

---

# Are Bayesian neural networks intrinsically good at out-of-distribution detection?

---

Christian Henning<sup>\*1</sup> Francesco D’Angelo<sup>\*1</sup> Benjamin F. Grewe<sup>1</sup>

## Abstract

The need to avoid confident predictions on unfamiliar data has sparked interest in *out-of-distribution* (OOD) detection. It is widely assumed that Bayesian neural networks (BNN) are well suited for this task, as the endowed epistemic uncertainty should lead to disagreement in predictions on outliers. In this paper, we question this assumption and provide empirical evidence that proper Bayesian inference with common neural network architectures does not necessarily lead to good OOD detection. To circumvent the use of approximate inference, we start by studying the infinite-width case, where Bayesian inference can be exact considering the corresponding Gaussian process. Strikingly, the kernels induced under common architectural choices lead to uncertainties that do not reflect the underlying data generating process and are therefore unsuited for OOD detection. Finally, we study finite-width networks using HMC, and observe OOD behavior that is consistent with the infinite-width case. Overall, our study discloses fundamental problems when naively using BNNs for OOD detection and opens interesting avenues for future research.

## 1. Introduction

In supervised learning, the dataset is assumed to be generated according to some unknown process  $\mathcal{D} \stackrel{i.i.d.}{\sim} p(\mathbf{x})p(\mathbf{y} | \mathbf{x})$ . The goal here is to infer from  $\mathcal{D}$  alone the distribution  $p(\mathbf{y} | \mathbf{x})$  in order to make predictions on unseen inputs  $\mathbf{x}^*$ . In the case of deep learning, this problem is approached by choosing a neural network  $f(\cdot; \mathbf{w})$  parametrized by  $\mathbf{w}$ , and predictions are made via the conditional  $p(\mathbf{y} | f(\cdot; \mathbf{w}))$  (also called the *likelihood*). Assuming

---

<sup>\*</sup>Equal contribution <sup>1</sup>Institute of Neuroinformatics, University of Zürich and ETH Zürich, Zürich, Switzerland. Correspondence to: Christian Henning <henningc@ethz.ch>, Francesco D’Angelo <fdangelo@student.ethz.ch>.

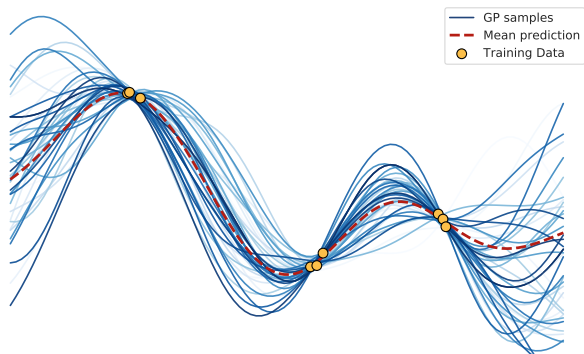


Figure 1. GP regression with an RBF kernel. The prior variance over function values is only squeezed around training points, which leaves epistemic uncertainty high in OOD regions.

that the induced class of models contains some  $\hat{\mathbf{w}}$  such that  $p(\mathbf{y} | f(\cdot; \hat{\mathbf{w}})) = p(\mathbf{y} | \mathbf{x})$  almost everywhere on the support of  $p(\mathbf{x})$ , Bayesian statistics can be used to infer plausible models  $p(\mathbf{w} | \mathcal{D})$  under the observed data given some prior knowledge  $p(\mathbf{w})$  (see MacKay (2003) for an introduction).

Bayesian inference has a multitude of benefits as it is not susceptible to overfitting and new evidence can be incorporated without requiring access to past data. Furthermore, the explicit handling of parameter uncertainty is even more important in the overparametrized regime where the increased number of degrees of freedom further extends the hypothesis class, providing an intuitive need of a Bayesian treatment, but also deceptively motivates the naive use of BNNs as OOD detectors. The issue with this argument is partly rooted in the fact, that there is no universal mathematical definition that characterizes an unseen input  $\mathbf{x}^*$  as out-of-distribution (see SM B for a detailed discussion). In this work, we boldly assume that the observed data  $\mathcal{D}$  allows us to generalize well and mimic  $p(\mathbf{y} | \mathbf{x})$  across the high-probability regions of  $p(\mathbf{x})$ . In this case, meaningful predictions can be made unless an input  $\mathbf{x}^*$  is unlikely under  $p(\mathbf{x})$  or even from outside the support of  $p(\mathbf{x})$ . These are the type of inputs we consider OOD, and providing guarantees for their detection is of utmost importance for safety-critical applications of AI and reliable predictions.

While it appears natural to tackle this problem from a generative perspective by explicitly modelling  $p(\mathbf{x})$ , this paper is solely concerned with the question of how justified it is to deploy uncertainty-based OOD detection via a Bayesian neural network.

The uncertainty captured by a BNN can be coarsely categorized into *aleatoric* and *epistemic* uncertainty. Aleatoric uncertainty is irreducible and intrinsic to the data  $p(\mathbf{y} | \mathbf{x})$ . For instance, a blurry image might not contain enough information to identify unique object classes. On the other hand, a BNN also models epistemic uncertainty by maintaining a distribution over parameters  $p(\mathbf{w} | \mathcal{D})$ . This distribution reflects the uncertainty about which hypothesis explains the data and can be reduced by observing more data. Arguably, aleatoric uncertainty is of little interest for detecting OOD inputs, while epistemic uncertainty might be useful if the following assumption holds: the hypotheses captured by  $p(\mathbf{w} | \mathcal{D})$  must agree on their predictions for in-distribution samples but disagree for OOD samples (Fig. 1). Certain Bayesian methods satisfy this assumption, e.g. Gaussian process regression with an RBF kernel (cf. Sec. 3). However, the uncertainty induced by Bayesian inference does not in general give rise to OOD capabilities. This can easily be verified by considering the following thought experiment: Assume a model class with only two hypotheses  $g(x)$  and  $h(x)$  such that  $g(x) \neq h(x) \forall x$  and data being generated according to  $g(x)$ . Once data is observed, we can commit to the ground-truth hypothesis  $g(x)$  and thus lose epistemic uncertainty in- and out-of-distribution. Finite-width neural networks on the other hand, form a powerful class of models, and are often put into context with universal function approximators (Hornik, 1991). But is this fact enough to attribute them with good OOD capabilities? The literature often seems to imply that a BNN is intrinsically good at OOD detection. For instance, the use of OOD benchmarks when introducing new methods for approximate inference creates the false impression that the true posterior is a good OOD detector (Louizos & Welling, 2017; Pawlowski et al., 2017; Krueger et al., 2017; Henning et al., 2018; Maddox et al., 2019; Ciosek et al., 2020). Our work is meant to kickstart a discussion among researchers about the properties of OOD uncertainty when performing exact Bayesian inference in neural networks. To achieve this, we contribute as follows:

- We empirically show that exact inference in infinite-width networks does not necessarily lead to desirable OOD behavior, and that these observations are consistent with results obtained via approximate inference in their finite-width counterparts.
- We study the OOD behavior of infinite-width networks on regression tasks by analysing the properties of the induced kernels. Moreover, we discuss desirable kernel features for OOD detection and empirically show that these are not present in common architectural choices.
- We emphasize that the choice of weight-space prior has a strong effect on OOD performance.
- Finally, we discuss uncertainty quantification, highlighting that the commonly used predictive entropy is confounded by aleatoric uncertainty.

