Local Calibration Testing in Supervised Machine Learning Models Using Input Space Kernels

Markus Walker, **Marcel Reith-Braun**, and **Uwe D. Hanebeck**
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
markus.walker@kit.edu, marcel.reith-braun@kit.edu, uwe.hanebeck@kit.edu

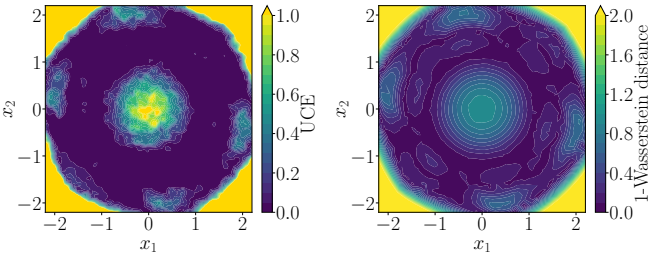
***Abstract*—Bayesian machine learning models—especially Bayesian neural networks (BNNs)—offer powerful black-box approaches for prediction and uncertainty quantification. However, these models frequently exhibit inconsistent prediction quality across input regions, and conventional global metrics (e.g., the mean squared error (MSE)) are inadequate for capturing such local discrepancies. To overcome this limitation, we introduce a novel kernel-based framework for local calibration testing that assesses how well predicted distributions reflect both the function to be learned and inherent uncertainties. In our approach, spherical input-space kernels are used to define relevant subsets in the neighborhood of a point to be tested. This enables the online assessment of these localized regions using calibration metrics or statistical tests. By aggregating results across multiple kernel widths, our method yields both robust binary decisions and a continuous analysis over arbitrary inputs. Numerical experiments on single- and multi-dimensional regression tasks demonstrate the efficiency and scalability of our approach, underscoring its potential for real-time and large-scale applications.**

***Index Terms*—Bayesian neural networks, uncertainty quantification, statistical testing, calibration testing.**

I. Introduction

In many safety-critical applications, effectively quantifying the uncertainty of predictive models is crucial for building trustworthy systems. In the realm of machine learning, Bayesian neural networks (BNNs) offer a powerful means of capturing uncertainty, with the promise that a model will say whether it is uncertain about its prediction, e.g., by increasing the variance of an output. In practice, however, it is observed that uncertainty estimates are of different quality depending on the region of the input space. E.g., a region densely covered by training data may yield *well-calibrated* predictions, whereas regions with sparse training data may produce inaccurate outputs. In this context, calibration refers to the consistency between the predicted uncertainty and the actual uncertainty inherent in the data-generating process, which is usually assessed using test samples. Therefore, identifying input space regions of suboptimal calibration is critical to ensuring trustworthy predictions, particularly when these predictions inform important decisions.

Despite numerous advances in approximate inference techniques, such as Markov Chain Monte Carlo (MCMC) [1], Variational Inference (VI) [2], or Expectation Propagation (EP) [3], challenges remain in evaluating a model's local calibration. Standard diagnostic metrics or calibration measures such as the mean squared error (MSE) or uncertainty calibration error (UCE) [4] typically focus on global evaluations and overlook input-dependent calibration discrepancies. Motivated by this gap, we propose a kernel-based testing framework that examines calibration in localized, input-dependent neighborhoods defined by spherical input-space kernels. By systematically combining results from multiple kernel widths, our method robustly reveals localized miscalibration. The relevance of our kernel-based approach stems from its flexibility and generality. Unlike our previous methods that rely on fixed global partitions [5]–[8] or are designed for specific input dimensionalities [5], [6], our proposed approach enables calibration assessment at arbitrary locations and scales in any input dimension. This adaptability is crucial for uncovering local miscalibration that may be missed by global or coarsely partitioned methods.



