Supplementary Materials: Bayesian Inference with Certifiable Adversarial Robustness

In these supplementary materials we provide the details to aid in the reproducibility of our results and report on further experiments to deepen our understanding of the presented method. For the code to reproduce both the experiments found in the main text and in these extended materials see the github code repository at: *https://github.com/matthewwicker/CertifiableBayesianInference*, and if anything proves to be unclear or broken please email Matthew Wicker at: *matthew.wicker@cs.ox.ac.uk*.

7 APPROXIMATE INFERENCE PARAMETERS

In this section of the Supplementary Material, we list the training parameters that we used for the training of each of the networks discussed in the main text.

7.1 MNIST and FashionMNIST Parameters

	SWAG	NoisyAdam	VOGN	BBB	HMC
Learning Rate	0.1	0.001	0.35	0.45	0.075
Prior Scaling	N/A	10	10	20	500
Batch Size	128	128	128	128	60k
Epochs/Samples	20/250	20/(N/A)	20/(N/A)	20/(N/A)	(N/A)/25
PGD Iterations	10	10	10	10	10

Each network trained on MNIST is a single hidden layer fully-connected architecture with 512 neurons in the hidden layer. The parameters used for the 5 training methods are listed in the table above. Prior scaling refers to a multiplicative constant w.r.t. the initialisation parameters described in Sutskever et al. (2013). In fact, we often find the initial variance described in the later to be too small for retrieving good uncertainty estimates, and, thus, we further multiply it by the values reported in the table. Further parameters that are specific to HMC, and not included in the table, are: 3 iterations of burn-in, with 20 steps of the leapfrog numerical integrator followed by the reported 25 samples from the posterior each which explore the chain for 25 steps with the leapfrog integrator. We again note that when we perform approximate inference with HMC and the robust likelihood that we choose the initial network parameters to be the result of 10 epochs of stochastic gradient descent rather than the full-data gradient descent used during normal burn-in. Finally, we note that we follow the empirically optimal procedure stated by Gowal et al. (2018). In particular, we train with an η linearly increasing to its target value at every epoch. Again as in Gowal et al. (2018), we set the target η value 10% larger than the 'desired' robustness value.

7.2 CIFAR10 Parameters

	SWAG	NoisyAdam	VOGN
Learning Rate	0.015	0.00025	0.25
LR Decay	0.0	0.025	0.025
Prior Scaling	N/A	5	5
Batch Size	128	128	128
Epochs/Samples	45/500	45/(N/A)	45/(N/A)
PGD Iterations	10	10	10

For CIFAR10, prior to inference we perform data augmentation which involves horizontal flipping as well as random translations by up to 4 pixels. We randomly select an image from the train set with uniform probability and then select a transformation (translation or horizontal flipping) until we have augmented the data size from 60k to 100k images Finally, the network architecture is made of two convolutional layers, respectively with 16 and 32 four by four filters, followed by a 2 by 2 max pooling layer, and a fully connected layer with 100 hidden neurons.

8 CERTIFIED ROBUST RADIUS RESULTS

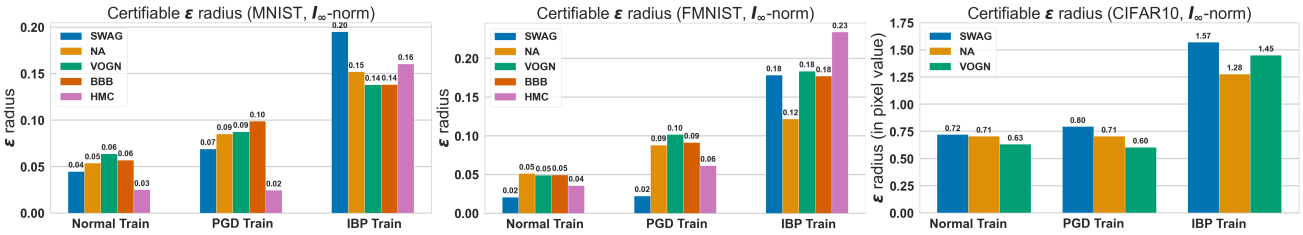

Figure 4: We plot the average certified radius for images from MNIST (left), FashionMNIST (center), and CIFAR10 (right) using the methods of Boopathy et al. (2019). We re-report the MNIST and CIFAR10 results here for ease of comparison. We observe that robust training with IBP roughly doubles the maximum verifiable radius of compared with standard training and that obtained by training on PGD adversarial examples.