## 2. Background

In this section, we briefly introduce the concepts on which we base our argumentation in Sec. 3. We start by introducing BNNs, which unfortunately rely on approximate inference. However, in the non-parametric limit and under a certain choice of prior, a BNN converges to a Gaussian process, an alternative Bayesian inference framework where exact inference is possible. The connection between BNNs and Gaussian processes will later allow us to make interesting conjectures about OOD behavior.

**Bayesian neural networks.** In supervised deep learning, we typically consider a likelihood function  $p(\mathbf{y} | f(\mathbf{x}; \mathbf{w}))$  parameterized by a neural network  $f(\mathbf{x}; \mathbf{w})$  and training data  $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ . In BNNs, we are interested in the posterior distribution of all likely networks given by:

$$p(\mathbf{w} | \mathcal{D}) \propto \prod_{i=1}^n p(\mathbf{y}_i | f(\mathbf{x}_i; \mathbf{w})) p(\mathbf{w}) \quad (1)$$

where  $p(\mathbf{w})$  is the prior distribution over weights. Crucially, when making a prediction on a test point  $\mathbf{x}^*$ , in the Bayesian approach we do not only use a single parameter  $\hat{\mathbf{w}}$  but we marginalize over the whole posterior, thus taking all possible explanations of the data into account:

$$p(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) = \int p(\mathbf{y}^* | f(\mathbf{x}^*; \mathbf{w})) p(\mathbf{w} | \mathcal{D}) d\mathbf{w} \quad (2)$$

**Gaussian process.** Gaussian processes are established Bayesian machine learning models that, despite their strong scalability limitations, can offer a powerful inference framework. Formally (Rasmussen, 2004):

**Definition 1** A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution.

Compared to parametric models, GPs have the advantage of performing inference directly in function space. A GP is defined by its mean  $m(\mathbf{x}) = \mathbb{E}[\mathbf{f}(\mathbf{x})]$  and covariance  $C(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}')) = \mathbb{E}[(\mathbf{f}(\mathbf{x}) - m(\mathbf{x}))(\mathbf{f}(\mathbf{x}') - m(\mathbf{x}'))]$ . The latter can be specified using a kernel  $k: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$  and implies a prior distribution over functions:

$$p(\mathbf{f} | X) = \mathcal{N}(\mathbf{0}, K(X, X)) , \quad (3)$$

where  $K(X, X)_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$  is the kernel Gram matrix on the training inputs  $X$ , and  $m(\mathbf{x})$  has been chosen to be

0. When observing the training data, the prior is reshaped to place more mass in the regions of functions that are more likely to have generated them, and this knowledge is then used to make predictions on unseen inputs  $X_*$ . In probabilistic terms, this operation corresponds to conditioning the joint Gaussian prior:  $p(\mathbf{f}_* | X_*, X, \mathbf{f})$ . To model the data more realistically, we assume to not have direct access to the function values but to noisy observations:  $\mathbf{y} = \mathbf{f}(\mathbf{x}) + \epsilon$  with  $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_d)$ . This assumption is formally equivalent to a Gaussian likelihood  $p(\mathbf{y} | \mathbf{f}) = \mathcal{N}(\mathbf{y} | \mathbf{f}, \sigma^2)$ . The conditional distribution on the noisy observation can then be written as (Rasmussen, 2004):

$$\begin{aligned} p(\mathbf{f}_* | X_*, X, \mathbf{y}) &= \int d\mathbf{f} p(\mathbf{f}_* | X_*, X, \mathbf{f}) p(\mathbf{f} | X, \mathbf{y}) \\ &= \mathcal{N}(\bar{\mathbf{f}}_*, C(\mathbf{f}_*)) \\ \bar{\mathbf{f}}_* &= K(X_*, X) [K(X, X) + \sigma^2 \mathbb{I}]^{-1} \mathbf{y} \\ C(\mathbf{f}_*) &= K(X_*, X_*) - K(X_*, X) [K(X, X) \\ &\quad + \sigma^2 \mathbb{I}]^{-1} K(X, X_*) \end{aligned} \quad (4)$$

where  $p(\mathbf{f} | X, \mathbf{y}) = \frac{p(\mathbf{y} | \mathbf{f}) p(\mathbf{f} | X)}{p(\mathbf{y} | X)}$ .

**The relation of infinite-width BNNs and GPs.** The connection between neural networks and GPs has recently gained significant attention. Neal (1996) showed that a 1-hidden layer BNN converges to a GP in the infinite-width limit. More recently, the work by Lee et al. (2018) extended this result to deeper networks, called *neural network Gaussian processes* (NNGP). Crucially, the kernel function of the related GP strictly depends on the used activation function. To better understand this connection we consider a fully connected network with  $L$  hidden layers  $l = 1, \dots, L$  with width  $H_l$ . For each input we use  $\mathbf{x}^l(\mathbf{x})$  to represent the post-activation with  $\mathbf{x}^0 = \mathbf{x}$  and  $\mathbf{f}^l$  the pre-activation so that  $f_i^l(\mathbf{x}) = b_i^l + \sum_{j=1}^{H_{l-1}} w_{ij}^l x_j^l$  with  $x_j^l = h(f_j^{l-1})$  and  $h(\cdot)$  a point-wise activation function. Furthermore, the weights and biases are distributed according to  $b_j^l \sim \mathcal{N}(0, \sigma_b^2)$  and  $w_{ij}^l \sim \mathcal{N}(0, \frac{\sigma_w^2}{H_l})$ . Given the independence of the weights and biases also the post-activations are independent so that the central limit theorem can be applied, and for  $H_l \rightarrow \infty$  we obtain  $f^l(\mathbf{x}) \sim \mathcal{GP}(0, C^l)$ . The covariance  $C^l$  is specified by the kernel induced by the network:

$$\begin{aligned} C^l(\mathbf{x}, \mathbf{x}') &= \overbrace{\mathbb{E} [f_i^l(\mathbf{x}) f_i^l(\mathbf{x}')] ]}^{k^l(\mathbf{x}, \mathbf{x}')} - \overbrace{\mathbb{E} [f_i^l(\mathbf{x})] \mathbb{E} [f_i^l(\mathbf{x}')] ]}_{=0} \\ &= \sigma_b^2 + \sigma_w^2 \mathbb{E}_{f_i^{l-1}} [h(f_i^{l-1}(\mathbf{x})) h(f_i^{l-1}(\mathbf{x}'))]. \end{aligned} \quad (5)$$

