Proceedings of the 7th Symposium on Advances in Approximate Bayesian Inference, 2025 1–10

Post-Hoc Uncertainty Quantification in Pre-Trained Neural Networks via Activation-Level Gaussian Processes

Anonymous Authors
Anonymous Institution

Abstract

Uncertainty quantification in neural networks through methods such as Dropout, Bayesian neural networks and Laplace approximations is either prone to underfitting or computationally demanding, rendering these approaches impractical for large-scale datasets. In this work, we address these shortcomings by shifting the focus from uncertainty in the weight space to uncertainty at the activation level, via Gaussian processes. More specifically, we introduce the Gaussian Process Activation function (GAPA) to capture neuron-level uncertainties. Our approach operates in a post-hoc manner, preserving the original mean predictions of the pre-trained neural network and thereby avoiding the underfitting issues commonly encountered in previous methods. We propose two methods. The first, GAPA-Free, employs empirical kernel learning from the training data for the hyperparameters and is highly efficient during training. The second, GAPA-Variational, learns the hyperparameters via gradient descent on the kernels, thus affording greater flexibility. Empirical results demonstrate that GAPA-Variational outperforms the Laplace approximation on most datasets in at least one of the uncertainty quantification metrics.

1. Introduction

Deep neural networks (DNNs) have achieved state-of-the-art performance in a wide range of pattern recognition tasks (Krizhevsky et al., 2012; Kenton and Toutanova, 2019; Mnih et al., 2015; Hinton et al., 2012; Litjens et al., 2017). However, traditional DNNs do not quantify epistemic uncertainty, limiting their reliability in risk-sensitive applications such as autonomous driving (Shafaei et al., 2018), healthcare (Begoli et al., 2019), and finance (Blasco et al., 2024). To address this limitation, numerous surrogate methods have been developed for downstream decision-making under uncertainty, particularly for anomaly detection and out-of-distribution detection (Li et al., 2023; Liu et al., 2023). Yet, a more principled Bayesian approach has been proposed to model uncertainty directly. This has led to methods that approximate distributions over weight space, including Bayesian Neural Networks (Neal, 2012), deep ensembles (Lakshminarayanan et al., 2017), and Markov Chain Monte Carlo methods. Additionally, regularization-based methods such as Dropout (Gal and Ghahramani, 2016) and SWAG (Maddox et al., 2019), as well as explicit modeling of weight uncertainty (Blundell et al., 2015), have shown promise in improving uncertainty estimates in deep learning models. However, there are many challenges that hinder the widespread applications of Bayesian modelling: In general, these methods are computationally expensive or even intractable in practice, for instance requiring the training of multiple DNNs or learning a distribution over each weight (Graves, 2011; Hernández-Lobato and Adams, 2015). With the rise of large pre-trained models in many domains like computer vision and natural language, the need to incorporate uncertainty-aware methods already during the model training phase is another limiting factor in their applications (Fort et al., 2019; Izmailov et al., 2021). Even methods, such as Monte-Carlo dropout, which may be present during training to act as a regularizer, require multiple forward passes to generate samples (Gal and Ghahramani, 2016; Neal, 2012; Lakshminarayanan et al., 2017). In addition, many Bayesian methods tend to suffer from underfitting, because uncertainty modelling is often inherently linked to regularization (mostly via the prior) (Wenzel et al., 2020; Osawa et al., 2019). Recently, Laplace approximations have become popular, arguably because they can be applied as a post-processing method to a pre-trained neural network without affecting its prediction and empirically capture uncertainty well without requiring sampling. Nevertheless, they demand the calculation of the Jacobian, which is computationally intensive. In addition, for scalability reasons, they are typically only employed in the last layer of a model, which potentially hinders their flexibility (Daxberger et al., 2021; Ortega et al., 2023).