Consistent with the analysis in the main text, we consider analyzing the robustness of the trained posteriors at varying values of ϵ (reported in Figure 4). In particular, we estimate the maximal ϵ radius for which each image is robust. To estimate this value, we follow the methodology of Boopathy et al. (2019): a binary search over the values of ϵ. We stress that during this procedure, we use linear propagation methods: CROWN for MNIST and FMNIST networks and CNN-Cert for CIFAR10 Networks. This is to reduce the bias of the evaluation of IBP trained networks. That is, IBP trained networks intuitively should evaluate well against IBP but it is important to see if tighter methods still show large improvements. As reported in the paper, we find that training with PGD does not tend to increase the certifiable radius in a significant way, while training with IBP allows one to double the certifiable radius.

9 ADVERSARIAL TRAINING PARAMETER STUDY

In this section we analyse the choice of p_ϵ, that is, the distribution that controls the adversarial perturbation strength at training time. Recall that the distribution used in the main text follows related work on training of deterministic neural networks:

$$p_\epsilon(\epsilon) = \begin{cases} \lambda & \text{if } \epsilon = 0 \\ 1 - \lambda & \text{if } \epsilon = \eta \end{cases}. \tag{8}$$

In particular, we first study the affect of changing the λ parameter in Eqn (8) which parameterizes the relative penalty between accuracy and robustness during inference. Next, we study the effect of changing η in Eqn (8) which sets a the maximum allowable manipulation magnitude during inference. Finally, we study the effect of changing the form of the ϵ probability density function to two different continuous, non-negative distribution.

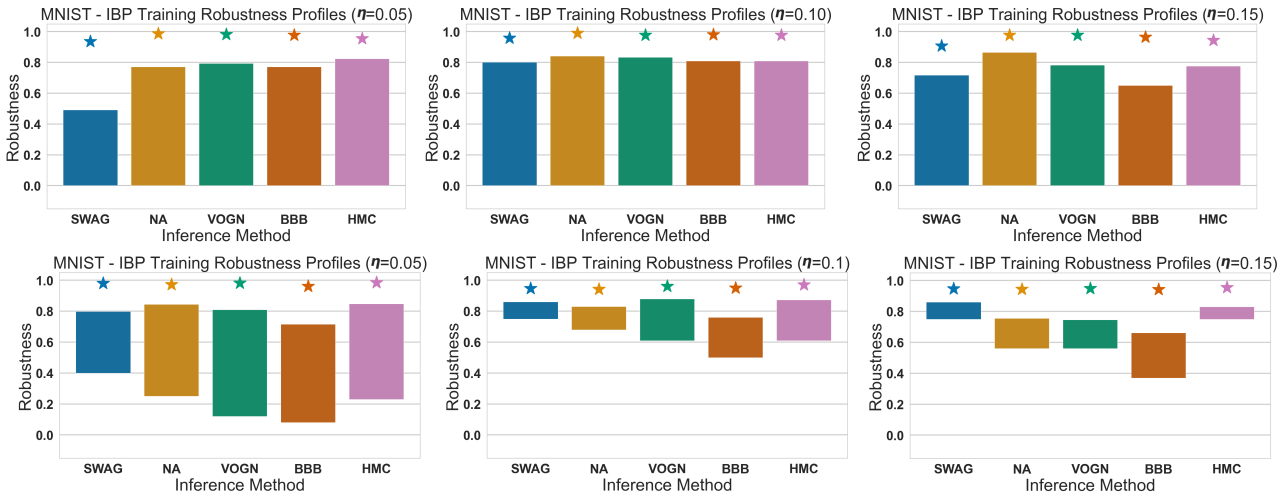

Figure 5: **Left to Right:** Effect of varying (increasing) values of η on the robustness profile of resulting approximate posteriors. **Top Row:** Robustness profiles of networks using the robust likelihood with PGD as an approximate worst-case adversary. **Bottom Row:** Robustness profiles of networks using the robust likelihood with IBP as an approximate worst-case adversary. Accuracy (plotted as star points), an empirical estimation of $\mathcal{R}_\epsilon$ obtained using PGD (upper bound of each bar), and $\mathcal{R}_\epsilon^{IBP}$ (lower bound of each bar), obtained for $\epsilon = 0.1$ on the MNIST dataset.