For the first layer we have:

$$k^0(\mathbf{x}, \mathbf{x}') = \mathbb{E} [f_i^0(\mathbf{x}) f_i^0(\mathbf{x}')] = \sigma_b^2 + \sigma_w^2 \left( \frac{\mathbf{x}^T \mathbf{x}'}{d} \right). \quad (6)$$

When using a ReLU activation function, the kernel has a closed form solution (Cho & Saul, 2009):

$$\begin{aligned} k_{ReLU}^l(\mathbf{x}, \mathbf{x}') &= \sigma_b^2 + \frac{\sigma_w^2}{2\pi} \sqrt{k^{l-1}(\mathbf{x}, \mathbf{x}) k^{l-1}(\mathbf{x}', \mathbf{x}')} \\ &\quad \left( \sin \theta_{\mathbf{x}, \mathbf{x}'}^{l-1} + (\pi - \theta_{\mathbf{x}, \mathbf{x}'}^{l-1}) \cos \theta_{\mathbf{x}, \mathbf{x}'}^{l-1} \right) \\ \theta_{\mathbf{x}, \mathbf{x}'}^l &= \cos^{-1} \left( \frac{k^l(\mathbf{x}, \mathbf{x}')}{\sqrt{k^l(\mathbf{x}, \mathbf{x}) k^l(\mathbf{x}', \mathbf{x}')}} \right) \end{aligned} \quad (7)$$

Whereas activations like sigmoid or hyperbolic tangent, require a Monte Carlo estimate of Eq. 5 (cf. Fig. S3).

### 3. Discussion and empirical observations

In the previous section, we recalled the connection between BNNs and GPs, namely that Bayesian inference in an infinite-width neural network can be studied in the GP framework (assuming a proper choice of prior). This connection allows studying how the kernels induced by architectural choices shape the prior in function space, and how these choices ultimately determine OOD behavior. In this section, we analyze this OOD behavior for traditional as well as NNGP-induced kernels. To minimize the impact of the approximations on our results, we exclusively focus on conjugate settings, i.e., regression.

**GP regression with an RBF kernel.** We first examine the predictive posterior of a GP with squared exponential (RBF) kernel  $k(\mathbf{x}_p, \mathbf{x}_q) = \exp(-\frac{1}{2l^2} \|\mathbf{x}_p - \mathbf{x}_q\|_2^2)$ , where  $l$  is a hyperparameter. Fig. 2a shows the standard deviation  $\sigma(\mathbf{f}_*) \equiv \sqrt{\text{diag}(C(\mathbf{f}_*))}$  of the predictive posterior. We can notice that  $\sigma(\mathbf{f}_*)$  nicely captures the data manifold and is thus well suited for OOD detection. This behavior can be understood by considering Eq. 4 while noting that  $k(\mathbf{x}_p, \mathbf{x}_q) = \text{const}$  if  $\mathbf{x}_p = \mathbf{x}_q$ . In this case, the variance of posterior function values can be written as  $\sigma^2(\mathbf{f}_*) = k(x_*, x_*) - \sum_{i=1}^n \beta_i(x_*) k(x_*, x_i)$  (cf. SM C), where  $\beta_i$  are dataset- and input-dependent. The second term is reminiscent of using kernel density estimation (KDE) to approximate  $p(\mathbf{x})$ , applying a Gaussian kernel, while the first term is the (constant) prior variance. Hence, the link between Bayesian inference and OOD detection can be made explicit, as the posterior variance is inversely related to the input distribution. In this view, one starts with high (prior) uncertainty everywhere, which is only reduced where data is observed. By contrast, learning a normalized generative model (e.g., using normalizing flows (Papamakarios et al., 2019)) often requires to start from an arbitrary probability distribution. Importantly, the desirable OOD behavior observed for RBFs is not generalizable to all kernels (cf. Fig. S1).

**Uncertainty quantification for OOD detection.** Uncertainty can be quantified in multiple ways, but is often measured as the entropy of the predictive posterior. The pre-

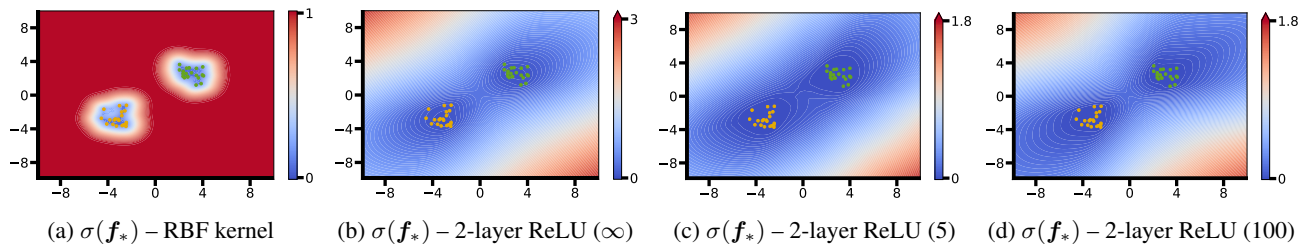


Figure 2. **Standard deviation of the predictive posterior.** We perform Bayesian inference on a mixture of two Gaussians dataset considering different priors in function space. The problem is treated as regression task to allow exact inference in combination with GPs.

dictive posterior, however, captures both aleatoric and epistemic uncertainty, which does not allow a distinction between OOD and ambiguous inputs (Mukhoti et al., 2021). While our choice of likelihood does not permit the modelling of input-dependent uncertainty (Gaussian with fixed variance), a softmax classifier can capture aleatoric uncertainty arbitrarily well. For this reason, uncertainty should be quantified in a way that allows OOD detection to be based on epistemic uncertainty only.

The function space view of GPs naturally provides such measure of epistemic uncertainty by considering the posterior variance of function values  $\sigma^2(\mathbf{f}_*)$  (illustrated in Fig. 1). Such measure can be naturally translated to BNNs by looking at the disagreement between network outputs when sampling from  $p(\mathbf{w} \mid \mathcal{D})$ . Note that in classification tasks all models drawn from  $p(\mathbf{w} \mid \mathcal{D})$  might lead to a high-entropy softmax without disagreement, and therefore only an uncertainty measure based on model disagreement prevents one from misjudging ambiguous points as OOD. As OOD detection is the focus of this paper, we always quantify uncertainty in terms of model disagreement.