In this work, we approach this problem from a different perspective: **What if we shift our focus from uncertainty in the weight space to uncertainty in the activations?** Specifically, we model uncertainty at each neuron's postactivation by fitting a one-dimensional Gaussian process to each neuron in the first layer. This approach is inexpensive to fit, and can be applied to pre-trained neural networks, without the need of re-training or fine-tuning. The second key ingredient is to propagate the obtained a uncetainties at the GP-infused layer (GAPA) through the network using deterministic propagation rules akin to determistic variational inference (Wu et al., 2018). Unlike Laplace approximation this combination allows us to model uncertainty at any layer of the network. The method is purely post-hoc (it only needs access to the pre-trained model and some training data), does not require fine-tuning of the model, and, unlike for instance dropout, can express uncertainty in a single forward-pass. Importantly, infusing uncertainty in this way does not change the original prediction of the pre-trained model in any way, thereby preserving the models predictive quality.

Specifically, we propose two Gaussian Process Activation function (GAPA) methods. The first, GAPA-Free, is a cost-effective approach that employs empirical kernel methods to compute the hyperparameters of the Gaussian process. The second, GAPA-Variational, uses variational inducing points to learn the hyperparameters, thereby allowing for greater flexibility. Our contributions are as follows:

- A post-hoc method for pre-trained neural networks that extends them through uncertainty modelling without affecting their predictions.
- A delta approximation method to propagate the uncertainty from the activation space to the output space.
- Empirical demonstration that GAPA—and in particular GAPA-Variational—delivers exceptional performance in uncertainty quantification, outperforming Laplace approximations on most datasets.
- A novel approach to uncertainty quantification by focussing on modelling the uncertainty at the activation level rather than in the weight space.

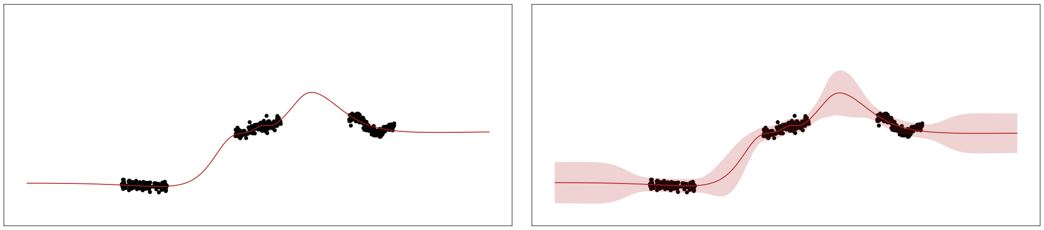

Figure 1: (Left) The architecture of the pre-trained backbone neural network. (Right) The GAPA module, applied post-hoc to the first layer to quantify uncertainty without modifying the original predictions. Illustration based on a toy regression problem from (Ortega et al., 2023).

2. Model Proposition: GAPA + Uncertainty Propagation

We begin by presenting the GAPA method, which aims to quantifiy uncertainty in a pre-trained neural network. We assume the network was first trained in a supervised manner on a dataset $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N}$. Then, to estimate uncertainty, we augment the network by applying a Gaussian Process (GP) to the output of each neuron in a layer. To highlight the generality of the approach we assume here, that this method is applied to the first hidden layer of the network. Figure 1 illustrates the backbone network and the GAPA module.

2.1. Pretrained Neural Network

Consider a standard feedforward neural network with L layers: For $l = 0, \ldots, L$, the $(l+1)$-th layer contains D_l neurons with weight matrix $W^l \in \mathbb{R}^{D_l \times D_{l-1}}$, biases $b^l \in \mathbb{R}^{D_l}$ and activation function a^l. For an input $x \in \mathbb{R}^{D_0}$, the network's prediction is given by

$$\hat{y}_{\mathbf{x}} = W^L a^L \Big(W^{L-1} a^{L-1} \big(\cdots a^1 (W^0 x + b^0) \cdots \big) + b^{L-1} \Big) + b^L.$$

This pre-trained network is optimised using standard supervised learning on $\mathcal{D}$, and its parameters are subsequently fixed.

