Detecting Failure Modes in Image Reconstructions with Interval Neural Network Uncertainty

Luis Oala [*1] **Cosmas Heiß** [*2] **Jan Macdonald** [*2] **Maximilian März** [*2] **Gitta Kutyniok** [2] **Wojciech Samek** [1]

Abstract

The quantitative detection of failure modes is important for making deep neural networks reliable and usable at scale. We consider three examples for failure modes in image reconstruction problems and demonstrate the potential of uncertainty quantification as a fine-grained alarm system. We propose a deterministic, modular and lightweight approach, called Interval Neural Networks, that produces fast and easy to interpret uncertainty scores which improve the detection of failure modes across four out of five image reconstruction experiments.

1. Introduction

Deep neural networks (DNNs) play an important role in many computational imaging tasks. Conceptually, these tasks can often be modelled as finite-dimensional linear inverse problems $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{\eta}$ where $\boldsymbol{y} \in \mathbb{R}^n$ is the unknown signal of interest, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ denotes the forward operator representing a physical measurement process, and $\boldsymbol{\eta} \in \mathbb{R}^m$ is modelling noise in the measurements. Solving the inverse problem amounts to computing an approximate reconstruction of $\boldsymbol{y}$ from its observed measurements $\boldsymbol{x}$[1]. Many popular applications such as image translation (domain mapping), super-resolution, denoising or image synthesis, fall in this problem category. Medical imaging technologies such as computed tomography (CT) or magnetic resonance imaging in particular are oft cited examples for the promise of DNN image reconstruction technology, see (Kang et al., 2017; Jin et al., 2017; Hammernik et al., 2018; Adler & Öktem, 2018; Arridge et al., 2019) for recent examples. Despite this progress, it has been demonstrated that the reliability of DNN based reconstructions are a concern (Adler & Öktem, 2018; Ardizzone et al., 2018; Huang et al., 2018; Antun et al., 2019; Gottschling et al., 2020) when compared to traditional, model-based approaches. Erroneous artifacts in the output image can be hard to detect when they blend in well with the rest of the output image. What is more, local reliability assessment of individual outputs can quickly become expensive and stifles the deployment of DNNs at scale. In this work we explore the automatic detection of DNN failure modes (Dietterich, 2019) using uncertainty quantification (UQ). We consider three failure modes during inference: adversarial noise artifacts, atypical input artifacts and prediction errors on benign inference data. We present a deterministic, modular and fast approach, called Interval Neural Networks (INN), to obtain uncertainty scores which improve the detection of failure modes across four out of five inverse problem experiments.

2. Related Work

Whereas a number of methods from classical statistical learning theory, such as Gaussian processes and approximations thereof (Denker et al., 1987; MacKay, 1992; Neal, 1996; Williams, 1996), come with built-in uncertainty estimates, DNNs have been limited in this regard. A surge of efforts to treat neural networks from a variational perspective (Barber & Bishop, 1998; Srivastava et al., 2014; Blundell et al., 2015; Kingma et al., 2015) started to change that. In addition, there exist strands of research in deep learning explicitly occupied with the detection of failure modes caused by adversarial and out of distribution (OoD) inputs. These include Maximum Mean Discrepancy, Kernel Density Estimation and other tools, see (Carlini & Wagner, 2017) or the Minimum Covariance Determinant method (Rousseeuw, 1984), Support Vector Data Description (Tax & Duin, 2004), ODIN (Liang et al., 2019), Outlier Exposure (Hendrycks et al., 2019), or detection in latent space (Gómez-Bombarelli et al., 2018). The detection of adversarial and OoD inputs in these works is typically done in the classification setting. We emphasize that image-to-image regression is a fundamentally different task: While classification is inherently discontinuous, image-to-image regression addresses a problem that allows for stable reconstruction

[*]Equal contribution [1]Machine Learning Group, Fraunhofer Institute for Telecommunications, Berlin, Germany [2]Department of Mathematics, Technical University of Berlin, Berlin, Germany. Correspondence to: Luis Oala <luis.oala@hhi.fraunhofer.de>.

Presented at the ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning.

[1]For this manuscript we use the $(\boldsymbol{x}, \boldsymbol{y})$ input-output notation common in machine learning.

methods in many cases, e.g. by sparse regularization. Furthermore, we are not interested in a crude, outright rejection of data points in the *input space* but rather seek to obtain fine-grained information about erroneous artifacts in the *output space*. More closely related to our goal is the work of (Gal & Ghahramani, 2016; Kendall & Gal, 2017), Monte Carlo dropout (MCDROP), and (Gast & Roth, 2018), direct variance estimation (PROBOUT), where epistemic and aleatoric uncertainty quantification, respectively, was considered for segmentation and depth-estimation tasks. Hence, we include their approaches as baseline detection methods. See Section E.1 in the appendices for a detailed description.

3. Method

Popular existing UQ frameworks for DNNs rely on placing parametric densities, most commonly a Gaussian density, over DNN parameters or predictions. Our INN method relies on bounding this distribution using intervals. The reason for exploring INNs for the detection of failure modes lies in their flexibility. They are modular and can be constructed *post hoc* for a trained prediction network that may already be in use. Thus INNs do not require to touch or modify the prediction network itself. Given a trained, underlying prediction DNN $\boldsymbol{\Phi}$ we construct an interval neural network around it. A schematic illustration is provided in Appendix A. Finally, the produced intervals are simple to interpret: they provide a range of values a DNN output node may take with exact upper and lower bounds. (Garczarczyk, 2000) have previously investigated the capacity of neural networks with interval *weights* and *biases* for fitting interval valued functions. Note that (Yang & Wu, 2012; Kowalski & Kulczycki, 2017) also explored interval neural networks for robust classification although in their setting the focus is purely on representing the *inputs* or *outputs* as intervals. Our resulting INN is different in that interval bounds are determined for all parameters of the network with the goal of providing uncertainty scores for the outputs. INNs have the following mechanisms that deviate from the usual arithmetic. The forward propagation of a component-wise interval valued input $[\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]$ through the INN can be expressed similarly to standard feed-forward neural networks but using interval arithmetic instead. For interval valued weight matrices $\left[\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}\right]$ and bias vectors $\left[\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}\right]$ the propagation through the l-th layer can be expressed by

$$[\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]^{(l+1)} = \varrho\left(\left[\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}\right]^{(l)} [\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]^{(l)} + \left[\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}\right]^{(l)}\right).$$

For positive values of $[\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]^{(l)}$, for example when using a non-negative activation function like ReLU, we can simplify this operation to

$$\overline{\boldsymbol{x}}^{(l+1)} = \varrho\Big(\min\Big\{\overline{\boldsymbol{W}}^{(l)}, 0\Big\}\underline{\boldsymbol{x}}^{(l)} + \max\Big\{\overline{\boldsymbol{W}}^{(l)}, 0\Big\}\overline{\boldsymbol{x}}^{(l)} + \overline{\boldsymbol{b}}^{(l)}\Big),$$
$$\underline{\boldsymbol{x}}^{(l+1)} = \varrho\Big(\max\Big\{\underline{\boldsymbol{W}}^{(l)}, 0\Big\}\underline{\boldsymbol{x}}^{(l)} + \min\Big\{\underline{\boldsymbol{W}}^{(l)}, 0\Big\}\overline{\boldsymbol{x}}^{(l)} + \underline{\boldsymbol{b}}^{(l)}\Big),$$

where the maximum and minimum functions are applied component-wise. Assuming $\underline{\boldsymbol{x}}^{(l)} = \overline{\boldsymbol{x}}^{(l)} =: \boldsymbol{x}^{(l)}$ for the input layer before the first ReLU the same operation can be represented as follows to also process negative values:

$$\overline{\boldsymbol{x}}^{(l+1)} = \varrho\Big(\overline{\boldsymbol{W}}^{(l)} \max\{\boldsymbol{x}^{(l)}, 0\} + \underline{\boldsymbol{W}}^{(l)} \min\{\boldsymbol{x}^{(l)}, 0\} + \overline{\boldsymbol{b}}^{(l)}\Big),$$
$$\underline{\boldsymbol{x}}^{(l+1)} = \varrho\Big(\underline{\boldsymbol{W}}^{(l)} \max\{\boldsymbol{x}^{(l)}, 0\} + \overline{\boldsymbol{W}}^{(l)} \min\{\boldsymbol{x}^{(l)}, 0\} + \underline{\boldsymbol{b}}^{(l)}\Big).$$

These formulas can then easily be used in existing deep learning frameworks to optimize the bounds of the interval parameters using backpropagation. As we want the output intervals to contain the target values after training, we define the interval loss to be zero if a target lies inside the interval and the squared distance to the interval boundary if it lies outside the interval. As this alone would lead to output intervals expanding until they cover the whole range of target values, we additionally employ a linear penalty on the interval size. For the data set $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^m$ consisting of inputs $\boldsymbol{x}_i \in \mathcal{X}$ and targets $\boldsymbol{y}_i \in \mathcal{Y}$, this leads to the following INN loss. Here, $\overline{\boldsymbol{\Phi}}\colon \mathcal{X} \to \mathcal{Y}$, $\underline{\boldsymbol{\Phi}}\colon \mathcal{X} \to \mathcal{Y}$ are the functions that map the input to the upper and the lower interval bounds in the output of the INN:

$$\mathcal{L}(\underline{\boldsymbol{\Phi}}, \overline{\boldsymbol{\Phi}}) = \sum_{i=1}^{m} \max\{\boldsymbol{y}_i - \overline{\boldsymbol{\Phi}}(\boldsymbol{x}_i), 0\}^2 + \max\{\underline{\boldsymbol{\Phi}}(\boldsymbol{x}_i) - \boldsymbol{y}_i, 0\}^2 + \beta \cdot \big(\overline{\boldsymbol{\Phi}}(\boldsymbol{x}_i) - \underline{\boldsymbol{\Phi}}(\boldsymbol{x}_i)\big). \quad (1)$$