**The OOD behavior induced by NNGP kernels.** Fig. 2b illustrates  $\sigma^2(\mathbf{f}_*)$  for an infinite-width 2-layer ReLU network (see Fig. S4 for other architectural choices). It is already visually apparent, that in this case the kernel is less suited for OOD detection compared to an RBF kernel. Moreover, we cannot justify why OOD detection based on  $\sigma^2(\mathbf{f}_*)$  would be principled for this kernel as the KDE analogy does not hold. This is due to two reasons: (1) the prior uncertainty  $k(x_*, x_*)$  is not constant (Fig. S6), and (2) we empirically do not observe that  $k(\mathbf{x}_p, \mathbf{x}_q)$  can be related to a distance measure (Fig. S5). We therefore argue that more theoretical work is necessary to justify the use of NNGP kernels in the context of OOD detection.

On this note, maintaining parameter uncertainty and being able to detect OOD samples are often considered crucial requirements of systems deployed in safety-critical applications. Given that Bayesian inference is not intrinsically linked to OOD detection, care should be taken to precisely communicate safety-relevant capabilities of BNNs to practi-

tioners.

**Infinite-width uncertainty is consistent with the finite-width uncertainty.** We next consider finite-width BNNs, where exact Bayesian inference is intractable. To mitigate the effects of approximate inference we resort to Hamiltonian Monte Carlo (HMC, Duane et al., 1987; Neal et al., 2011). Fig. 2c and 2d show an estimate of  $\sigma^2(\mathbf{f}_*)$  for finite-width networks. Already for moderate layer widths, the uncertainty modelled resembles the one of the corresponding NNGP. Given this close correspondence, we conjecture that the tools available for the infinite-width case are useful for designing architectural guidelines that enhance OOD detection. We explore this idea using RBF networks in SM D.

**The importance of the weight-space prior.** The correspondence between NNGPs and BNNs only applies for zero-mean priors where the variance is corrected by the layer width. However, the combination of architecture and  $p(\mathbf{w})$  induces a prior over functions (Wilson & Izmailov, 2020), which will ultimately shape posterior’s OOD uncertainty. We highlight the importance of  $p(\mathbf{w})$  in Fig. S7. Note, that studying GP kernels with respect to their OOD performance can also help designing meaningful weight-space priors for a given architecture, as  $p(\mathbf{w})$  can be learned to induce a prior in function space that mimics a GP prior (e.g., Flam-Shepherd et al., 2017).

**Conclusion.** We discussed the relation of BNNs with OOD detection from a GP perspective. Due to computational restrictions, our empirical results are limited to low-dimensional problems. However, the fact that empirically BNNs are often only marginally superior in OOD detection compared to models that do not maintain epistemic uncertainty (Snoek et al., 2019; Henning et al., 2021), indicates fundamental limitations that are not solely explained by the use of approximate inference.

Overall, this work raises doubts about the capabilities of BNNs for OOD detection and challenges the community to provide evidence that justifies the wide-spread use for that purpose.

## Acknowledgements

This work was supported by the Swiss National Science Foundation (B.F.G. CRSII5-173721 and 315230.189251), ETH project funding (B.F.G. ETH-20 19-01) and funding from the Swiss Data Science Center (B.F.G. C17-18). In particular, we would like to thank Maria R. Cervera for countless fruitful discussions and her assistance in writing the manuscript. Furthermore, we would like to thank Benjamin Ehret, João Sacramento, Simone Carlo Surace and Jean-Pascal Pfister for discussions and feedback.

## References

- Cho, Y. and Saul, L. Kernel methods for deep learning. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.
- Ciosek, K., Fortuin, V., Tomioka, R., Hofmann, K., and Turner, R. Conservative uncertainty estimation by fitting prior networks. In *International Conference on Learning Representations*, 2020.
- Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.
- Flam-Shepherd, D., Requeima, J., and Duvenaud, D. Mapping gaussian process priors to bayesian neural networks. In *NIPS Bayesian deep learning workshop*, 2017.
- Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.
- Henning, C., von Oswald, J., Sacramento, J., Surace, S. C., Pfister, J.-P., and Grewe, B. F. Approximating the predictive distribution via adversarially-trained hypernetworks. In *NeurIPS Bayesian Deep Learning Workshop*, 2018.
- Henning, C., Cervera, M. R., D’Angelo, F., von Oswald, J., Traber, R., Ehret, B., Kobayashi, S., Sacramento, J., and Grewe, B. F. Posterior meta-replay for continual learning. *arXiv*, 2021.
- Hornik, K. Approximation capabilities of multilayer feed-forward networks. *Neural Networks*, 4(2):251–257, 1991. ISSN 0893-6080. doi: [https://doi.org/10.1016/0893-6080\(91\)90009-T](https://doi.org/10.1016/0893-6080(91)90009-T).
- Krueger, D., Huang, C.-W., Islam, R., Turner, R., Lacoste, A., and Courville, A. Bayesian Hypernetworks. *arXiv:1710.04759 [cs, stat]*, October 2017. *arXiv*: 1710.04759.
- Lee, J., Sohl-dickstein, J., Pennington, J., Novak, R., Schoenholz, S., and Bahri, Y. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.
- Liu, Q., Lee, J., and Jordan, M. A kernelized stein discrepancy for goodness-of-fit tests. 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. 276–284, New York, New York, USA, 20–22 Jun 2016. PMLR.
- Louizos, C. and Welling, M. Multiplicative Normalizing Flows for Variational Bayesian Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, pp. 2218–2227. JMLR.org, 2017.
- MacKay, D. J. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.
- MacKay, D. J. C. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, USA, 2002. ISBN 0521642981.
- Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- Mukhoti, J., Kirsch, A., van Amersfoort, J., Torr, P. H., and Gal, Y. Deterministic neural networks with appropriate inductive biases capture epistemic and aleatoric uncertainty. *arXiv*, 2021.
- Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Do deep generative models know what they don’t know? In *International Conference on Learning Representations*, 2019a.
- Nalisnick, E., Matsukawa, A., Teh, Y. W., and Lakshminarayanan, B. Detecting out-of-distribution inputs to deep generative models using typicality. *arXiv preprint arXiv:1906.02994*, 2019b.
- Neal, R. M. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996. ISBN 0387947248.
- Neal, R. M. et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.
- Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *arXiv*, 2019.