In each figure, we maintain the plotting conventions used in Figure 1 of the main text. For each posterior: accuracy is plotted as a star point, an empirical estimation of $\mathcal{R}_\epsilon$ obtained using PGD (upper bound of each bar), and $\mathcal{R}_\epsilon^{IBP}$ (lower bound of each bar), obtained for $\epsilon = 0.1$ on the MNIST dataset. For the following analysis we only report the lower-bound based on IBP.

9.1 The Effect of Adversarial Magnitude During Inference

When approximating the robust likelihood with PGD during inference, we find that the shift in magnitude of η on the resulting robustness estimates is largely dependant on the method of approximate inference. Interestingly, we find that for SWAG and BBB, that training with $\eta = 0.15$ becomes problematic as it seems that with the current training parameters (reported in the previous section), the 1 layer, 512 neuron network may not have had enough capacity to accurately capture good adversarial robustness. The connection between robustness of (gradient-based) adversarial trained deterministic networks and capacity is discussed at length in Madry et al. (2017). We find that NA and HMC are relatively unaffected by small changes to the η magnitude and enjoy similar heightened robustness for each observed value.

The effect of η is much more pronounced when we perform inference with the IBP robust likelihood. We see that having an η smaller than ϵ (in $\mathcal{R}_\epsilon$) results in worse lower-bound potentially indicating a less robust posterior. For parameter and natural gradient VI, we also find that having an η that is much larger than ϵ can be detrimental as too strong of an adversary can be problematic for learning.

9.2 The Effect of Trading Accuracy and Robustness

In Eqn (8) the parameter λ effectively controls the relative weighting of accuracy-error and robust-error during the inference procedure. Specifically, we note the cases $\lambda = 1.0$ which results in the standard likelihood (a.k.a. the categorical cross-entropy in the case of classification), and $\lambda = 0.0$ results in a framework in which give importance solely to robustness. In Figure 6 we report the change in robustness profiles for $\lambda \in \{0.75, 0.5, 0.25\}$ for training with the worst-case approximated by PGD (top row) and IBP (bottom row).

When approximating the robust likelihood with PGD, we find that HMC and natural gradient methods (VOGN, NA) are not strongly affected by the choice of λ, whereas we see the most pronounced difference with SWAG which is greatly affected by choice of λ. In particular we highlight roughly a 20% raw increase in the robustness to gradient based attacks for each 0.25 decrease in λ. On the other hand, when training with IBP there is large shift in the resulting robustness profiles for parameter and natural gradient VI methods (BBB, VOGN, NA). Notably, we see a large (50% raw) increase in

