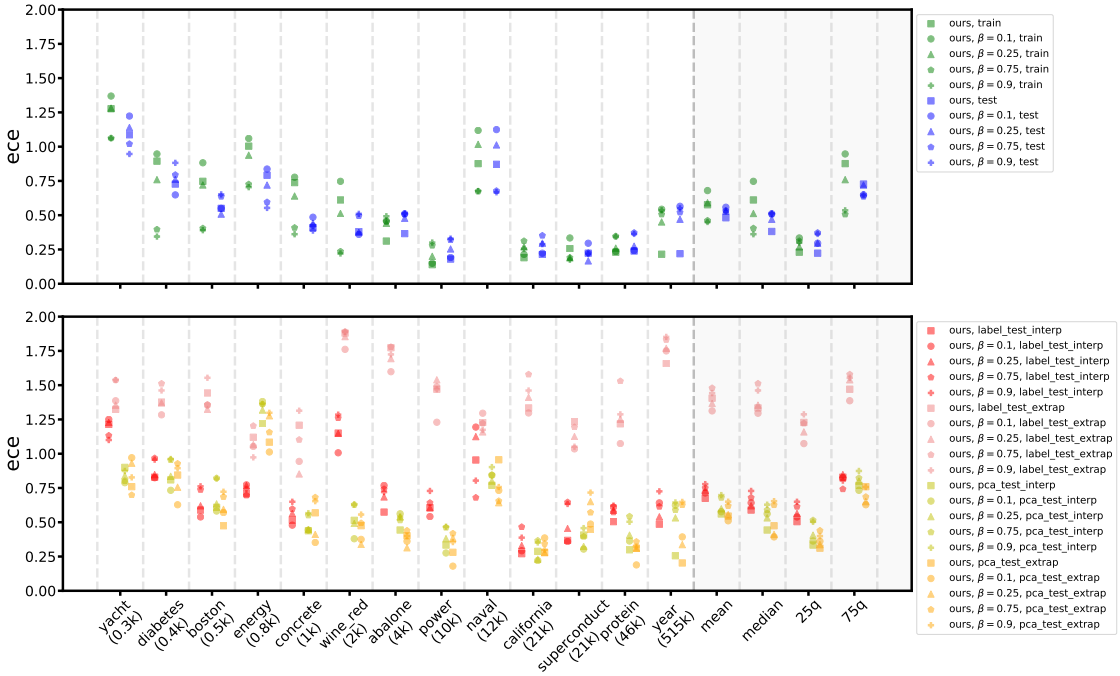

Figure 12: Expected calibration errors (ECEs) for SML-trained networks with hyper-parameters $\beta = 0.1, 0.25, 0.5, 0.75, 0.9$. We consider 13 UCI regression datasets under i.i.d. conditions (top) and under data shift (bottom). β-values are encoded via plot marker, data splits via color. Each plot point corresponds to a cross-validated trained network. Summarizing statistics (rhs) are indicated by a light grey background.

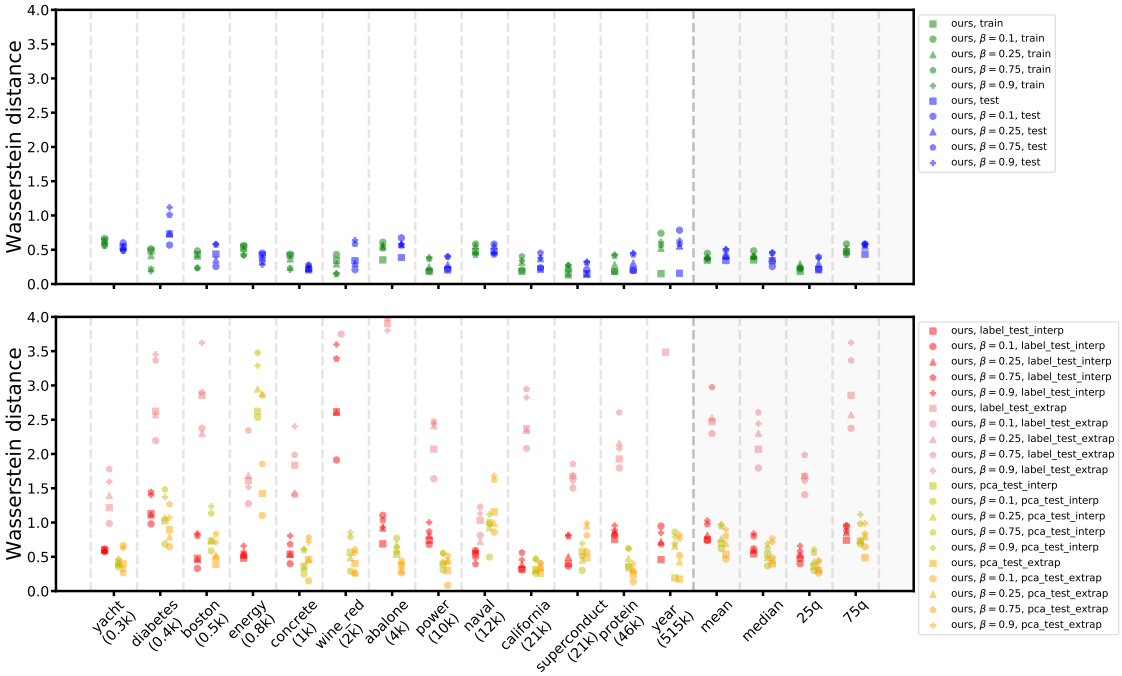

Figure 13: Wasserstein distances for SML-trained networks with hyper-parameters $\beta = 0.1$, $0.25, 0.5, 0.75, 0.9$. We consider 13 UCI regression datasets under i.i.d. conditions (top) and under data shift (bottom). β-values are encoded via plot marker, data splits via color. Each plot point corresponds to a cross-validated trained network. Summarizing statistics (rhs) are indicated by a light grey background.