- Pawlowski, N., Brock, A., Lee, M. C., Rajchl, M., and Glocker, B. Implicit weight uncertainty in neural networks. *arXiv*, 2017.
- Pimentel, M. A., Clifton, D. A., Clifton, L., and Tarassenko, L. A review of novelty detection. *Signal Processing*, 99: 215–249, 2014.
- Rasmussen, C. E. *Gaussian Processes in Machine Learning*, pp. 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-28650-9. doi: 10.1007/978-3-540-28650-9\_4.
- Snoek, J., Ovadia, Y., Fertig, E., Lakshminarayanan, B., Nowozin, S., Sculley, D., Dillon, J., Ren, J., and Nado, Z. Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pp. 13969–13980, 2019.
- Williams, C. K. Computing with infinite networks. *Advances in neural information processing systems*, pp. 295–301, 1997.
- Wilson, A. G. and Izmailov, P. Bayesian deep learning and a probabilistic perspective of generalization. In *Advances in Neural Information Processing Systems*, 2020.
- Zimek, A. and Filzmoser, P. There and back again: Outlier detection between statistical reasoning and data mining algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(6):e1280, 2018.

# Supplementary Material: Are Bayesian neural networks intrinsically good at out-of-distribution detection?

Christian Henning\*, Francesco D’Angelo\* and Benjamin F. Grewe

## A. Additional experiments and results

In this section, we report additional experiments and results.

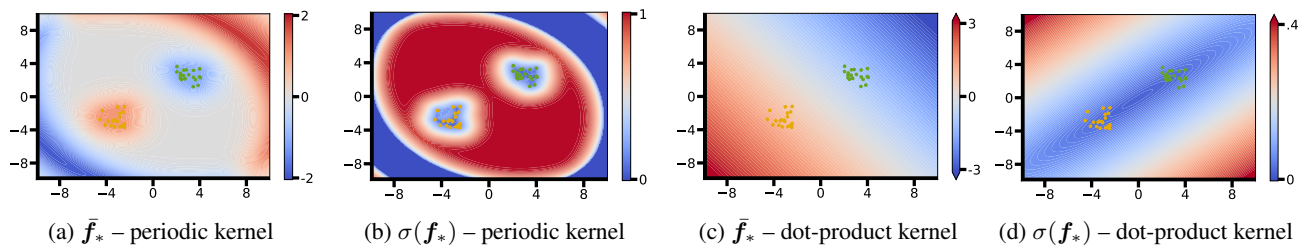


Figure S1. Mean  $\bar{f}_*$  and standard deviation  $\sigma(f_*)$  of the posterior  $p(f_* | X_*, X, \mathbf{y})$  visualized for GPs with traditional kernels. In contrast to a GP with RBF kernel (cf. Fig. 2), the kernels used here induce epistemic uncertainty  $\sigma(f_*)$  that is undesirable for the purpose of OOD detection. Therewith, this plot highlights the importance of the function space prior for the task of detecting anomalies using predictive uncertainty.

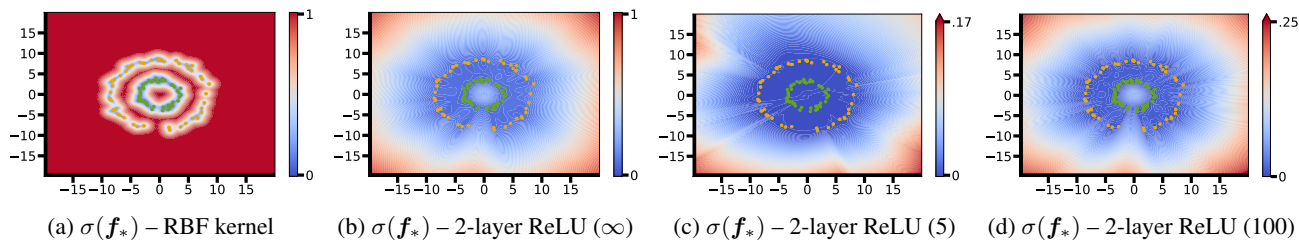


Figure S2. **Standard deviation of the predictive posterior.** We perform Bayesian inference on a datasets composed by two concentric rings comparing a GP with RBF kernel and ReLU networks in the finite and infinite regimes (in analogy to Fig. 2). Only the function space prior induced by an RBF kernel causes epistemic uncertainties that allow outlier detection in the center or in between the two circles.

Are Bayesian neural networks intrinsically good at out-of-distribution detection?

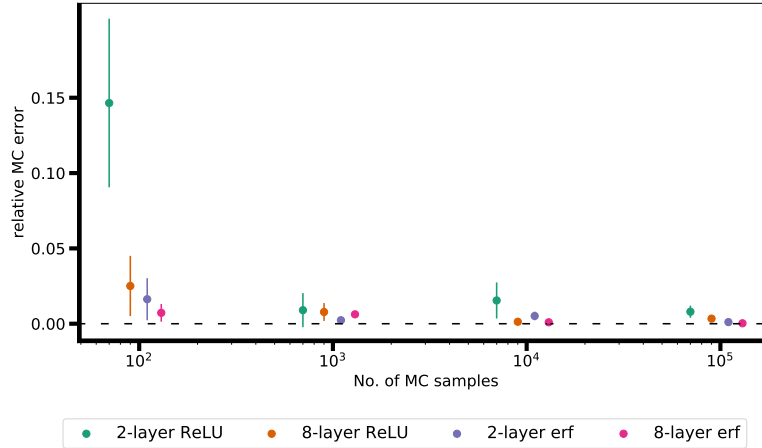


Figure S3. Eq. 5 requires estimation whenever no analytic kernel expression is available (for instance, when using a hyperbolic tangent nonlinearity as in Fig. S4). Here, we visualize the error caused by this approximation for ReLU and error function (erf) networks when computing kernel values  $k(x_*, x_*)$ . Eq. 5 requires a recursive estimation of expected values, where we estimate each of them using  $N$  samples ( $N \in [10^2, 10^3, 10^4, 10^5]$ ). Note, that even small errors can cause eigenvalues of the kernel matrix to become negative. However, with our chosen likelihood variance of  $\sigma = 0.02$  we experience no numerical instabilities during inference, and obtain consistent results using either analytic or estimated ( $N = 10^5$ ) kernel matrices.