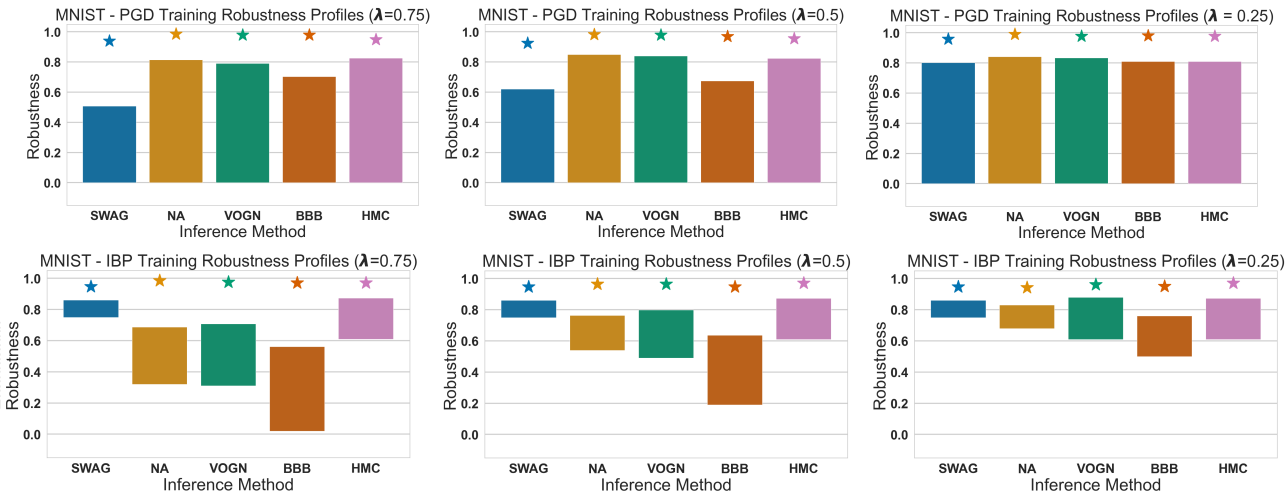

Figure 6: **Left to Right:** Effect of varying (decreasing) values of λ on the robustness profile of resulting approximate posteriors. **Top Row:** Robustness profiles of networks using the robust likelihood with PGD as an approximate worst-case adversary. **Bottom Row:** Robustness profiles of networks using the robust likelihood with IBP as an approximate worst-case adversary. Accuracy (plotted as star points), an empirical estimation of $\mathcal{R}_\epsilon$ obtained using PGD (upper bound of each bar), and $\mathcal{R}_\epsilon^{IBP}$ (lower bound of each bar), obtained for $\epsilon = 0.1$ on the MNIST dataset.

the lower-bound for BBB as the value for λ varies between 0.75 and 0.25.

9.3 On the Choice of Density Function for Adversarial Magnitude

In Figure 7 and Figure 8, we study changing the form of p_ϵ from the density given in Eqn (8) to a Rayleigh distribution and an Exponential distribution, respectively. We have chosen these distributions in particular because they have non-negative support and a single controlling variable. In principle, however, any distribution (with a positive support) can be chosen for the form of p_ϵ. As noted in the main text, during the computation of the loss function, one must marginalize over the selected p_ϵ distribution, which in this case is done via Monte Carlo with only 10 samples from p_ϵ per batch. Consistent with the study presented in the main text, we evaluate robustness profiles with ϵ set to 0.1.

9.3.1 Using a Rayleigh Distribution

In Figure 7, we plot the case in which training is done by using an Rayleigh distribution with the scale set to η for p_ϵ as follows:

$$p_\epsilon(\epsilon) = \frac{\epsilon}{\eta^2}\exp\left(\frac{-\epsilon^2}{2\eta^2}\right) \tag{9}$$

In our experiments, we find that using a Rayleigh distribution for p_ϵ does marginally improve the robustness ($\mathcal{R}_\epsilon$) when training against a PGD adversary ($\approx 4\%$ on average). We also find that when using the altered pdf, the main result stated in the paper, that training with robust likelihood is the only method that gives non-trivial lower bounds on robustness, still holds. However, we find that the use of the Rayleigh distribution has an adverse affect on the overall robustness profile compared to training with Eqn (8).

9.3.2 Using an Exponential Distribution

In Figure 8, we give the results when p_ϵ is selected as an exponential distribution with the rate set to η^{-1}:

$$p_\epsilon(\epsilon) = \frac{1}{\eta}\exp\left(\frac{-\epsilon}{\eta}\right) \tag{10}$$

When training against a PGD adversary, we found that an using an exponential distribution for p_ϵ also leads to small increases in robustness against adversarial attacks, with an average increase of $\approx 5\%$. Consistent with the results for the