The tightness parameter $\beta > 0$ determines how outlier-sensitive the intervals are trained. In practice, choosing β similar to the mean absolute error made by the underlying prediction network works well. The output uncertainty estimates of an INN are then given by the width of the prediction interval, i.e., $\boldsymbol{u}_{\text{INN}}(\boldsymbol{x}) = \overline{\boldsymbol{\Phi}}(\boldsymbol{x}) - \underline{\boldsymbol{\Phi}}(\boldsymbol{x})$. In terms of run time, INNs scale linearly in the number of underlying prediction DNN operations K with a constant factor of 2. In contrast, (Gal & Ghahramani, 2016) scales linearly with a factor T which is proportional to the number of stochastic forward passes and $T = 10$ is recommended by the authors as a rule of thumb.

The coverage bounds represented by the intervals are theoretically justified. Assuming the loss in Equation (1) is optimized during training to yield an INN for which the expected gradient with respect to the data distribution is zero, we can give the following bound using the Markov Inequality: Let the training data be represented by the random variable $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ distributed on $\mathcal{X} \times \mathcal{Y}$ according to the training data distribution. Then, for any $\lambda > 0$, we obtain

$$\mathbb{P}(\underline{\boldsymbol{\Phi}}(\boldsymbol{x}^*) - \lambda\beta < \boldsymbol{y}^* < \overline{\boldsymbol{\Phi}}(\boldsymbol{x}^*) + \lambda\beta \,|\boldsymbol{x}^*) \geq 1 - \frac{1}{\lambda},$$

i.e. for any input and target sampled form the distribution of training samples, the probability of the target lying inside

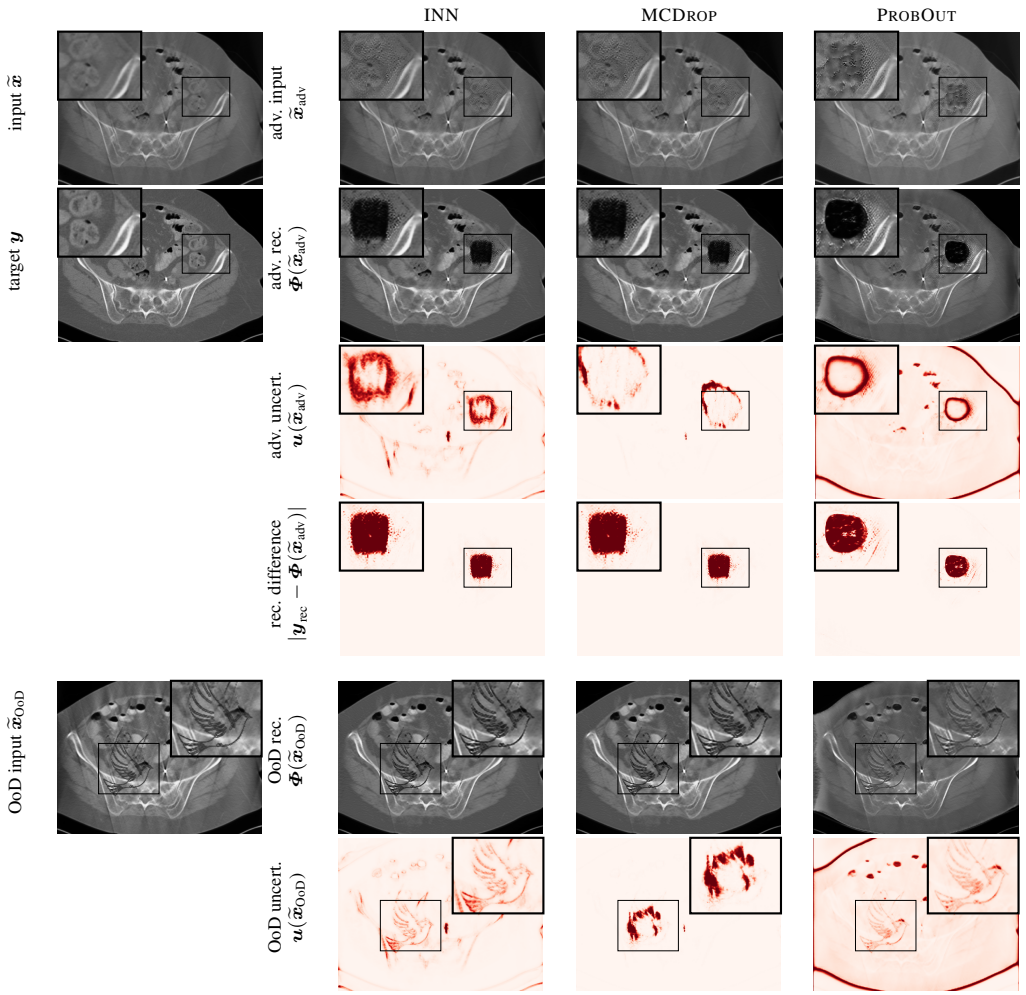

Figure 1. Results of three UQ methods for the ADVDETECT and ARTDETECT experiments for one exemplary data sample of the Limited Angle CT task. The plotting windows are slightly adjusted for better contrast.

the predicted interval, which is enlarged by $\lambda\beta$, is at least $1 - \frac{1}{\lambda}$. See Appendix C for a full proof. We devised a toy inverse problem 1D Deconvolution (1DDECONV), inspired by a one-dimensional deconvolution, to illustrate this point. The task for the DNN is to reconstruct the original signal from the blurred measurements (see top of Figure 4 for an illustration). Indeed, we can observe that 89% of ground truth values lie inside the output intervals. This type of assessment is not possible with the popular Gaussian-based UQ methods. Further, INNs are able to capture noise contained in the inputs using the interval bias parameters in the last layer: the average interval size increases with increasing noise levels as summarized in Figure 5 for the controlled 1DDECONV simulation. Note how the MCDROP approach is not able to capture these deviations in the output. This difference can also be observed in the right column of Figure 4 for the 1DDECONV task with independent Gaussian noise ($\sigma = 0.05$) added to the inputs and targets. Thus, we place INNs at the intersection of epistemic and aleatoric uncertainty: we are able to capture and propagate model uncertainty via interval weights while at the same time accommodating observation noise. Finally, if the prediction from the underlying network lies closer to one boundary of the output interval, one can infer directional information about the error. The directional information contained in INN uncertainty scores leads to direction accuracy that is 12 to 25 percentage points above chance for the 1DDECONV task as documented in Figure 6. This is in contrast to symmetric uncertainty score approaches like MCDROP and PROBOUT. We note that it is possible to explore asymmetry, e.g. via exponential family distributions (Wang et al., 2016), and intervals, e.g. via quantile regression (Koenker & Hallock, 2001; Rodrigues & Pereira, 2020), in the probabilistic setting, too, but in contrast to INNs this would imply substantial modifications to and retraining of the underlying prediction network. For completeness we provide a formal comment on how to treat INNs from a probabilistic, Bayesian perspective in Appendix D. Next, we demonstrate how these mechanisms can be successfully used for the detection of failure modes in image reconstructions.

Table 1. Mean test results (± standard deviation) averaged over three experimental runs. Pearson correlation coefficients for the Adversarial Artifact Detection and Atypical Artifact Detection experiments and PWCC with MSE for the EC experiment.