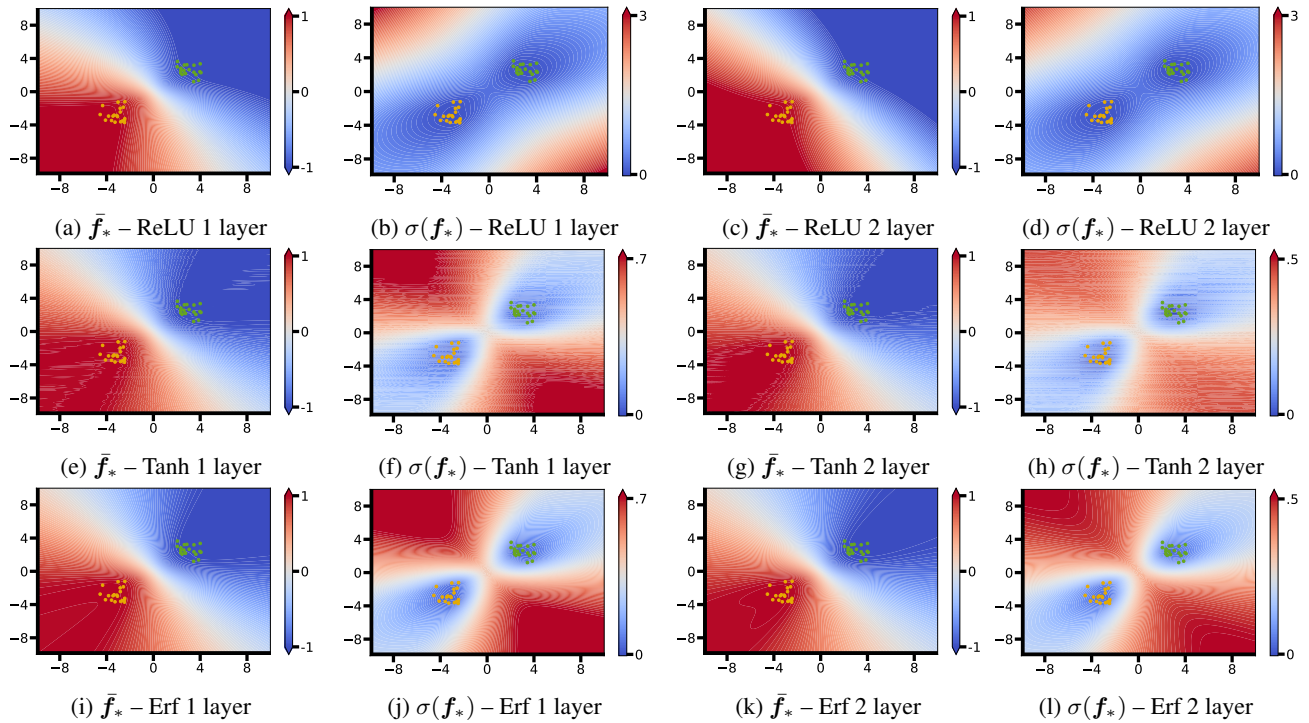


Figure S4. Mean  $\bar{f}_*$  and standard deviation  $\sigma(f_*)$  of the posterior  $p(f_* | X_*, X, y)$  visualized for GP regression using NNGP kernels for various architectural choices, such as number of layers or non-linearity. Note, that Tanh and Erf nonlinearities are quite similar in shape, which is reflected in the similar predictive posterior that is induced by these networks. We use the analytically known kernel expression for the Erf kernel (Williams, 1997), and use MC sampling for the Tanh network.

Are Bayesian neural networks intrinsically good at out-of-distribution detection?

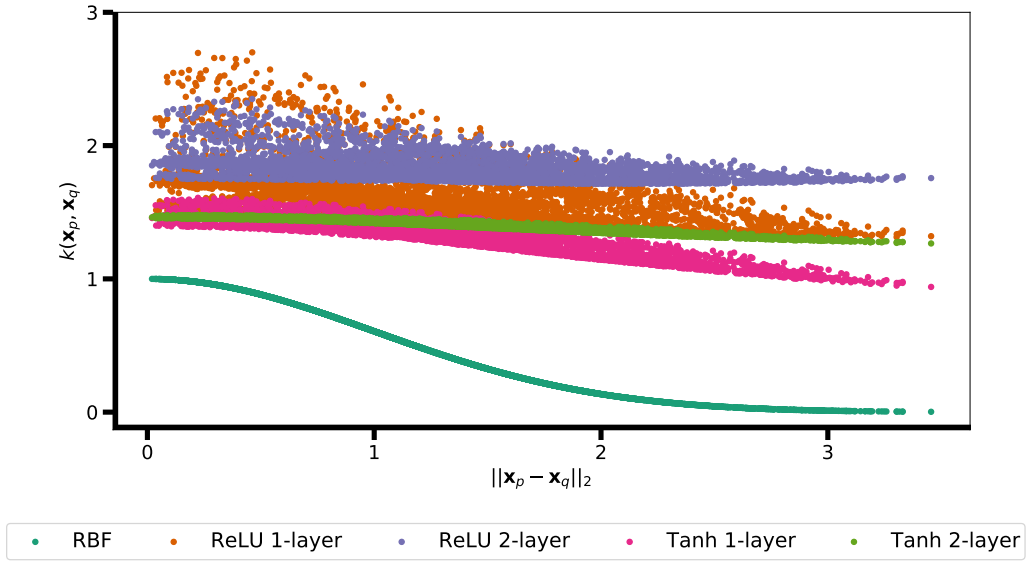


Figure S5. Kernel values plotted as a function of the Euclidean distance of points (using pairs of training points from the two Gaussian mixtures dataset). As outlined in Sec. 3 and Sec. C, the interpretation of an RBF kernel as Gaussian kernel that can be used in a KDE of  $p(\mathbf{x})$  is important to justify implied OOD capabilities. Unfortunately, the kernels induced by common architectures do not seem to be distance-aware and are thus not useful for KDE.

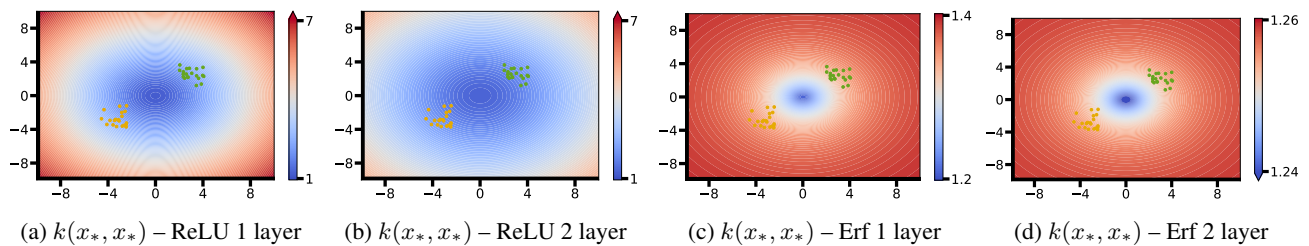


Figure S6. NNGP kernel values for various architectural choices. Note, that the NNGP kernel value  $k(x_*, x_*)$  represents the prior variance of function values under the induced GP prior at the location  $x_*$  (cf. Eq. 3). As emphasized in the main text,  $k(x_*, x_*)$  is constant for an RBF kernel, which has important implications for OOD detection, such as that a priori (before seeing any data) all points are treated equally. This is not the case for the Relu Kernel, which has an angular dependence and depends on the input norm (cf. Eq. 6 and Eq. 7). The kernel induced by networks using a error function (Erf) as nonlinearity seems to be more desirable in this respect (note the scale of the colorbars).