Appendix D. In-depth investigation of uncertainty measures

D.1. Dependencies between uncertainty measures

All uncertainty-related measures (NLL, ECE, Wasserstein distance) 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. The empirically observed dependencies between these uncertainty measures are visualized in Fig. 14. Additionally to Wasserstein distances, we consider Kolmogorov-Smirnov (KS) distances (Stephens, 1974) on normalized residuals there. It estimates a distance between the sample 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 samples that strongly deviate from a standard Gaussian one.

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 data splits in Fig. 14 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: MC is 'diamond', MC-LL is 'thin diamond', DE is 'cross', PU is 'point', PU-DE is 'pentagon' and second-moment loss is 'square'. Some Wasserstein distances lie above the x-axis cut-off and are thus not visualized.

D.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 contraintuitive 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, $\mathrm{NLL}(\tilde{x}_2) \approx 1.4 > 0.9 \approx \mathrm{NLL}(\tilde{x}_1)$, where

$$\mathrm{NLL}(y) := \log \sqrt{2\pi\sigma^2} + \frac{1}{N} \sum_{i=1}^{N} \frac{(y_i - \mu)^2}{2\sigma^2} \,. \tag{5}$$

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 $\tilde{x}_i$. 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

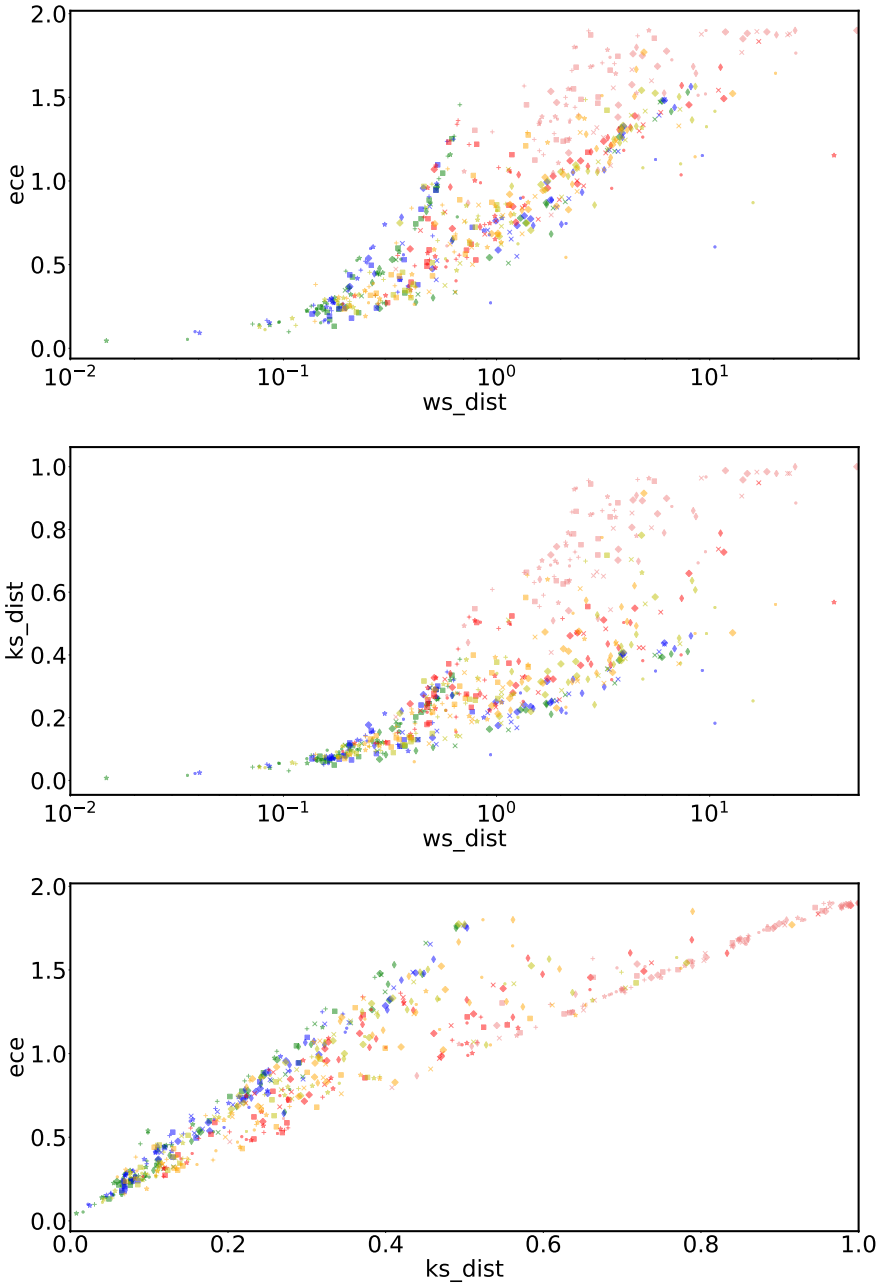

Figure 14: Dependencies between the three uncertainty measures ECE, Wasserstein distance and Kolmogorov-Smirnov distance. Uncertainty methods are encoded via plot markers, data splits via color. Datasets are not encoded and cannot be distinguished (see text for more details). Each plot point corresponds 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.

"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 equation (5) leads to $\mathrm{NLL}(z_2) = \mathrm{NLL}(z_1) + \log(\sigma_2/\sigma_1)$ 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 heteroskedastic 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.