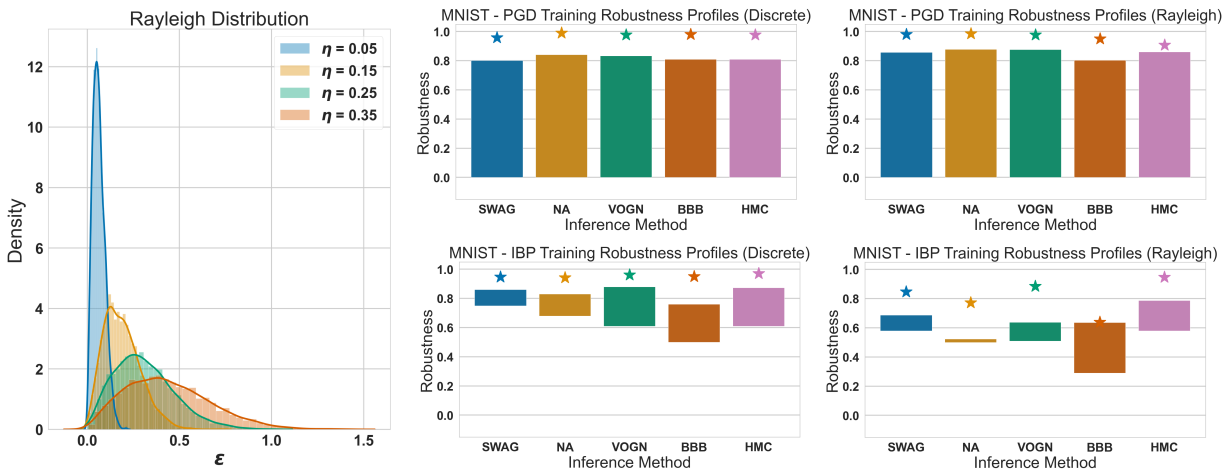

Figure 7: **Left:** Effect of varying the scale η of the Rayleigh distribution on the density p_ϵ when training we use $\eta = 0.1$. **Right, Top Row:** Robustness profiles of networks using the robust likelihood with PGD as an approximate worst-case adversary. **Right, Bottom Row:** Robustness profiles of networks using the robust likelihood with IBP as an approximate worst-case adversary. Accuracy (plotted as star points), an empirical estimation of $\mathcal{R}_\epsilon$ obtained using PGD (upper bound of each bar), and $\mathcal{R}_\epsilon^{IBP}$ (lower bound of each bar), obtained for $\epsilon = 0.1$ on the MNIST dataset.

Rayleigh distribution, the main result stated in the paper, that training with the robust likelihood is the only method that gives non-trivial lower bounds on robustness. However, we continue to find that the use of the exponential distribution when training with IBP, consistent with the Rayleigh distribution, has an adverse affect on the overall robustness profile compared to training with Eqn (8).

10 Likelihood Ratios

Expanding briefly on the evaluation of uncertainty on out-of-distribution points, we also observe the affect of training with robust likelihood on the 'likelihood ratio' of in and out-of-distribution (OOD) points. Similarly to how we evaluate OOD points in the main text, we use FashionMNIST dataset as out-of-distribution points for networks trained on MNIST. The likelihood ratio is calculated as the average softmax probability coming from out-of-distribution points divided by the average predictive probability of in-distribution points. Thus, a likelihood ratio of 1.0 represents predictions which are equally confident in and out of distribution. Conversely, a low likelihood ratio represents less certain predictions on out-of-distribution points. We show that for our method of training, consistently with the often used entropy measure reported in the main text, IBP training consistently improves the calibration of uncertainty on out-of-distribution points compared with normal training.

10.1 Extended Out-of-Distribution Entropy Plots

In Figure 10 we extend the out of distribution MNIST plots given in the main text to the other approximate inference techniques and find that the same result that is discussed in the main text holds for HMC and NA as well.