	AdvDetect		ArtDetect		EC	
UQ Method	CT	Denoise	CT	Denoise	PWCC	MSE
INN	$\mathbf{0.56 \pm 0.05}$	0.77 ± 0.008	$\mathbf{0.52 \pm 0.03}$	$\mathbf{0.69 \pm 0.006}$	$\mathbf{2211 \pm 403}$	$7.4 \pm 0.65 \times 10^{-4}$
MCDrop	0.28 ± 0.02	0.20 ± 0.001	0.26 ± 0.01	0.44 ± 0.02	2170 ± 513	$7.4 \pm 0.65 \times 10^{-4}$
ProbOut	0.48 ± 0.12	$\mathbf{0.81 \pm 0.002}$	0.34 ± 0.04	0.44 ± 0.01	190 ± 28	$6.7 \pm 2 \times 10^{-3}$

4. Experiments

Two image reconstruction tasks are considered for the failure mode detection experiments: using a DNN to enhance the reconstructions of subsampled CT measurements (CT) and using a DNN to remove Gaussian noise from grayscale images (Denoise). We consider a straight-forward approach to solving the reconstruction tasks, which is based on post-processing a standard model-based inversion by a neural network (Zhang et al., 2016; Kang et al., 2017; Jin et al., 2017). Thus, the reconstruction is given by $\boldsymbol{y}_{\text{rec}} = \boldsymbol{\Phi}(\boldsymbol{A}^\dagger \boldsymbol{x})$ where $\boldsymbol{\Phi}\colon \mathbb{R}^n \to \mathbb{R}^n$ denotes the prediction network (trained to minimize the loss $\|\boldsymbol{y} - \boldsymbol{\Phi}(\boldsymbol{A}^\dagger \boldsymbol{x})\|_2^2$), $\widetilde{\boldsymbol{x}} = \boldsymbol{A}^\dagger \boldsymbol{x}$ denotes the model-based inversion and $\boldsymbol{A}^\dagger$ the non-learned model-based inversion operator (filtered back-projection (Natterer, 2001) for the CT task and identity for the Denoise task). More details about the data and neural network architectures used for both tasks can be found Appendix E. We consider three failure modes during inference: adversarial noise artifacts, atypical input artifacts and prediction errors on benign inference data. First, in the Adversarial Artifact Detection (AdvDetect) experiments we assess the capacity of UQ to capture artifacts in the output that were caused by adversarial noise. To that end, we create perturbed inputs for each measurement sample $\boldsymbol{y}$ in the test set by employing the box-constrained L-BFGS algorithm (Byrd et al., 1995) to minimize

$$\|\boldsymbol{\Phi}(\widetilde{\boldsymbol{x}}_{\text{adv}}) - \boldsymbol{y}_{\text{adv. tar.}}\|_2^2 + \lambda \|\widetilde{\boldsymbol{x}}_{\text{adv}} - \widetilde{\boldsymbol{x}}\|_2^2, \tag{2}$$

wrt $\widetilde{\boldsymbol{x}}_{\text{adv}} \in [0,1]^n$, where $\boldsymbol{y}_{\text{adv. tar.}}$ represents the adversarial target, and $\lambda \geq 0$ is a parameter for balancing the two terms in (2). Details about the noise generation process are documented in Appendix E. In order to assess the detection capacity for this failure mode, the different UQ schemes are then used to produce uncertainty heatmaps for the generated adversarial inputs. A quantitative evaluation is carried out by computing the mean Pearson correlation coefficient between the pixel-wise change in the uncertainty heatmaps $|\boldsymbol{u}(\widetilde{\boldsymbol{x}}) - \boldsymbol{u}(\widetilde{\boldsymbol{x}}_{\text{adv}})|$ and the change of reconstructions $|\boldsymbol{y}_{\text{rec}} - \boldsymbol{\Phi}(\widetilde{\boldsymbol{x}}_{\text{adv}})|$. The results are summarized in Table 1 and illustrated in Figures 1 and 9. We observe that both INN and ProbOut are able to detect the image region of adversarial perturbations, with ProbOut achieving slightly higher correlations in the denoising task and INN having the highest correlation in the CT task. This shows that both methods are able to visually highlight the effect that almost imperceptible input perturbations can have on the reconstructions. The second experiment, Atypical Artifact Detection (ArtDetect), is designed analogous to the setup described by (Gottschling et al., 2020), i.e., an atypical artifact, which was not present in the training data, is randomly placed in the input to produce $\widetilde{\boldsymbol{x}}_{\text{OoD}}$. For the Denoise task this is achieved by locally changing the noise distribution. For the CT task the silhouette of a peace dove is inserted in each image of the test set; see Figure 1. As for the previous experiments, please see Appendix E for details on the data manipulation. A quantitative evaluation is carried out by computing the mean Pearson correlation coefficient between the change in the uncertainty heatmaps $|\boldsymbol{u}(\widetilde{\boldsymbol{x}}) - \boldsymbol{u}(\widetilde{\boldsymbol{x}}_{\text{OoD}})|$ and a binary mask marking the region of change in the inputs. The results are summarized in Table 1 and illustrated in Figures 1 and 9. All three UQ methods are correlated with the input change, however INN achieves the highest correlation in both the Denoise and CT task. This shows that UQ in general, and INNs in particular, can serve as a warning system for inputs containing atypical features that might otherwise lead to unnoticed and possibly erroneous reconstruction artifacts. Finally, in the third experiment we evaluate how helpful UQ scores are in tracking the actual prediction error on benign inputs on the more challenging CT task. The *performance weighted correlation coefficient* (PWCC) of the uncertainty scores of each UQ method and the absolute prediction errors are compared. Performance weighted means the correlation coefficient is weighted by the mean squared error. This is necessary to discourage rewards for poor prediction models with high uncertainties everywhere. For a datapoint $(\widetilde{\boldsymbol{x}}_i, \boldsymbol{y}_i)$ and a corresponding uncertainty map $\boldsymbol{u}_i$, the performance weighted correlation coefficient (PWCC) is thus computed as:

$$\text{PWCC}(\widetilde{\boldsymbol{x}}_i, \boldsymbol{y}_i, \boldsymbol{u}_i) = \frac{\text{corr}(|\boldsymbol{\Phi}(\widetilde{\boldsymbol{x}}_i) - \boldsymbol{y}_i|, \boldsymbol{u}_i)}{\text{MSE}(\boldsymbol{\Phi}(\widetilde{\boldsymbol{x}}_i), \boldsymbol{y}_i)}.$$

The interval size (INN) and standard deviation (MCDrop and ProbOut) in the output are used as uncertainty scores. As documented in Table 1 INN and MCDrop are able to detect the error prone regions. Apart from this, the INN method also highlights other regions in the image with high local intensity variations, see Figure 10 for an example. In addition, we can again observe the INNs performance with respect to coverage and directional information. A total $76 \pm 6\%$ of test targets in the CT data are contained in the produced intervals. In Figure 7 the directional accuracy of the INN becomes more pronounced from 57% up to 72% as the interval direction threshold grows.

Code

Code will be made available at https://github.com/luisoala/inn.

Acknowledgements

The authors would like to thank Sören Becker and anonymous reviewers for their helpful feedback on a first draft of this manuscript. M.M. acknowledges support by the DFG Priority Programme DFG-SPP 1798 Grants KU 1446/21 and KU 1446/23.

References

Adler, J. and Öktem, O. Learned Primal-dual Reconstruction. *IEEE T. Med. Imaging*, 37(6):1322–1332, 2018.

Adler, J. and Öktem, O. Deep Bayesian Inversion. *arXiv:1811.05910*, November 2018. arXiv: 1811.05910.

Antun, V., Renna, F., Poon, C., Adcock, B., and Hansen, A. C. On instabilities of deep learning in image reconstruction - does AI come at a cost? *CoRR*, abs/1902.05300, 2019.

Ardizzone, L., Kruse, J., Wirkert, S. J., Rahner, D., Pellegrini, E. W., Klessen, R. S., Maier-Hein, L., Rother, C., and Köthe, U. Analyzing inverse problems with invertible neural networks. In *International Conference on Learning Representations*, 2018.

Arridge, S., Maass, P., Öktem, O., and Schönlieb, C.-B. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019. doi: 10.1017/S0962492919000059.

Barber, D. and Bishop, C. Ensemble learning in bayesian neural networks. In *Generalization in Neural Networks and Machine Learning*, pp. 215–237. Springer Verlag, January 1998.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight Uncertainty in Neural Networks. *arXiv:1505.05424 [cs, stat]*, May 2015. arXiv: 1505.05424.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, 1995.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, pp. 3–14, 2017.

Denker, J. S., Schwartz, D. B., Wittner, B. S., Solla, S. A., Howard, R. E., Jackel, L. D., and Hopfield, J. J. Large Automatic Learning, Rule Extraction, and Generalization. *Complex Systems*, 1, 1987.

Dietterich, T. G. Robust artificial intelligence and robust human organizations. *Frontiers of Computer Science*, 13 (1):1–3, 2019.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.

Garczarczyk, Z. Interval neural networks. In *2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No.00CH36353)*, volume 3, pp. 567–570, Geneva, Switzerland, 2000. Presses Polytech. Univ. Romandes. ISBN 978-0-7803-5482-1. doi: 10.1109/ISCAS.2000.856123.

Gast, J. and Roth, S. Lightweight Probabilistic Deep Networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3369–3378, 2018.

Gottschling, N. M., Antun, V., Adcock, B., and Hansen, A. C. The troublesome kernel: why deep learning for inverse problems is typically unstable? arXiv:2001.01258, 2020.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

Hammernik, K., Klatzer, T., Kobler, E., Recht, M. P., Sodickson, D. K., Pock, T., and Knoll, F. Learning a variational network for reconstruction of accelerated mri data. *Magnetic Resonance in Medicine*, 79(6):3055–3071, 2018.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. *Proc. of the International Conference on Learning Representations*, 2019.

Huang, Y., Würfl, T., Breininger, K., Liu, L., Lauritsch, G., and Maier, A. Some investigations on robustness of deep learning in limited angle tomography. In Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C., and Fichtinger, G. (eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pp. 145–153, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00928-1.

Jin, K. H., McCann, M. T., Froustey, E., and Unser, M. Deep Convolutional Neural Network for Inverse Problems in Imaging. *IEEE Transactions on Image Processing*, 26: 4509–4522, 2017.

Kang, E., Min, J., and Ye, J. C. A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Med. Phys.*, 44(10):360–375, 2017.

Kendall, A. and Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 5580–5590, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Kingma, D. P. and Ba, J. A. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pp. 2575–2583, Cambridge, MA, USA, 2015. MIT Press.