2.2. GAPA: Gausisan Process Activations

To quantify the uncertainty of a pre-trained network without affecting its mean predictions, we attach an independent one-dimensional GP to each neuron in the first layer. Here, the pre-trained network (with fixed parameters) has been optimised on the dataset $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N}$ using standard supervised learning. Let $X := W^0 x + b^0 \in \mathbb{R}^{D^1}$ denote the neurons of the first layer. For $d \in \{1, \ldots, D^1\}$, let $Y_d := a^1(X_d)$ be the activation of the d-th neuron. We introduce uncertainty at the activation-level by replacing $a^1(X_d)$ with a GP $f_d(X_d) + \epsilon_d$. Here, we assume a GP prior $f_d \sim \mathcal{GP}(m_d, k_d)$, with mean function $m_d(X_d) := a^1(X_d)$, and a covariance kernel k_d (specifically, the RBF kernel with hyperparameters learned via an empirical method; see Appendix A for further details). Denote the neurons and activations of the training data at the first layer by $\mathbf{X}$ and $\mathbf{Y}_d = a^1(\mathbf{X}_d)$. The posterior mean is computed as

$$\mu_d(X_d) = m_d(X_d) + k_d(X_d, \mathbf{X}_d) \Big[K_d(\mathbf{X}_d, \mathbf{X}_d) + \sigma_n^2 I_N \Big]^{-1} \Big(\mathbf{Y}_d - m_d(\mathbf{X}_d) \Big).$$

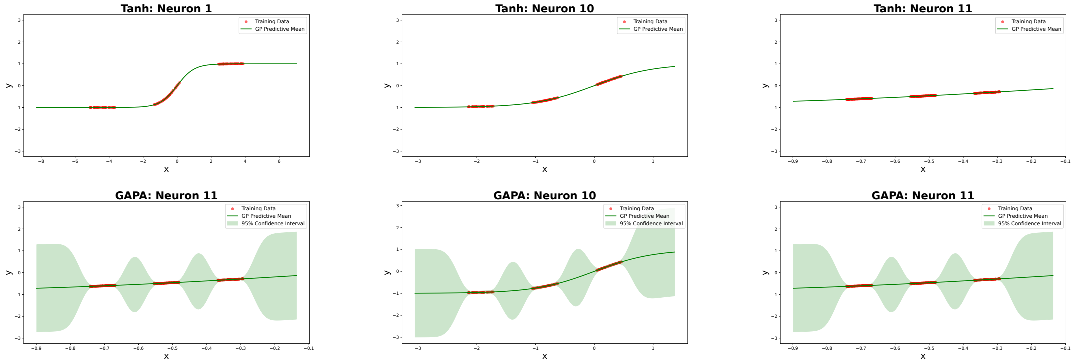

Figure 2: Baseline activations (Top) versus GAPA activations (bottom) for neurons 1, 10, 11. GAPA preserves the mean activation while providing an uncertainty estimate.

As we have $Y_d = m_d(X_d)$ by construction, it follows that $\mu_d(X_d) = m_d(X_d) = a^1(X_d)$.

Hence the pre-trained network's original activation is preserved. The posterior covariance

$$\Sigma_d(X_d, X'_d) = k_d(X_d, X'_d) - k_d(X_d, \mathbf{X}_d)\Big[K_d(\mathbf{X}_d, \mathbf{X}_d) + \sigma_n^2 I_N\Big]^{-1} k_d(\mathbf{X}_d, X'_d),$$

quantifies the epistemic uncertainty in the d-th neuron's activation. Note, that this doesn't depend on the prior mean. As shown in Figure 2 for neurons 1, 10, and 11, the GAPA model preserves the baseline activations while adding a principled uncertainty estimate. In summary, by using a GP whose prior mean is set equal to the neuron's true activation (i.e. its label), we preserve the pre-trained network's mean predictions while simultaneously providing a rigorous uncertainty epistemic estimate via the GP's posterior covariance.