(a) local calibration (b) ground truth distance

Fig. 1. Local calibration using input space kernels and the UCE is shown in (a), providing insights into the model's calibration across the input space $[x_1 \ x_2]^\top \in \mathbb{R}^2$. Note that our method does not require knowledge of the exact data generation process. For comparison, (b) depicts the distance between the model's predictions and the data-generating process.

Contribution: In this paper, we first review the Bayesian learning setup and the resulting test problem in Secs. III and IV, respectively. Building upon our prior work on identifying trustworthy regions [5]–[8], we then introduce an improved kernel-based approach for local calibration testing via spherical input-space kernels, given only a test data set. In contrast to state-of-the-art methods, our approach does not require global partitioning of the input space. We further propose a mechanism to aggregate local tests across multiple kernel widths, yielding both binary calibration decisions and continuous calibration measures. Lastly, we provide experimental validation on single- and multi-dimensional regression tasks, demonstrating the efficiency, scalability, and alignment of our method with principled ground truth distances, as shown in Fig. 1.

This work is part of the German Research Foundation (DFG) AI Research Unit 5339 regarding the combination of physics-based simulation with AI-based methodologies for the fast maturation of manufacturing processes.

Notation: In this paper, underlined letters, e.g., $\underline{x}$, denote vectors, boldface letters, such as $\underline{\boldsymbol{x}}$, represent random variables, while sets are represented as calligraphic letters, e.g., $\mathcal{D}$.

II. Related Work

A. Approximate Inference

Unlike classical neural networks with deterministic weights, BNN weights, since they are represented by distributions, cannot be trained using standard backpropagation. Instead, learning weight distributions relies on approximate probabilistic inference. We exclusively consider the principles used in the evaluation (Sec. IX) in this subsection. For a comprehensive overview of these methods, see [9].

The MCMC method, initially proposed in [1], has become a widely used approach for probabilistic inference in training BNNs. Despite its effectiveness, its high computational cost—due to the need for generating a large number of samples, as exemplified by the Metropolis–Hastings algorithm [10]—remains a significant drawback. To enhance its efficiency, several improvements have been developed, including Gibbs sampling [11], Hamiltonian Monte Carlo [12], and the No-U-Turn Sampler (NUTS) [13].

Another class of methods, collectively known as Variational Inference (VI) [2], transforms the complex inference task into an optimization problem by minimizing the empirical lower bound of the reverse Kullback–Leibler divergence between the variational distribution and the true posterior. Several advancements have enhanced scalability for larger networks by using scaled gradients from random subsets of training data as implemented in Stochastic Variational Inference (SVI) [14] or by employing deterministic moment propagation [15].

Expectation Propagation (EP) [3] approximates the true posterior with a more tractable distribution. In contrast to VI, which minimizes the reverse Kullback–Leibler divergence, EP minimizes the forward divergence. This approach has gained considerable attention in BNNs, with prominent examples such as probabilistic backpropagation (PBP) [16].

B. Calibration Measures for Regression

Assessing prediction quality via calibration measures involves quantifying how closely predictive distributions align with the true data-generating process [17]. This evaluation is complicated by the limited number of test samples and the absence of ground truth uncertainty estimates. Various tools exist for this purpose, including calibration plots [18], which visually compare predicted confidence levels against empirical observations. For regression models, scoring rules are often employed to measure the quality of uncertainty estimates [17].

In the case of *univariate* and normally distributed predictions, calibration measures such as the uncertainty calibration error (UCE) [4] and expected normalized calibration error [19] are used. These metrics assess discrepancies between predicted variances and the MSE computed over binned test data $\mathcal{B}_s$. The UCE is defined by

$$\mathrm{UCE} = \sum_{s=1}^{S} \frac{|\mathcal{B}_s|}{N_{\mathrm{Test}}} |\mathrm{MSE}(\mathcal{B}_s) - \mathrm{MV}(\mathcal{B}_s)| \ , \tag{1}$$

where $\mathrm{MSE}(\mathcal{B}_s)$ represents the MSE between the predicted means and the observed outputs within the s-th bin, while $\mathrm{MV}(\mathcal{B}_s)$ is the mean variance of the predictions within the s-th bin, and S is the number of bins. $|\mathcal{B}_s|$ is the number of test data points within the s-th bin, whereas N_{Test} is the total number of test data points. For normally distributed predictions in arbitrary dimensions, [17] introduced the quantile calibration error, which compares observed frequencies with selected quantile values of chi-squared distributed errors.