Koenker, R. and Hallock, K. F. Quantile regression. *Journal of economic perspectives*, 15(4):143–156, 2001.

Kowalski, P. A. and Kulczycki, P. Interval probabilistic neural network. *Neural Computing and Applications*, 28 (4):817–834, April 2017. ISSN 0941-0643, 1433-3058. doi: 10.1007/s00521-015-2109-3.

Liang, S., Li, Y., and Srikant, R. Principled detection of out-of-distribution examples in neural networks. *Proceedings of the International Conference on Learning Representations*, 2019.

MacKay, D. J. C. *Bayesian methods for adaptive models*. Phd, California Institute of Technology, 1992.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pp. 416–423, July 2001.

McCollough, C. Tu-fg-207a-04: Overview of the low dose ct grand challenge. *Med. Physs*, 43(6 Part 35):3759–3760, 2016.

Natterer, F. *The Mathematics of Computerized Tomography*. SIAM, 2001.

Neal, R. M. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996. ISBN 0387947248.

Nix, D. A. and Weigend, A. S. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 1, pp. 55–60 vol.1, June 1994. doi: 10.1109/ICNN.1994.374138.

Rodrigues, F. and Pereira, F. C. Beyond expectation: deep joint mean and quantile regression for spatiotemporal problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F. (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pp. 234–241. Springer International Publishing, 2015. ISBN 978-3-319-24574-4.

Rousseeuw, P. J. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

Schmidt, U. and Roth, S. Shrinkage fields for effective image restoration. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2774–2781, 2014.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

Tax, D. M. J. and Duin, R. P. W. Support vector data description. *Mach. Learn.*, 54(1):45–66, 2004.

Wang, H., Xingjian, S., and Yeung, D.-Y. Natural-parameter networks: A class of probabilistic neural networks. In *Advances in Neural Information Processing Systems*, pp. 118–126, 2016.

Williams, C. K. I. Computing with infinite networks. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS'96, pp. 295–301, Cambridge, MA, USA, 1996. MIT Press.

Yang, D. and Wu, W. A smoothing interval neural network. *Discrete Dynamics in Nature and Society*, 2012, 2012.

Zhang, H., Li, L., Qiao, K., Wang, L., et al. Image Prediction for Limited-angle Tomography via Deep Learning with Convolutional Neural Network. *arXiv:1607.08707*, 2016.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Imag. Proc.*, 26: 3142–3155, 2017.

A. INN Schematic Overview

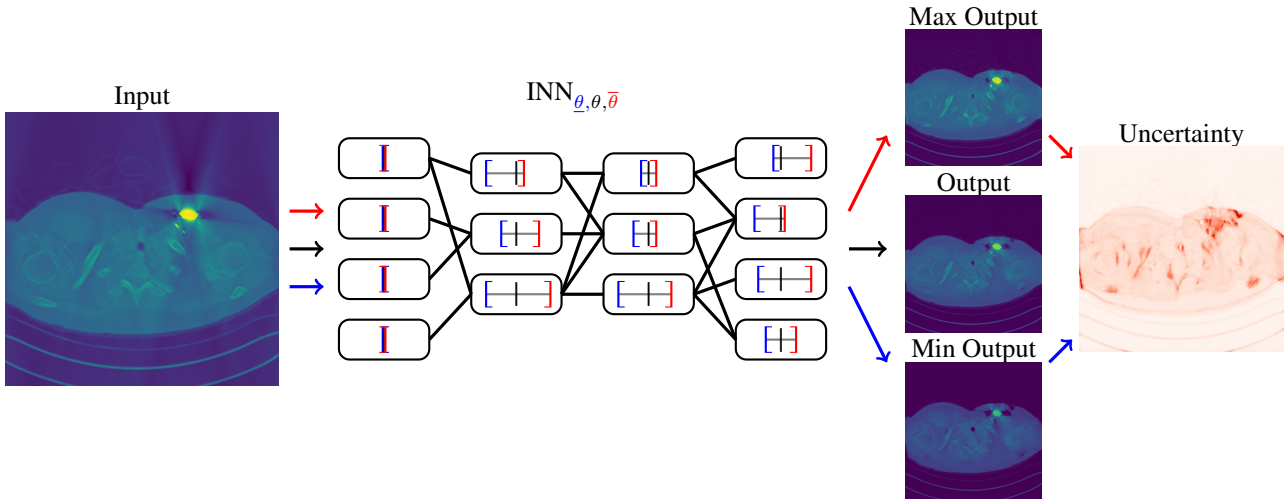

Figure 2. A schematic overview of INNs. The input (left picture) is interpreted as a point interval in the first layer. It is then propagated through the network by the interval valued weights and biases (black connections) using interval arithmetic. As the weights are constrained using the underlying network, the interval valued neurons contain the value from the original prediction (black bar inside intervals) in every layer. In the output, three images are obtained: the original prediction and the images containing the lower and upper bounds. The latter two can then be used to construct a pixel-wise uncertainty score from the interval size.

B. Interval Arithmetic in Neural Networks

We give a derivation of the lower and upper interval bounds $\underline{\boldsymbol{\Phi}}$ and $\overline{\boldsymbol{\Phi}}$ in equation (3) of the main paper. Interval Neural Networks (INNs) make use of interval arithmetic that deviates from customary arithmetic. The forward pass through a ReLU neural network layer $\boldsymbol{x} \mapsto \varrho(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$ in interval arithmetic is as follows: Given a component-wise interval valued input $[\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]$ and interval valued weight matrices $\left[\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}\right]$ and bias vectors $\left[\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}\right]$ the output interval $[\underline{\boldsymbol{z}}, \overline{\boldsymbol{z}}]$ after propagation through the layer is formally expressed as

$$[\underline{\boldsymbol{z}}, \overline{\boldsymbol{z}}] = \varrho\left(\left[\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}\right][\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}] + \left[\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}\right]\right).$$

In the special case where $[\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]$ is non-negative—for example image inputs scaled to the intensity range $[0, 1]$ or outputs of a previous ReLU layer—this can be explicitly calculated via

$$\begin{aligned}
\underline{\boldsymbol{z}} &= \varrho\left(\max\left\{\underline{\boldsymbol{W}}, 0\right\}\underline{\boldsymbol{x}} + \min\left\{\underline{\boldsymbol{W}}, 0\right\}\overline{\boldsymbol{x}} + \underline{\boldsymbol{b}}\right), \\
\overline{\boldsymbol{z}} &= \varrho\left(\min\left\{\overline{\boldsymbol{W}}, 0\right\}\underline{\boldsymbol{x}} + \max\left\{\overline{\boldsymbol{W}}, 0\right\}\overline{\boldsymbol{x}} + \overline{\boldsymbol{b}}\right),
\end{aligned}$$

where the maximum and minimum functions are applied component-wise. Applying this for all network layers finally yields $\underline{\boldsymbol{\Phi}}$ and $\overline{\boldsymbol{\Phi}}$.

C. INN Coverage Bound

For both data sets in the main paper the proportion of ground truth values that lie inside the intervals were documented. Furthermore, it was argued that this type of coverage bound can be theoretically justified using the Markov Inequality. In the following this argument from the main paper is formally derived.

For some data distribution X, Y and a tightness parameter β the following loss is used:

$$\begin{aligned}
\mathcal{L}(\underline{\Phi}, \overline{\Phi}) &= \mathbb{E}\left[\max(\boldsymbol{y} - \overline{\Phi}(\boldsymbol{x}), 0)^2 + \max(\underline{\Phi}(\boldsymbol{x}) - \boldsymbol{y}, 0)^2 + \beta \cdot (\overline{\Phi}(\boldsymbol{x}) - \underline{\Phi}(\boldsymbol{x}))\right] \\
&= \int_{\mathcal{X}} \mathbb{E}\left[\max(\boldsymbol{y} - \overline{\Phi}(\boldsymbol{x}), 0)^2 \,\middle|\, \boldsymbol{x}\right] + \mathbb{E}\left[\max(\underline{\Phi}(\boldsymbol{x}) - \boldsymbol{y}, 0)^2 \,\middle|\, \boldsymbol{x}\right] + \beta \cdot (\overline{\Phi}(\boldsymbol{x}) - \underline{\Phi}(\boldsymbol{x}))\, \mathrm{d}\mathbb{P}_X(\boldsymbol{x}).
\end{aligned}$$

Assuming that this loss is optimized during training yields

$$
\begin{aligned}
& 0 = \int_{\mathcal{X}} \frac{\partial}{\partial \overline{\Phi}(\boldsymbol{x})} \left(\mathbb{E}\left[\max(\boldsymbol{y} - \overline{\Phi}(\boldsymbol{x}), 0)^2 \,\middle|\, \boldsymbol{x} \right] + \mathbb{E}\left[\max(\underline{\Phi}(\boldsymbol{x}) - \boldsymbol{y}, 0)^2 \,\middle|\, \boldsymbol{x} \right] + \beta \cdot (\overline{\Phi}(\boldsymbol{x}) - \underline{\Phi}(\boldsymbol{x})) \right) \mathrm{d}\mathbb{P}_X(\boldsymbol{x}) \\
\iff \quad & 0 = -\int_{\mathcal{X}} 2\mathbb{E}\big[\max(\boldsymbol{y} - \overline{\Phi}(\boldsymbol{x}), 0) \big] \, \mathrm{d}\mathbb{P}_X(\boldsymbol{x}) + \beta \\
\iff \quad & \frac{1}{2}\beta = \int_{\mathcal{X}} \mathbb{E}\big[\max(\boldsymbol{y} - \overline{\Phi}(\boldsymbol{x}), 0) \big] \, \mathrm{d}\mathbb{P}_X(\boldsymbol{x})
\end{aligned}
$$

and analogously

$$
\frac{1}{2}\beta = \int_{\mathcal{X}} \mathbb{E}\big[\max(\underline{\Phi}(\boldsymbol{x}) - \boldsymbol{y}, 0) \big] \, \mathrm{d}\mathbb{P}_X(\boldsymbol{x}).
$$