2.3. Propagating the Variance through the Network

Since the GP at the first layer is constructed to preserve the pre-trained network's mean activations, the mean forward pass remains identical to that of the pre-trained model. We now need to define a variance-forward path. For this we identify two scenarios: linear layers (such as in dense and convolutional layers) and non-linear activation functions.

Linear Transformation of Variance. Since a linear transformation of a Gaussian remains Gaussian, if the input variance is Σ_a and the linear layer applies a transformation $z = Wa$, then the resulting variance is given by $\Sigma_z = W\,\Sigma_a\,W^\top$.

Propagation Rules for Non-Linear Activations. For a non-linear activation $y = g(z)$ applied to a Gaussian random variable $z \sim \mathcal{N}(\mu, \sigma^2)$, we approximate $g(z)$ by a first-order Taylor expansion (delta approximation)

$$g(z) \approx g(\mu) + g'(\mu)(z - \mu).$$

Since $z - \mu \sim \mathcal{N}(0, \sigma^2)$, this yields an approximate variance of $\mathrm{Var}(y) \approx (g'(\mu))^2\,\sigma^2$.

Overall Variance Propagation. By sequentially applying the linear transformation rule for variance and the delta approximation for non-linear activations, we obtain a tractable, layer-wise method for propagating uncertainty from the first layer (where the GP is applied) to the final network output.

2.4. GAPA-Free: Linear Scaling of the Output Variance

After propagating uncertainty to the network output, we refine the variance using a simple linear transformation:

$$\mathrm{Var}_{\mathrm{final}} = \theta_1 \ \mathrm{Var}_{\mathrm{output}} + \theta_2,$$

where θ_1 (a scaling factor) and θ_2 (an offset) are learned to capture any residual uncertainty. This calibration is computationally efficient since it involves only two parameters and requires no additional backpropagation through the network.

2.5. GAPA-Variational

In GAPA-Variational, rather than applying a fixed linear scaling, the GP variational parameters (similar to those used in variational GPs (Titsias, 2009)) are optimized via maximum likelihood. For each neuron d, we assume a GP prior $f_d \sim \mathcal{GP}\big(m_d, k_d\big)$ with $m_d(X_d) = a^1(X_d)$ (i.e. the neuron's activation) and a covariance kernel k_d (e.g. the RBF kernel with empirically determined hyperparameters). We introduce inducing variables $\mathbf{u}_d$ with fixed inducing inputs $\mathbf{Z}_d = \mathbf{X}_d$ (taken from the training data of the first layer) and set the inducing mean to $m_d(\mathbf{Z}_d) = a^1(\mathbf{Z}_d)$. The corresponding variational distribution is defined as $q(\mathbf{u}_d) = \mathcal{N}\big(m_d(\mathbf{Z}_d),\, S_d\big)$, where S_d (the variational covariance) and the kernel hyperparameters θ_d are learned. Let y_i denote the target for the ith input, and let μ_i and σ_i^2 be the predictive mean and variance obtained by propagating the GP uncertainties through the network (using, for example, the delta approximation). Because the GP prior mean is fixed to the pre-trained activation, the posterior mean remains unchanged and only the uncertainty (variance) is learned. Consequently, the overall training objective is the Gaussian negative log-likelihood (NLL) $\mathcal{L} = \sum_{i=1}^{N} \frac{1}{2} \log\big(2\pi\sigma_i^2\big) + \frac{(y_i - \mu_i)^2}{2\sigma_i^2}$.

This loss function is optimized by backpropagating the NLL from the network's final output while keeping the pre-trained network weights fixed. In this way, GAPA-Variational provides a flexible, data-driven uncertainty estimate through the learned GP covariance, all while preserving the original mean predictions of the pre-trained network.

3. Results

We compare GAPA's predictive distribution with state-of-the-art Laplace-based methods for post-hoc uncertainty quantification in pre-trained networks—including VaLLA, LLA variants, and ELLA (Daxberger et al., 2021; Izmailov et al., 2020; Ortega et al., 2023)—on three benchmark regression datasets: (i) the UCI Year dataset, (ii) the US flight delay (Airline) dataset (Dutordoir, 2020), and (iii) the Taxi dataset (Salimbeni and Deisenroth, 2017). We follow the original train/test splits used in prior studies.