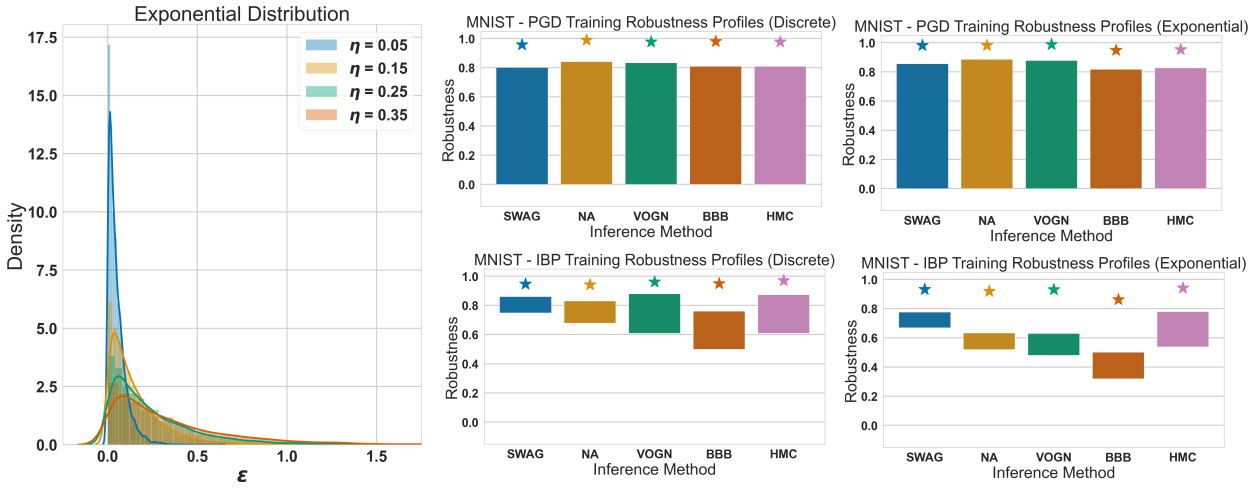

Figure 8: **Left:** Effect of varying the scale η of the exponential distribution on the density p_ϵ when training we use $\eta = 0.1$. **Right, Top Row:** Robustness profiles of networks using the robust likelihood with PGD as an approximate worst-case adversary. **Right, Bottom Row:** Robustness profiles of networks using the robust likelihood with IBP as an approximate worst-case adversary. Accuracy (plotted as star points), an empirical estimation of $\mathcal{R}_\epsilon$ obtained using PGD (upper bound of each bar), and $\mathcal{R}_\epsilon^{IBP}$ (lower bound of each bar), obtained for $\epsilon = 0.1$ on the MNIST dataset.

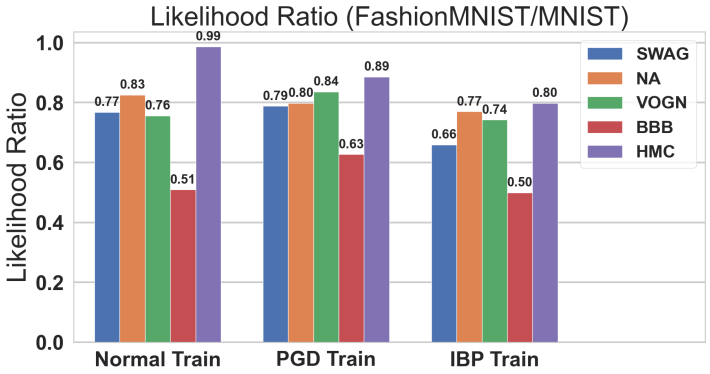

Figure 9: Likelihood Ratios using FashionMNIST as out-of-distribution samples for posteriors inferred on the MNIST datset. A likelihood ratio of 1.0 represents predictions which are equally confident in and out of distribution. Conversely, a low likelihood ratio represents less certain predictions on out-of-distribution points.

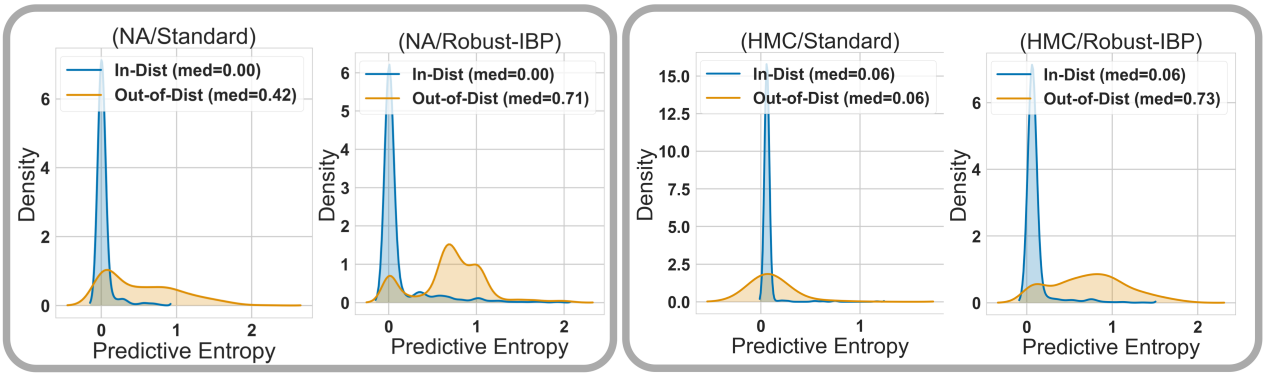

Figure 10: We plot the in-distribution (blue) and out-of-distribution (orange) predictive uncertainty. Each pair of figures corresponds to an inference method where the left figure represents the entropy distributions for standard training and the right figure represents robust IBP training. We find that robust training improves the uncertainty calibration of the network w.r.t. out-of-distribution samples.