Using the Markov Inequality with $h_1(\zeta) := \max(\zeta - \overline{\Phi}(\boldsymbol{x}), 0)$ and $h_2(\zeta) := \max(\zeta + \underline{\Phi}(\boldsymbol{x}), 0)$, we obtain that for the marginalized distribution the following holds true:

$$
\mathbb{P}(\boldsymbol{y} \geq \overline{\Phi}(\boldsymbol{x}) + \lambda\beta) \leq \frac{\mathbb{E}\big[h_1(\boldsymbol{y})\big]}{h_1(\overline{\Phi}(\boldsymbol{x}) + \lambda\beta)} = \frac{\mathbb{E}\big[\max(\boldsymbol{y} - \overline{\Phi}(\boldsymbol{x}), 0) \big]}{\lambda\beta}
$$

and

$$
\mathbb{P}(\boldsymbol{y} \leq \underline{\Phi}(\boldsymbol{x}) - \lambda\beta) = \mathbb{P}(-\boldsymbol{y} \geq -\underline{\Phi}(\boldsymbol{x}) + \lambda\beta) \leq \frac{\mathbb{E}\big[h_2(-\boldsymbol{y})\big]}{h_2(-\underline{\Phi}(\boldsymbol{x}) + \lambda\beta)} = \frac{\mathbb{E}\big[\max(\underline{\Phi}(\boldsymbol{x}) - \boldsymbol{y}, 0)\big]}{\lambda\beta}.
$$

Hence, we conclude that

$$
\begin{aligned}
\mathbb{P}(\big\{\text{Label is inside interval bounds plus } \lambda\beta\big\}) &= \int_X \mathbb{P}(\underline{\Phi}(\boldsymbol{x}) - \lambda\beta \leq \boldsymbol{y} \leq \overline{\Phi}(\boldsymbol{x}) + \lambda\beta) \, d\mathbb{P}_X \\
&= 1 - \int_X \mathbb{P}(\boldsymbol{y} \leq \underline{\Phi}(\boldsymbol{x}) - \lambda\beta) + \mathbb{P}(\boldsymbol{y} \geq \overline{\Phi}(\boldsymbol{x}) + \lambda\beta) \, d\mathbb{P}_X \\
&\geq 1 - \int_X \frac{\mathbb{E}\big[\max(\boldsymbol{y} - \overline{\Phi}(\boldsymbol{x}), 0)\big]}{\lambda\beta} + \frac{\mathbb{E}\big[\max(\underline{\Phi}(\boldsymbol{x}) - \boldsymbol{y}, 0)\big]}{\lambda\beta} d\mathbb{P}_X \\
&= 1 - \frac{1}{\lambda}.
\end{aligned}
$$

We can furthermore bound the probability that for a given data point $\boldsymbol{x}$ the label $\boldsymbol{y}$ has a probability of more than α to be outside the interval bounds:

$$
\begin{aligned}
& \frac{1}{\lambda} \geq \mathbb{E}_X\left[\mathbb{P}(\boldsymbol{y} < \underline{\Phi}(\boldsymbol{x}) - \lambda\beta, \ \boldsymbol{y} > \overline{\Phi}(\boldsymbol{x}) + \lambda\beta) \right] \geq \mathbb{E}_X\left[\alpha \mathbb{1}_{\mathbb{P}(\boldsymbol{y} < \underline{\Phi}(\boldsymbol{x}) - \lambda\beta, \ \boldsymbol{y} > \overline{\Phi}(\boldsymbol{x}) + \lambda\beta) > \alpha} \right] \\
\iff \quad & \frac{1}{\lambda\alpha} \geq \mathbb{E}_X\left[\mathbb{1}_{\mathbb{P}(\boldsymbol{y} < \underline{\Phi}(\boldsymbol{x}) - \lambda\beta, \ \boldsymbol{y} > \overline{\Phi}(\boldsymbol{x}) + \lambda\beta) > \alpha} \right] = 1 - \mathbb{E}_X\left[\mathbb{1}_{\mathbb{P}(\underline{\Phi}(\boldsymbol{x}) - \lambda\beta \leq \boldsymbol{y} \leq \overline{\Phi}(\boldsymbol{x}) + \lambda\beta) \geq 1 - \alpha} \right] \\
\iff \quad & 1 - \frac{1}{\lambda\alpha} \leq \mathbb{E}_X\left[\mathbb{1}_{\mathbb{P}(\underline{\Phi}(\boldsymbol{x}) - \lambda\beta \leq \boldsymbol{y} \leq \overline{\Phi}(\boldsymbol{x}) + \lambda\beta) \geq 1 - \alpha} \right].
\end{aligned}
$$

In words, for $\lambda > 0$ and $\alpha > 0$ the probability mass of all samples $\boldsymbol{x}$, for which the corresponding label $\boldsymbol{y}$ has the probability of at least $1 - \alpha$ to be inside the interval, is at least $1 - \frac{1}{\lambda\alpha}$.

D. INNs and the Bayesian View

As described in Section 2 on related work, popular UQ approaches for neural networks have their roots in a Bayesian treatment of the learning problem. In a nutshell, this involves modelling the unknown data distribution (X, Y) on $\mathcal{X} \times \mathcal{Y}$ via a neural network $\Phi_W : \mathcal{X} \to \mathcal{Y}$, where W is now also a random variable and represents the collection of all network parameters. More precisely, one assumes that $p(Y|X, W)$ follows a simple distribution depending on X and W through

$\Phi_W(X)$. A typical choice is a Gaussian distribution $Y|X, W \sim \mathcal{N}(\Phi_W(X), \tau^{-2}I)$ with mean $\Phi_W(X)$ and some fixed precision τ. The network training requires the estimation of

$$p(W|\boldsymbol{X}, \boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\boldsymbol{X}, W)p(W)}{\int p(\boldsymbol{Y}|\boldsymbol{X}, W)p(W)\,\mathrm{d}W}, \tag{3}$$

from given training data $(\boldsymbol{X}, \boldsymbol{Y}) = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^m$ and some prior assumption $p(W)$ on the network parameters. Inference for a new input $\boldsymbol{x}$ requires the estimation of the posterior

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{Y}) = \int p(\boldsymbol{y}|\boldsymbol{x}, W)p(W|\boldsymbol{X}, \boldsymbol{Y})\,\mathrm{d}W. \tag{4}$$

The evidence, that is the denominator in (3), is typically intractable. Variational Bayesian methods try to approximate $p(W|\boldsymbol{X}, \boldsymbol{Y})$ by another distribution $q_\theta(W)$ from a family of distributions q_θ parametrized by θ. Minimizing the KL-divergence between $p(W|\boldsymbol{X}, \boldsymbol{Y})$ and $q_\theta(W)$ is equivalent to maximizing the evidence lower bound (ELBO)

$$\int q_\theta(W) \log p(\boldsymbol{Y}|\boldsymbol{X}, W)\,\mathrm{d}W - \mathrm{KL}(q_\theta(W)||p(W)). \tag{5}$$

Training the network in this variational setting entails finding the optimal parameter choice θ^* maximizing (5), and inference can be approximated by

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{Y}) \approx \int p(\boldsymbol{y}|\boldsymbol{x}, W)q_{\theta^*}(W)\,\mathrm{d}W. \tag{6}$$

In light of this tradition, we want to briefly demonstrate how our Interval Neural Networks can also be viewed within the framework of variational Bayesian networks.

For an L-layer neural network with weight matrices $\boldsymbol{W}^{(l)}$ and bias vectors $\boldsymbol{b}^{(l)}$, our interval network approach introduces upper and lower bound parameters $\theta = \{\underline{\boldsymbol{W}}^{(l)}, \overline{\boldsymbol{W}}^{(l)}, \underline{\boldsymbol{b}}^{(l)}, \overline{\boldsymbol{b}}^{(l)}\}_{l=1}^L$. But instead of precisely parametrizing the approximating distribution by θ, we allow q_θ to be any distribution of weights and biases supported within the specified intervals. We now want to analyze the ELBO in (5) and the approximate posterior in (6) in this situation.