Table 1 summarizes the performance of our proposed models compared to state-of-the-art post-processing methods on several regression datasets. Our evaluation metrics include Negative Log-Likelihood (NLL), Continuous Ranked Probability Score (CRPS) (Gneiting and Raftery, 2007), and the Centered Quantile Metric (CQM) (Ortega et al., 2023). In the table, the best values are highlighted in purple, the second-best in teal, and the third-best in bronze. Our experimental results show that both GAPA-Free and GAPA-Variational achieve competitive performance. Notably, GAPA-Variational consistently enhances uncertainty

Table 1: Results on regression datasets. Best values are in purple, second-best in teal, and third-best in bronze. An asterisk (*) indicates a last-layer LLA variant.

Model	**Airline**			**Year**			**Taxi**		
	NLL	**CRPS**	**CQM**	**NLL**	**CRPS**	**CQM**	**NLL**	**CRPS**	**CQM**
MAP	5.087	18.436	0.158	3.674	5.056	0.164	3.763	3.753	0.227
LLA Diag	5.096	18.317	0.144	3.650	4.957	0.122	3.714	3.979	0.270
LLA KFAC	5.097	18.317	0.144	3.650	4.955	0.121	3.705	3.977	0.270
LLA*	5.097	18.319	0.144	3.650	4.954	0.120	3.718	3.965	0.270
LLA* KFAC	5.097	18.317	0.144	3.650	4.954	0.120	3.705	3.977	0.270
ELLA	5.086	18.437	0.158	3.674	5.056	0.164	3.753	3.754	0.227
VaLLA 100	4.923	18.610	0.109	3.527	5.071	0.084	3.287	3.968	0.188
VaLLA 200	4.918	18.615	0.107	3.493	5.026	0.076	3.280	3.993	0.188
GAPA-Free	5.083	18.394	0.115	3.644	4.909	0.084	3.668	4.01	0.274
GAPA-Variational	5.067	18.282	0.135	3.545	4.796	0.053	3.268	3.552	0.154

quantification. For example, on the *Airline* dataset, it attains the best CRPS while its NLL and CQM values rank among the top three. On the *Year* dataset, GAPA-Variational records the best CRPS and CQM scores with a competitive NLL. Most importantly, on the *Taxi* dataset, it outperforms all other methods across all metrics. These findings indicate that our approach successfully propagates uncertainty from the activation space to the network's final output without altering the pre-trained network's predictions. As a result, GAPA-Variational preserves the base network's predictive accuracy while providing a more reliable and nuanced uncertainty estimate, making it well suited for risk-sensitive applications.

4. Related Work

In Morales-Alvarez et al. (2020), auNN replaces activations with GPs and trains them jointly across layers via variational inference, requiring multiple samples at inference time. In contrast, our method uses the original activation (e.g., ReLU) as the GP prior mean—thereby preserving the pre-trained network's predictions—and fits GPs solely to quantify the uncertainty of the activation function. This post-hoc approach avoids re-training the network and achieves uncertainty estimation with a single forward pass.

5. Conclusion

In this work, we have introduced the Gaussian Process Activation function (GAPA), a novel framework designed to quantify uncertainty in pre-trained neural networks. We have also presented a theoretically principled method to propagate uncertainty from the activations space to the output space using the delta approximation approach. Our approach empirically outperforms the Laplace approximation method, achieving faster training times. Nevertheless, Gaussian processes remain computationally expensive at inference time. Future work will focus on exploring scalable models or approximations to Gaussian processes to optimise computational efficiency, as well as extending the model to classification task.

References

Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. *Nature Machine Intelligence*, 1(1):20–23, 2019.

Txus Blasco, J Salvador Sánchez, and Vicente García. A survey on uncertainty quantification in deep learning for financial time series prediction. *Neurocomputing*, 576:127339, 2024.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 34:20089–20103, 2021.

et al. Dutordoir. Us flight delay dataset, 2020. Dataset available at https://...