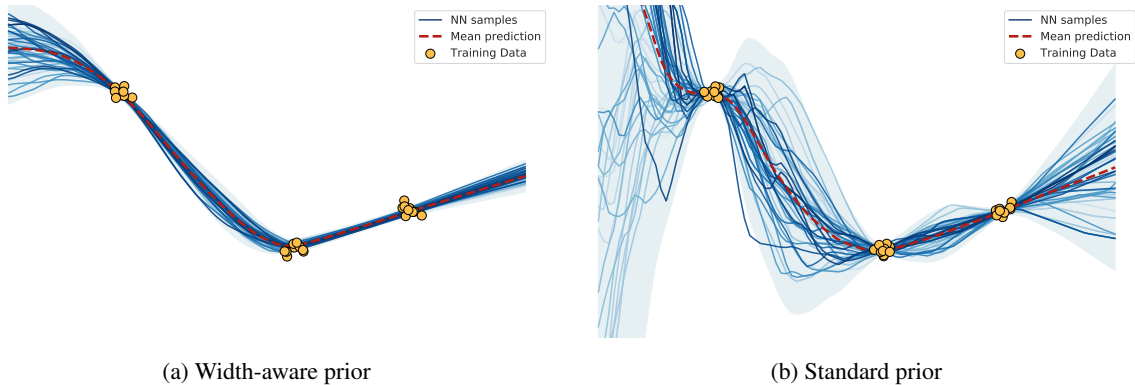


Figure S7. This figure illustrates the importance of the choice of weight prior  $p(w)$ . Here, we perform 1d regression using HMC with either a width-aware prior  $\mathcal{N}(0, \frac{\sigma_w^2}{H_l})$  or a standard prior  $\mathcal{N}(0, \sigma_w^2)$ .

### B. What is an out-of-distribution input?

The corresponding Wikipedia article uses the following formulation: "Determining whether or not an observation is an outlier is ultimately a subjective exercise" (Zimek & Filzmoser, 2018).<sup>1</sup> Pimentel et al. (2014) reviews methods for outlier detection, putting them coarsely into 5 categories: (1) probabilistic, (2) distance-based, (3) reconstruction-based, (4) domain-based, and (5) information-theoretic. Our focus lies on a probabilistic characterization of an OOD point, where a statistical criterion allows to decide whether a given input is significantly different from the observed training population.<sup>2</sup> In this section, we discuss pitfalls regarding obvious choices for such a criterion.

We would like to start by emphasizing why a mathematical characterization of "out of distribution" is desirable. As indicated in Sec. 1, applying Bayesian statistics to neural networks should ultimately serve the goal of providing safety guarantees for critical applications. Any such guarantee will rely on assumptions (e.g., the assumption that the neural network architecture is sufficient to encompass the ground-truth function). However, it is important to be aware of any such assumptions, such that their validity can be tested. The scope of this work is meant to encourage research towards design principles that facilitate BNNs that are more reliable at OOD detection. We argue that the effectiveness of such BNNs can only be thoroughly examined if the concept of OOD is clearly defined.

There are many definitions that could be considered for a point to be OOD as we will outline below. Any of these definitions may change the notion of OOD for individual inputs and will therefore affect how a BNN should be designed such that predictive uncertainty adheres to the underlying OOD definition. Considering a generative process  $p(x)$ , the first question arising is regarding the regions outside the support of  $p(x)$  or even the space outside the manifold where samples  $x$  are defined on. For instance, assume the data to be images embedded on a lower-dimensional manifold. Are points outside this manifold clearly OOD, given that minor noise corruptions are likely to leave the manifold? Even disregarding these topological issues solely focusing on the density  $p(x)$ , makes the distinction between in- and out-of-distribution challenging. For instance, a threshold-criterion on the density might cause samples in a zero probability region to be considered in-distribution (Nalisnick et al., 2019a). To overcome such challenges, one may resort to concepts from information theory, such as the notion of a typical set (MacKay, 2002; Nalisnick et al., 2019b). Unfortunately, an OOD criterion based on this notion would require looking at sets rather than individual points. We hope that this short outline highlights the challenges regarding the definition of OOD, but also clarifies that a proper definition is relevant when assessing OOD capabilities on high-dimensional data, where a visual assessment as in Fig. 2 is not possible.

### C. Note on the relation between GP regression and kernel density estimation

GP regression with an RBF kernel allows a direct understanding of the OOD capabilities that a Bayesian posterior may possess, as the a priori uniform epistemic uncertainty is reduced in direct correspondence to density of  $p(x)$ .

<sup>1</sup><https://en.wikipedia.org/wiki/Outlier>, accessed: 2021-06-01

<sup>2</sup>Note, that we are not considering the equally important problem of outlier detection within the training set (Zimek & Filzmoser, 2018).

## Are Bayesian neural networks intrinsically good at out-of-distribution detection?

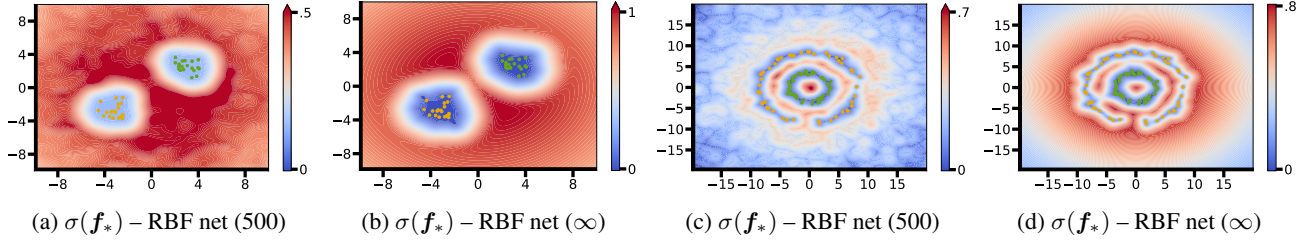


Figure S8. Standard deviation  $\sigma(\mathbf{f}_*)$  of the posterior  $p(\mathbf{f}_* | X_*, X, \mathbf{y})$  visualized for a one hidden-layer RBF network with 500 hidden units (a,c) and the correspondent infinite GP (b,d).  $\sigma(\mathbf{f}_*)$  has been estimated using HMC for the finite cases.

Below, we rewrite the variance of an input  $x_*$  as defined in Eq. 4:

$$\sigma^2(\mathbf{f}_*) = k(x_*, x_*) - \sum_{i=1}^n \beta_i(x_*) k(x_*, x_i) \quad (8)$$

with  $\beta_i(x_*) = \sum_{j=1}^n (K(X, X) + \sigma^2 \mathbb{I})_{ij}^{-1} k(x_*, x_j)$ .