Recall that, given the interval bounds θ, the range of possible values of $\Phi_W(\boldsymbol{x})$ for a fixed input $\boldsymbol{x}$ and W distributed according to $q_\theta(W)$ is denoted as $[\underline{\Phi}(\boldsymbol{x}), \overline{\Phi}(\boldsymbol{x})]$. Further, for any target $\boldsymbol{y}$ we denote the choice of weights achieving the best and worst approximation within this range as

$$\underline{\boldsymbol{W}}(\boldsymbol{x}, \boldsymbol{y}) = \operatorname*{argmin}_{W\sim q_\theta} \|\Phi_W(\boldsymbol{x}) - \boldsymbol{y}\|_2^2 \qquad \text{and} \qquad \overline{\boldsymbol{W}}(\boldsymbol{x}, \boldsymbol{y}) = \operatorname*{argmax}_{W\sim q_\theta} \|\Phi_W(\boldsymbol{x}) - \boldsymbol{y}\|_2^2.$$

This allows us to estimate the first term in the ELBO as

$$\int q_\theta(W) \log p(\boldsymbol{Y}|\boldsymbol{X}, W)\,\mathrm{d}W \le -m\log(C) - \sum_{i=1}^m \frac{\tau^{2d}}{2}\|\Phi_{\underline{\boldsymbol{W}}(\boldsymbol{x}_i,\boldsymbol{y}_i)}(\boldsymbol{x}_i) - \boldsymbol{y}_i\|_2^2,$$

where $C = (2\pi\tau^{-2})^{d/2}$ is the normalizing constant of the Gaussian density with precision τ. Similarly

$$\begin{aligned}
\int q_\theta(W) \log p(\boldsymbol{Y}|\boldsymbol{X}, W)\,\mathrm{d}W &\ge -m\log(C) - \sum_{i=1}^m \frac{\tau^{2d}}{2}\|\Phi_{\overline{\boldsymbol{W}}(\boldsymbol{x}_i,\boldsymbol{y}_i)}(\boldsymbol{x}_i) - \boldsymbol{y}_i\|_2^2 \\
&\ge -m\log(C) - \sum_{i=1}^m \tau^{2d}\big(\|\Phi_{\underline{\boldsymbol{W}}(\boldsymbol{x}_i,\boldsymbol{y}_i)}(\boldsymbol{x}_i) - \boldsymbol{y}_i\|_2^2 \\
&\qquad\qquad + \|\Phi_{\overline{\boldsymbol{W}}(\boldsymbol{x}_i,\boldsymbol{y}_i)}(\boldsymbol{x}_i) - \Phi_{\underline{\boldsymbol{W}}(\boldsymbol{x}_i,\boldsymbol{y}_i)}(\boldsymbol{x}_i)\|_2^2\big) \\
&\ge -m\log(C) - \sum_{i=1}^m \tau^{2d}\big(\|\Phi_{\underline{\boldsymbol{W}}(\boldsymbol{x}_i,\boldsymbol{y}_i)}(\boldsymbol{x}_i) - \boldsymbol{y}_i\|_2^2 + \|\overline{\Phi}(\boldsymbol{x}_i) - \underline{\Phi}(\boldsymbol{x}_i)\|_2^2\big).
\end{aligned}$$

We observe that minimizing the INN loss $\mathcal{L}(\underline{\Phi}, \overline{\Phi})$ with $\beta = 1$ corresponds to maximizing a lower bound for one part of the ELBO. The other part of the ELBO, the KL-divergence to the prior, corresponds to weight regularisation during the

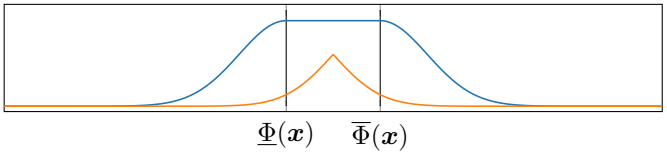

Figure 3. A schematic visualization of the lower and upper bounds for the predictive posterior of INN inference derived from variational Bayesian principles. The INN prediction interval is marked by vertical lines.

network training, e.g. weight decay. Further, the gap between the upper and lower bound on the ELBO is determined by $\sum_i \tau^{2d} \|\overline{\Phi}(\boldsymbol{x}_i) - \underline{\Phi}(\boldsymbol{x}_i)\|_2^2$. Therefore, the size of the output intervals also corresponds to how far from the true ELBO we are, when considering the training loss $\mathcal{L}$ instead.

During inference, the approximate posterior in (6) can then be estimated from the bounds

$$\frac{1}{C} e^{-\frac{\tau^{2d}}{2}\|\Phi_{\overline{\boldsymbol{W}}(\boldsymbol{x},\boldsymbol{y})}(\boldsymbol{x})-\boldsymbol{y}\|_2^2} \leq \int p(\boldsymbol{y}|\boldsymbol{x}, W) q_{\theta^*}(W)\, \mathrm{d}W \leq \frac{1}{C} e^{-\frac{\tau^{2d}}{2}\|\Phi_{\underline{\boldsymbol{W}}(\boldsymbol{x},\boldsymbol{y})}(\boldsymbol{x})-\boldsymbol{y}\|_2^2}.$$

A schematic visualization of these bounds can be seen in Figure 3. Even though the true posterior can lie anywhere between the bounds, we observe a fast decay of the probability of the target $\boldsymbol{y}$ lying far outside the predicted interval $[\underline{\Phi}(\boldsymbol{x}), \overline{\Phi}(\boldsymbol{x})]$. This is line with the findings derived via the Markov bound in Appendix C.

E. Experiments Details

E.1. Baseline Methods

In addition to our Interval Neural Network approach we consider two other fast and lightweight UQ baselines mentioned. First, there is Monte Carlo dropout (MCDROP) proposed by (Gal & Ghahramani, 2016; Kendall & Gal, 2017). Here, uncertainty scores are obtained through the sample variance of multiple stochastic forward passes on the same input data point. In other words, if $\boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_T$ are realizations of independent draws of random dropout masks for the same prediction network $\boldsymbol{\Phi}$, then the pixel-wise uncertainty estimate is given by

$$\boldsymbol{u}_{\text{MCDROP}}(\widetilde{\boldsymbol{x}}) = \tfrac{1}{T-1}\left(\sum_{t=1}^T \boldsymbol{\Phi}_t(\widetilde{\boldsymbol{x}})^2 - \tfrac{1}{T}\left(\sum_{t=1}^T \boldsymbol{\Phi}_t(\widetilde{\boldsymbol{x}})\right)^2\right).$$

Second, there is direct variance estimation (PROBOUT) proposed by (Nix & Weigend, 1994) and later expanded by (Gast & Roth, 2018). It comprises a simple recipe for uncertainty scores: the number of output components of the prediction network is doubled and trained to approximate the mean and variance of a Gaussian distribution.

$$\boldsymbol{\Phi}_{\text{PROBOUT}} \colon \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n, \quad \widetilde{\boldsymbol{x}} \mapsto (\boldsymbol{\Phi}_{\text{mean}}(\widetilde{\boldsymbol{x}}), \boldsymbol{\Phi}_{\text{var}}(\widetilde{\boldsymbol{x}}))$$

which are trained by minimizing the empirical loss

$$\sum_i \left\| \frac{\boldsymbol{y}_i - \boldsymbol{\Phi}_{\text{mean}}(\widetilde{\boldsymbol{x}}_i)}{\sqrt{\boldsymbol{\Phi}_{\text{var}}(\widetilde{\boldsymbol{x}}_i)}} \right\|_2^2 + \|\log \boldsymbol{\Phi}_{\text{var}}(\widetilde{\boldsymbol{x}}_i)\|_1.$$

The pixel-wise uncertainty score of PROBOUT is then simply given by the variance estimate, i.e., $\boldsymbol{u}_{\text{PROBOUT}}(\widetilde{\boldsymbol{x}}) = \boldsymbol{\Phi}_{\text{var}}(\widetilde{\boldsymbol{x}})$.

E.2. Tasks, Data and Neural Network Architectures

E.2.1. 1D DECONVOLUTION

We choose $\boldsymbol{A} = \boldsymbol{D}^\top \boldsymbol{S} \boldsymbol{D} \in \mathbb{R}^{512\times512}$, where $\boldsymbol{D} \in \mathbb{R}^{512\times512}$ is a discrete cosine transform and $\boldsymbol{S} \in \mathbb{R}^{512\times512}$ is a diagonal matrix with exponentially decaying values. We consider discretizations of piecewise constant functions with random jump positions and heights as the signal distribution in $\mathbb{R}^{512}$. The blurred measurements $\boldsymbol{x} \in \mathbb{R}^{512}$ corresponding to each signal sample $\boldsymbol{y}$ are simulated by computing $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{\eta}$. The considered data set consists of 2000 sample pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, 1600 of which were used for training, 200 for validation and 200 for testing. This one-dimensional data allows for good

illustration of the different INN characteristics. The prediction DNN for the 1DDECONV task, called DeconvNet, consists of a convolutional neural network (CNN) trained to directly map $\boldsymbol{x}$ to $\boldsymbol{y}$. It consists of 10 convolutional layers and three dropout layers, one with dropout probability 0.2 and the other two with probability 0.5. The number of channels increases through the first 7 layers to 256 and decreases back to 1 in the successive layers. No pooling is employed and the data size of each channel is held to be the same as the input size throughout the network. The prediction DNN was trained for 100 epochs using Adam (Kingma & Ba, 2014) with a learning rate of 10^{-3} and batch size 256. The interval parameters of the INN were then trained for another 100 epochs and with a learning rate of 10^{-5} and $\beta = 2 \times 10^{-3}$. For the MCDROP comparison, 64 samples were used to estimate mean and variance and for the PROBOUT comparison, the PROBOUT loss was also optimized for 100 epochs using again the Adam optimizer with a learning rate of 10^{-4}.