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

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

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

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, pages 1861–1869. PMLR, 2015.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

Pavel Izmailov, Wesley J Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Subspace inference for bayesian deep learning. In *Uncertainty in Artificial Intelligence*, pages 1169–1179. PMLR, 2020.

Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are bayesian neural network posteriors really like? In *International conference on machine learning*, pages 4629–4640. PMLR, 2021.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1. Minneapolis, Minnesota, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Jingyao Li, Pengguang Chen, Zexin He, Shaozuo Yu, Shu Liu, and Jiaya Jia. Rethinking out-of-distribution (ood) detection: Masked image modeling is all you need. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11578–11589, 2023.

Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

Xixi Liu, Yaroslava Lochman, and Christopher Zach. Gen: Pushing the limits of softmax-based out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23946–23955, 2023.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, 32, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Pablo Morales-Alvarez, Daniel Hernández-Lobato, Rafael Molina, and José Miguel Hernández-Lobato. Activation-level uncertainty in deep neural networks. In *International Conference on Learning Representations*, 2020.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Luis A Ortega, Simón Rodríguez Santana, and Daniel Hernández-Lobato. Variational linearized laplace approximation for bayesian deep learning. *arXiv preprint arXiv:2302.12565*, 2023.

Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with bayesian principles. *Advances in neural information processing systems*, 32, 2019.

Hannes Salimbeni and Marc Peter Deisenroth. Deep gaussian processes for regression using expectation propagation. In *Advances in Neural Information Processing Systems*, 2017.

Sina Shafaei, Stefan Kugele, Mohd Hafeez Osman, and Alois Knoll. Uncertainty in machine learning: A safety perspective on autonomous driving. In *Computer Safety, Reliability, and Security: SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, Proceedings 37*, pages 458–464. Springer, 2018.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR, 2009.

Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020.

Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E Turner, Jose Miguel Hernandez-Lobato, and Alexander L Gaunt. Deterministic variational inference for robust bayesian neural networks. *arXiv preprint arXiv:1810.03958*, 2018.

Appendix A. Empirical Estimation of Inducing Inputs and RBF Kernel Hyperparameters

Inducing Input Selection: To set the RBF kernel hyperparameters in a data-driven manner, we first select inducing inputs for each neuron's GP based on the empirical cumulative distribution function (CDF) of its pre-activation values. Let x denote the one-dimensional pre-activation values for a given neuron, and assume these values are sorted as

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(N)}.$$

The empirical CDF is then given by

$$F(x_{(i)}) = \frac{i}{N}, \quad i = 1, \ldots, N.$$

To robustly capture the data distribution—especially the boundaries critical for out-of-distribution detection—we always include the minimum $x_{(1)}$ and maximum $x_{(N)}$ as inducing points. The remaining inducing inputs are selected by partitioning the CDF into equal quantile intervals. Specifically, if M inducing points are desired (with two reserved for the minimum and maximum), then the other $M-2$ inducing points correspond to quantile levels

$$p_m = \frac{m+1}{M-1}, \quad m = 1, 2, \ldots, M-2.$$

Each inducing input is chosen as the $x_{(i)}$ whose empirical CDF value is closest to the corresponding p_m.

RBF Kernel Hyperparameter Estimation: The RBF kernel is defined as

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right),$$

where:

- ℓ is the lengthscale, and
- σ_f^2 is the output scale (variance constant).

We estimate the lengthscale ℓ as a chosen quantile (e.g., the 25th percentile) of the pairwise Euclidean distances among the selected inducing inputs:

$$\ell = \text{quantile}\Big(\{|x_i - x_j| : i \neq j\},\, q\Big), \quad \text{with } q = 0.25.$$

The output scale is set based on the variance of the training outputs (activation function):

$$\sigma_f^2 = \max\Big(1,\, \text{Var}(y_{\text{train}})\Big).$$