Note, that  $k(x_*, x_*)$  is a positive constant for an RBF kernel, and that  $\beta_i(x_*) \geq 0$ . Furthermore, as the RBF kernel  $k(x_*, x_j)$  behaves exponentially inverse to the distance between  $x_*$  and  $x_j$ ,  $\beta_i(x_*)$  is approximately zero for all  $x_*$  that are far from all training points. In addition, a training point  $x_i$  can only decrease the prior variance if it is close to  $x_*$ . This analogy closely resembles the philosophy of KDE, which becomes an exact generative model in the limit of infinite data and bandwidth  $l \rightarrow 0$ .

### D. The kernel induced by RBF networks

Commonly, we consider neural networks where computational units (neurons) apply an affine transformation to their inputs followed by a non-linear function. Another class of neural networks are formed by radial basis function (RBF) networks, which further enhance the connection with Gaussian processes. Indeed the use of the exponential activation function, while being an uncommon choice in the deep learning community, can offer intriguing properties for OOD detection that closely resemble the ones of a GP with RBF kernel. In particular, units in a finite RBF network perform a linear combination of radial basis functions:

$$f(\mathbf{x}) = \sum_{j=1}^H w_j \exp\left(-\frac{1}{2\sigma_g^2} \|\mathbf{x} - \boldsymbol{\mu}_j\|^2\right) + b$$

where the centers  $\boldsymbol{\mu}_j$ , the bias  $b$  and the coefficients  $w_j$  are learned. Analogous to the NNGP case in Sec. 2 such networks converge to a GP with  $f(\mathbf{x}) \sim \mathcal{GP}(0, C)$  in the infinite-width case when assuming a Gaussian prior over the centers  $\boldsymbol{\mu}_j \sim \mathcal{N}(0, \sigma_\mu^2 \mathbb{I})$  and  $w_j \sim \mathcal{N}(0, \sigma_w^2)$ . An analytical expression for the correspondent kernel (Williams, 1997) in the case of a single hidden layer is given by:

$$\begin{aligned} C(\mathbf{x}, \mathbf{x}') &= \overbrace{\mathbb{E}[f^l(\mathbf{x})f^l(\mathbf{x}')] }^{k^l(\mathbf{x}, \mathbf{x}')} - \overbrace{\mathbb{E}[f^l(\mathbf{x})] \mathbb{E}[f^l(\mathbf{x}')] }^{=0} \\ &= \sigma_b^2 + \sigma_w^2 \mathbb{E} \left[ \exp\left(-\frac{1}{2\sigma_g^2} \|\mathbf{x} - \boldsymbol{\mu}_j\|^2\right) \exp\left(-\frac{1}{2\sigma_g^2} \|\mathbf{x}' - \boldsymbol{\mu}_j\|^2\right) \right] \end{aligned}$$

Where the last expectation has a closed form solution so that:

$$C(\mathbf{x}, \mathbf{x}') = \sigma_b^2 + \sigma_w^2 \left(\frac{\sigma_e}{\sigma_\mu}\right)^d \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma_m^2}\right) \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma_s^2}\right) \exp\left(-\frac{\|\mathbf{x}'\|^2}{2\sigma_m^2}\right)$$

where  $d$  is the dimension of the data  $\mathbf{x}$ ,  $1/\sigma_e^2 = 2/\sigma_g^2 + 1/\sigma_\mu^2$ ,  $\sigma_s^2 = 2\sigma_g^2 + \sigma_g^4/\sigma_\mu^2$  and  $\sigma_m^2 = 2\sigma_\mu^2 + \sigma_g^2$ . It is important to notice that there is an equivalence with the RBF kernel up to a constant when the inputs have constant norm. In addition, the kernel becomes proportional to the RBF kernel in the limit  $\sigma_\mu^2 \rightarrow \infty$  (Williams, 1997).

We tested the OOD performance of a finite-width RBF neural network in Fig. S8a and S8c using HMC and with  $\sigma_\mu^2 = 100$ . For moderate norms in input space, the OOD behavior closely assimilates the behavior obtained when using an RBF kernel as in Fig. 2a. Crucially, the network is able to detect OOD points in the center and in-between region (in contrast to common neural network architectures). The OOD behavior for inputs with a large norm can be improved by further increasing  $\sigma_\mu^2$ . We also report the correspondent infinite-width limit Fig. S8d and S8b.

### E. Sampling in-distribution data via epistemic uncertainty

Relating the epistemic uncertainty induced by a BNN to  $p(\mathbf{x})$  opens up interesting new possibilities. As we have shown in Sec. C, the epistemic uncertainty of a GP with RBF kernel can be related to a KDE approximation of  $p(\mathbf{x})$ . Therefore, one may consider the epistemic uncertainty as an energy function to create a generative model from which to sample from the input space.

Moreover, this approach may be used to empirically validate the OOD capabilities of a BNN. Indeed, OOD performance is commonly validated by selecting a specific set of known OOD datasets. However, as the OOD space comprises everything except the in-distribution data, it is infeasible to gain good coverage on high-dimensional data with such testing approach. We therefore suggest a reverse approach that checks the consistency of certainty regions instead of the uncertain ones by sampling via the epistemic uncertainty. Indeed, if these samples and thus the generative model based on the uncertainty are consistent with the in-distribution data and their distribution then a strong indication for trustworthiness is provided. More specifically a sample-based discrepancy measure (Liu et al., 2016; Gretton et al., 2012) can be used to compare the generated samples and training data points and quantitatively assess like this to which extent the epistemic uncertainty is related to  $p(x)$ .

### F. Experimental details

In this section, we report details on our implementation for the experiments conducted in this work. In all two-dimensional experiments: the likelihood is fixed to be Gaussian with variance 0.02. Unless noted otherwise, the prior in weight space is always a width-aware centered Gaussian with  $\sigma_w = 1.0$  and  $\sigma_b = 1.0$  except for the experiments involving RBF networks where we select  $\sigma_w = 200.0$ . The RBF kernel bandwidth was fixed to 1.0 in all corresponding experiments.

**Gaussian Mixture** We created a two-dimensional mixture of two Gaussians with means  $\mu_1 = (-2, -2)$ ,  $\mu_2 = (2, 2)$  and covariance  $\Sigma = 0.5 \cdot \mathbb{I}$  and sampled 20 training datapoints.

**Two rings** We uniformly sampled 50 training datapoints from two rings centred in  $(0, 0)$  with inner and outer radii  $R_{i,1} = 3, R_{o,1} = 4, R_{i,2} = 8, R_{o,2} = 9$ , respectively.

**HMC** We use 5 parallel chains for 5000 steps with each constituting 50 leapfrog steps with stepsize 0.001 for width 5 and 0.0001 for width 100. We considered a burn-in phase of 1000 steps and collected a total of 1000 samples.