E.2.2. LIMITED ANGLE CT

For this task, we consider a simulation of the noiseless Radon transform with a moderate missing wedge of $30°$ for the forward model. The non-learned inversion $\boldsymbol{A}^\dagger$ is based on the filtered backprojection algorithm (FBP) (Natterer, 2001). The underlying prediction network is a U-Net (Ronneberger et al., 2015) variant. Our experiments are based on a data set consisting of 512×512 human CT scans from the AAPM Low Dose CT Grand Challenge data (McCollough, 2016).[2] In total, it contains 2580 images of 10 patients. Eight of these ten patients were used for training (2036 samples), one for validation (214 samples) and one for testing (330 samples). We use a bit-depth of at least 16 in all steps so that no details of the 12bit DICOM data are lost. Limited angle measurements are simulated. Input images are full dose reconstructions with a slice thickness of 3mm.

E.2.3. IMAGE DENOISING

This task consists of removing additive Gaussian noise with standard deviation $25/255$ from greyscale images (rescaled to the intensity range $[0, 1]$) from the Berkeley Segmentation Dataset (Martin et al., 2001). The prediction network underlying all uncertainty methods is a fully-convolutional residual network with 17 convolution layers, inspired by (Zhang et al., 2017).

E.3. Data Preparation

E.3.1. ADVERSARIAL ARTIFACT DETECTION

For the DENOISE data we use $\lambda = 0.5$, and the adversarial targets are created by adding noise to a random 50×50 patch in the reconstruction $\boldsymbol{y}_{\text{rec}} = \boldsymbol{\Phi}(\widetilde{\boldsymbol{x}})$. Thus, the denoising network is forced to fail its task in that region; see Figure 9. For the CT task we found that the second term in (2) is not required, i.e., we use $\lambda = 0$. Adversarial targets are created by subtracting 1.5 times its mean value from $\boldsymbol{x}_{\text{rec}}$ within a random 50×50 square, leading to clearly visible artifacts in the corresponding reconstructions; see Figure 1.

E.3.2. ATYPICAL ARTIFACT DETECTION

For the DENOISE task this is achieved by locally changing the noise distribution, i.e., we replace the Gaussian noise by Salt & Pepper noise in one half of each image in the test set; see Figure 9. For the CT task the silhouette of a peace dove is inserted in each image of the test set; see Figure 1. The simulation of the measurements and model-based inversions is carried out on the new test set as before.

[2]See: https://www.aapm.org/GrandChallenge/LowDoseCT/; We would like to thank Dr. Cynthia McCollough, the Mayo Clinic, and the American Association of Physicists in Medicine as well as the grants EB017095 and EB017185 from the National Institute of Biomedical Imaging and Bioengineering for providing the AAPM data.

Table 2. Summary of the technical details regarding the neural network architecures, training, and data sets for the two use cases of DENOISE and CT. DENOISE data is available at `https://github.com/husqin/DnCNN-keras` (not affiliated with authors of this paper).

	Image Denoising	**Limited Angle CT**
Base Network	based on (Zhang et al., 2017) dropout (0.05) after every other conv. trained with Adam(Kingma & Ba, 2014), 50 epochs learning rate: 10^{-4} mini-batch size: 128 no batch normalization as in (Zhang et al., 2017) 128 instead of 64 conv. channels, cf. (Zhang et al., 2017)	U-Net of (Ronneberger et al., 2015) dropout (0.7) after down-/up-sampling trained with Adam, 400 epochs learning rate: $7.5 \cdot 10^{-5}$ mini-batch size: 12
INN	10 epochs with Adam learning rate: 10^{-6} $\beta = 10^{-3}$ mini batch size: 96 interval arithmetic in last 8 layers	15 epochs with Adam learning rate: 10^{-6} $\beta = 10^{-4}$ mini batch size: 6 interval arithmetic in last 12 layers
MCDROP	$T = 128$ forward passes	$T = 16$ forward passes
PROBOUT	additional output channel otherwise same setup as base network	additional output channel 400 more epochs with Adam learning rate: 10^{-7} mini-batch size: 12
Data	Berkeley Segmentation Dataset (Martin et al., 2001) 400 128×128-images; see (Schmidt & Roth, 2014; Zhang et al., 2017) overlapping 40×40-patches, stride 10 rescaled to intesity range $[0, 1]$ Gaussian noise, standard dev. $25/255$ testing: 68 images of varying size; cf. (Zhang et al., 2017)	AAPM Low Dose CT Grand Challenge 10 patients: 2580 512×512-images (8/1/1 for training/validation/testing) noiseless Radon transform 30° missing wedge Ramp-filter for FBP

F. Results: Additional Figures

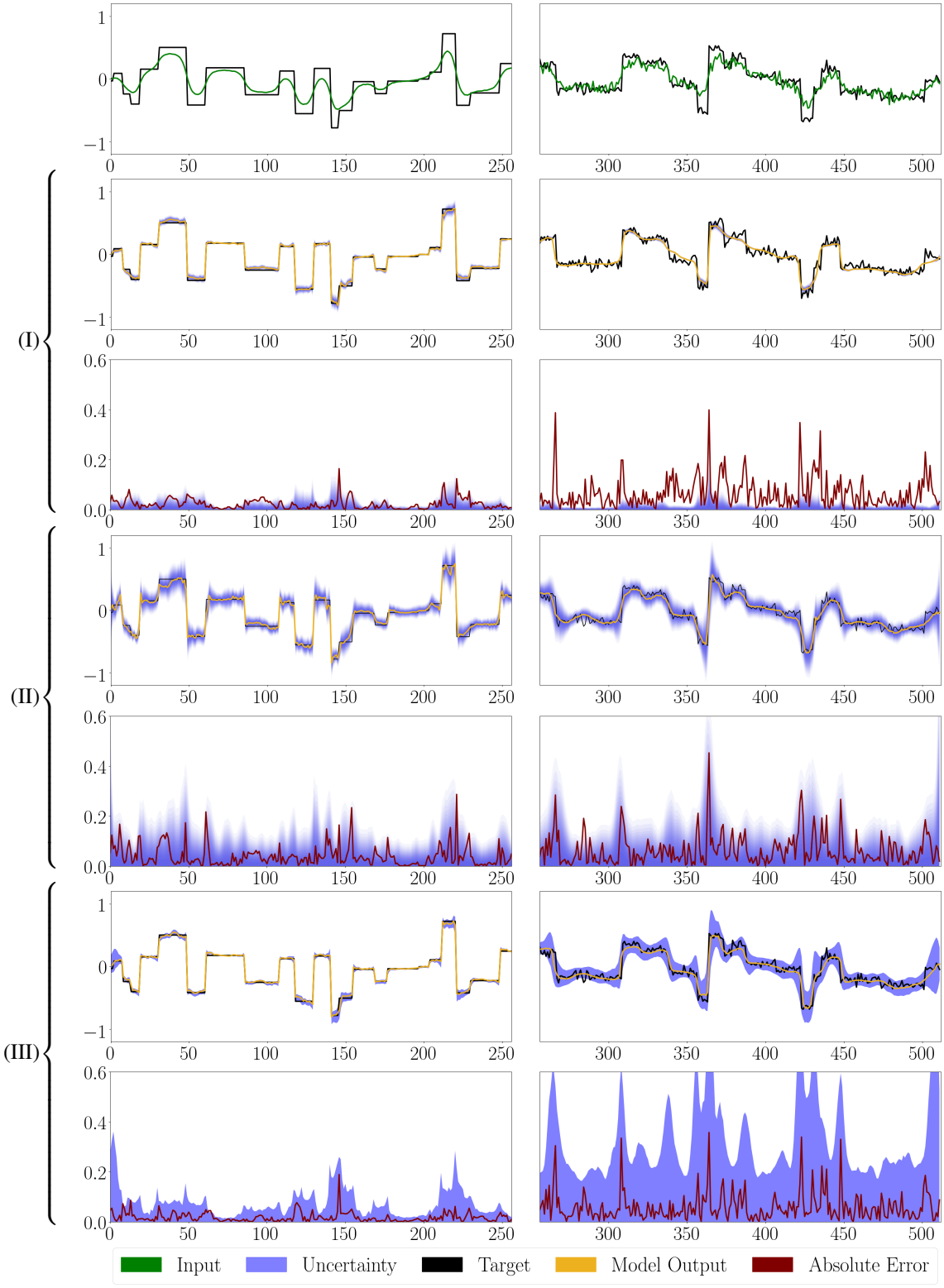

Figure 4. Results for the 1DDECONV task on the same sample without noise on the left and with Gaussian noise ($\sigma = 0.05$) on the right. The first row shows the input and target vector. The figures below show the target, the network output, together with the uncertainty estimation for the upper graph and the uncertainty estimation plotted against the absolute error in the lower graph for each corresponding method; (I) MCDROP, (II) PROBOUT, (III) INN.

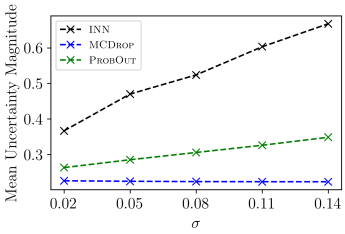

Figure 5.