C. Statistical Testing

Statistical testing is used to decide between competing hypotheses by using a test statistic T. In practice, one tests a null hypothesis H_0 against an alternative hypothesis H_1, rejecting H_0 if the data provide sufficient evidence for H_1. For a detailed treatment of statistical testing principles, see [20].

To assess whether data points are consistent with a predicted normal distribution, the chi-square test based on the average squared Mahalanobis distance—commonly known as the averaged normalized estimation error squared (ANEES) test [21]—can be used. The ANEES test statistic is given by

$$T_{\mathrm{ANEES}} = \frac{1}{N_{\mathrm{Test}}} \sum_{n=1}^{N_{\mathrm{Test}}} \left(\underline{y}_n - \underline{\mu}_n^{\underline{\boldsymbol{y}}}\right)^{\top} \left(\mathbf{C}_n^{\underline{\boldsymbol{y}}}\right)^{-1} \left(\underline{y}_n - \underline{\mu}_n^{\underline{\boldsymbol{y}}}\right) \ , \tag{2}$$

where $\underline{y}_n \in \mathbb{R}^{d_y}$ is the n-th output sample from the test data set and $\mathcal{N}(\underline{\mu}_n^{\underline{\boldsymbol{y}}}, \mathbf{C}_n^{\underline{\boldsymbol{y}}})$ is its corresponding normally distributed prediction. For normally distributed predictions, the average of the squared Mahalanobis distances follows a chi-squared distributed test statistic with $k = d_y \cdot N_{\mathrm{Test}}$ degrees of freedom. The p-value for the two-sided test is given by

$$p^{\mathrm{val}} = 2 \cdot \min\left(p_{\mathrm{l}}^{\mathrm{val}}, p_{\mathrm{u}}^{\mathrm{val}}\right) \ , \tag{3}$$

where $p_{\mathrm{l}}^{\mathrm{val}} = F_{\chi_k^2}(T_{\mathrm{ANEES}})$ and $p_{\mathrm{u}}^{\mathrm{val}} = 1 - F_{\chi_k^2}(T_{\mathrm{ANEES}})$ are the lower and upper tail probabilities from the chi-square cumulative density function (CDF), respectively. For a binary decision, this p-value is compared to a significance level α, and H_0 is rejected if $p^{\mathrm{val}} < \alpha$.

In cases where no distributional assumption can be made, nonparametric tests such as the Kolmogorov–Smirnov [22] or the Anderson–Darling test [23] may be employed. The binomial test [20] can also be used, for instance, to verify that the $95\,\%$ confidence interval of predictions covers $95\,\%$ of test outputs.

D. Trust Region Identification

In previous work aimed at assessing local calibration in Bayesian models [5], we proposed a two-phase testing methodology that involves partitioning the input space of regression models. The approach first partitions the input space *globally*, i.e., identifies candidate regions in the input space where test data are present—these are essentially segments of the input domain selected for further testing—and then assesses the calibration of predictions within these regions. In contrast to global calibration measures from Sec. II-B, which are computed over the entire test data set, the methodology introduced in [5] enables a localized evaluation of the model's performance per candidate region.