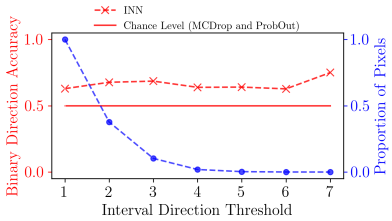

Figure 6.

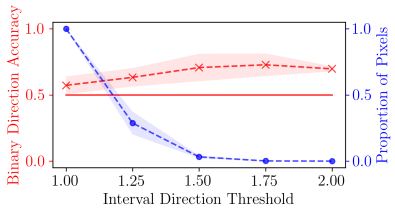

Figure 7.

Figure 8. Illustration of INN properties. Figure 5 displays the relationship of mean uncertainty magnitude measured and additive Gaussian noise on the data from the 1DDECONV task. The standard deviation of the additive Gaussian noise for the input and target data is displayed on the x-axis. The mean uncertainty magnitude, which is measured in interval size for INN (in black) and standard deviation for MCDROP (in blue) and PROBOUT (in green) averaged over the test data, is displayed on the y-axis. Figure 6 and Figure 7 display the directional information contained in the output intervals for the 1DDECONV and the CT task, respectively. Interval direction thresholds are displayed on the x-axis. These are computed by dividing the larger interval half by the smaller half. Interval halves are computed relative to the point prediction. The left y-axis (in red) displays the direction accuracy which is the mean agreement between the interval directions and the actual position of the target relative to the prediction. Finally, the right y-axis (in blue) displays the proportion of pixels that are considered at a given threshold and accuracy evaluation. CT results are means and standard deviations across the three experimental runs.

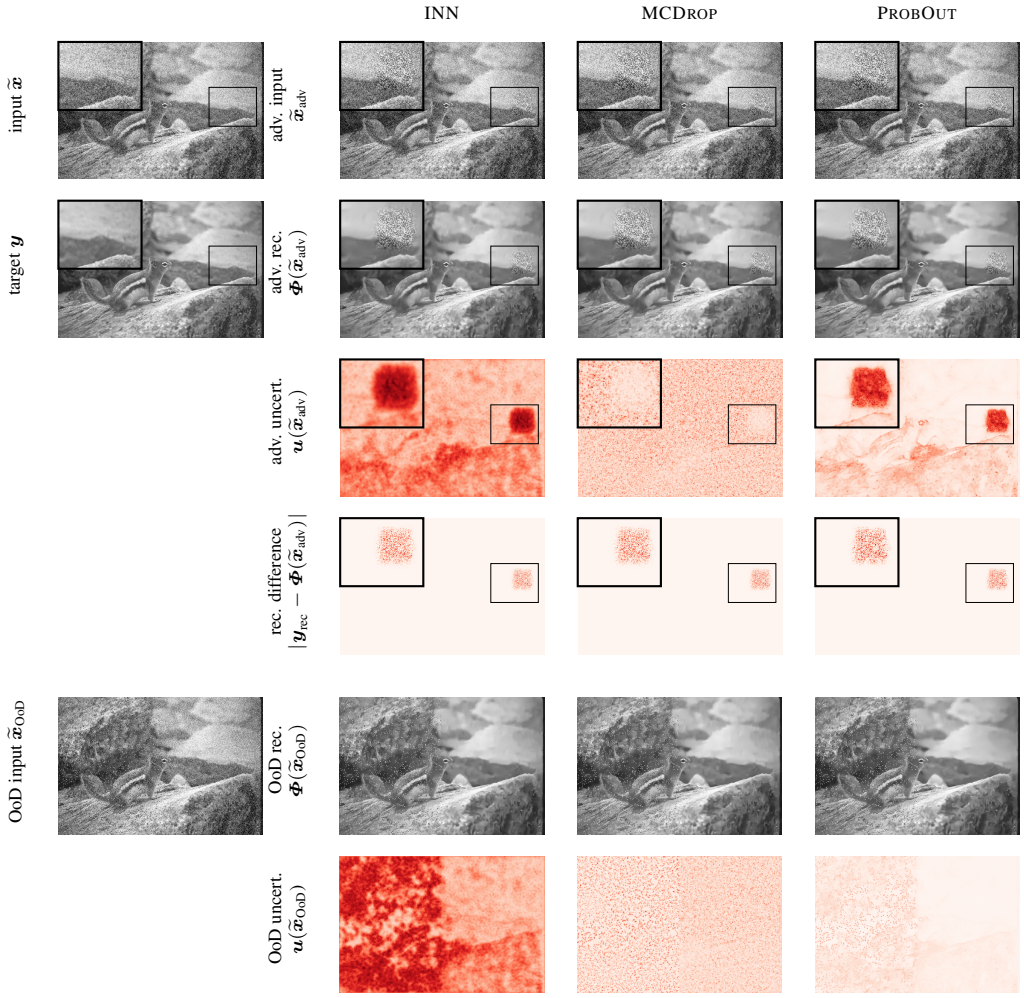

Figure 9. Results of three UQ methods for the ADVDETECT and ARTDETECT experiments for one exemplary data sample of the task.

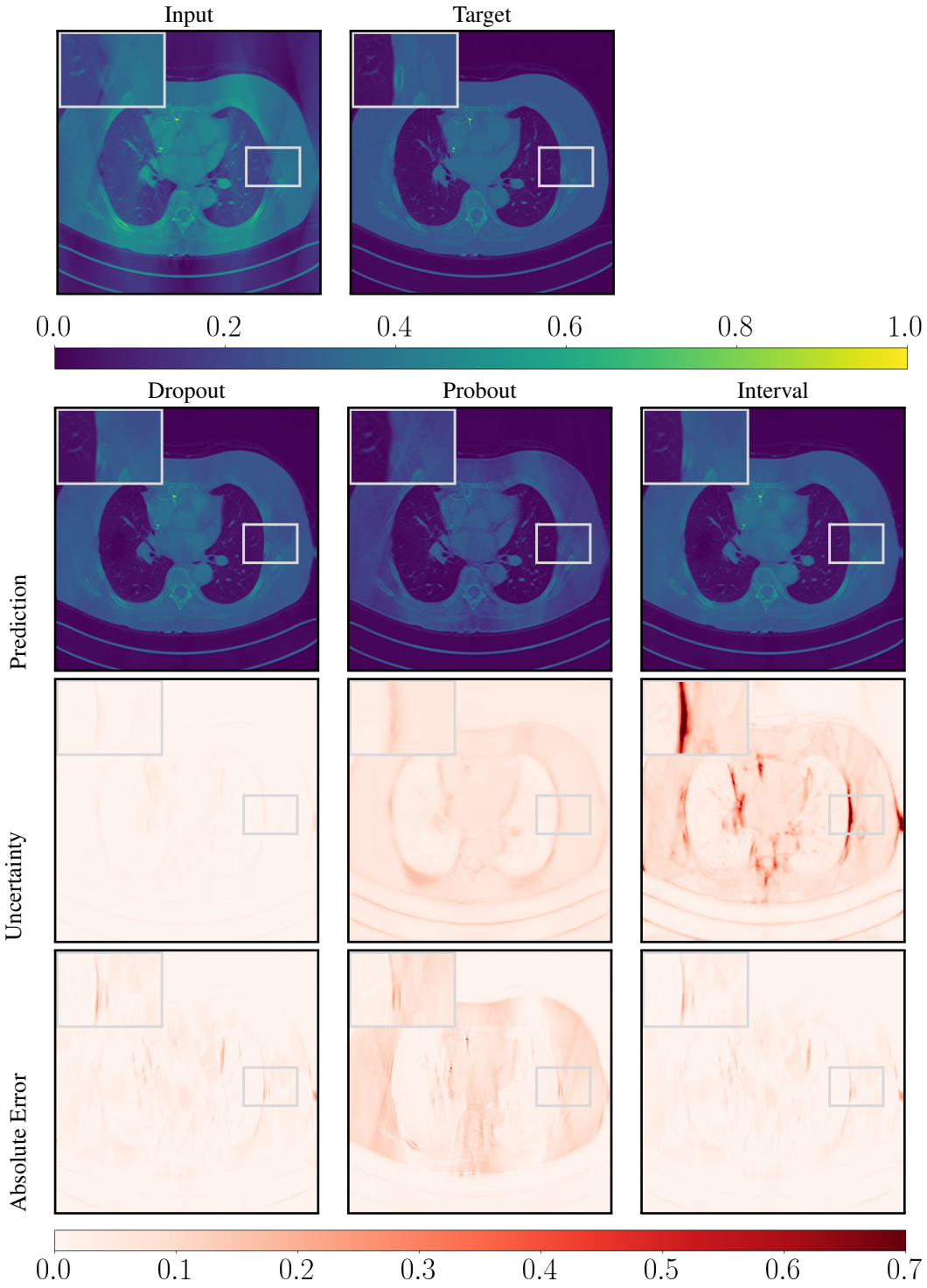

Figure 10. An example data point from the EC experiment. The first image on the top left displays the input and the second image on the top right displays the corresponding target. The images below display the corresponding predictions (top row), uncertainty scores as the standard deviation for MCDROP and PROBOUT and the interval size for INN (middle row) and absolute errors (bottom row) for each of the three uncertainty methods.