The first phase focuses on how regions are represented and identified. In [5], regions are defined as intervals, which

TABLE I
REGION REPRESENTATIONS OF TRUST REGION METHODS.

dimensionality	region representation	partioning
1	intervals [5]	global
2	Voronoi tessellation [6]	global
arbitrary	k-d trees partitions [7]	global
arbitrary	ball tree partitions [8]	global
arbitrary	**kernels (proposed method)**	**local**

works only for single-input systems. For multi-dimensional inputs, [6] employs Voronoi tessellations in two dimensions, while [7] and [8] extend the approach to arbitrary dimensions using k-d trees [24] and ball tree [25] partitions, respectively. A concise overview of the methodologies is presented in Tab. I.

In the second phase, candidate regions are tested for calibration using statistical tests such as the ANEES and the binomial test. Regions that are found to be untrustworthy based on these statistical tests are subsequently rejected. However, arbitrary calibration measures can be used to evaluate candidate regions. E.g., the expected calibration error [26] is employed for classification tasks in [7], while the UCE is utilized for regression tasks in [6], [8].

III. Supervised Learning in Bayesian Models

We consider a supervised learning setup where the training data $\mathcal{D} = \{(\underline{x}_n, \underline{y}_n)\}_{n=1}^{N}$ are drawn from the true data-generating process. Each pair consists of an input vector $\underline{x}_n \in \mathbb{R}^{d_x}$ and its corresponding output realization $\underline{y}_n \in \mathbb{R}^{d_y}$. Typically, the true mapping between inputs and noisy outputs is unknown, so the goal is to learn this mapping from the available data. To achieve this, the relationship between inputs and outputs is modeled by a feedforward BNN defined as $\underline{\boldsymbol{y}} = f(\underline{x}, \underline{\boldsymbol{w}})$, where $\underline{x}$ is the deterministic input, $\underline{\boldsymbol{w}}$ is the random weight vector containing all network weights, and $\underline{\boldsymbol{y}}$ is the random output vector. For architectures where weights are organized as per-layer matrices, the weight vector $\underline{\boldsymbol{w}}$ is obtained by flattening and concatenating these matrices into a single d_w-dimensional vector. The learning process is based on Bayes' rule for estimating the weight posterior

$$p(\underline{w} \mid \mathcal{D}) = \frac{p(\mathcal{Y} \mid \mathcal{X}, \underline{w})\, p(\underline{w})}{p(\mathcal{Y} \mid \mathcal{X})} \quad ,$$

where $\mathcal{X} = \{\underline{x}_1, \ldots, \underline{x}_N\}$ and $\mathcal{Y} = \{\underline{y}_1, \ldots, \underline{y}_N\}$ are the sets of input and output data of the training data set $\mathcal{D}$, $p(\underline{w})$ is the prior, $p(\mathcal{Y} \mid \mathcal{X}) = \int_{\Omega_{\underline{\boldsymbol{w}}}} p(\mathcal{Y} \mid \mathcal{X}, \underline{w})\, p(\underline{w})\, \mathrm{d}\underline{w}$ is the normalization constant, and $\Omega_{\underline{\boldsymbol{w}}} \subseteq \mathbb{R}^{d_w}$ is the sample space of the weights. The likelihood $p(\mathcal{Y} \mid \mathcal{X}, \underline{w})$ is defined by the model architecture and, by assuming independent output realizations $\underline{y}_n$, can be written as $p(\mathcal{Y} \mid \mathcal{X}, \underline{w}) = \prod_{n=1}^{N} p(\underline{y}_n \mid \underline{x}_n, \underline{w})$. For example, if there is a (possibly unknown) normal noise probability density function (PDF), the likelihood function can be modeled by $p(\underline{y}_n \mid \underline{x}_n, \underline{w}) = \mathcal{N}(f(\underline{x}_n, \underline{w}), \mathbf{C})$. Although $\underline{x}$ is deterministic, we condition on the input data in our notation to emphasize the dependence on the input data. The predictive distribution is then obtained by

$$p(\underline{y} \mid \underline{x}, \mathcal{D}) = \int_{\Omega_{\underline{\boldsymbol{w}}}} p(\underline{y} \mid \underline{x}, \underline{w})\, p(\underline{w} \mid \mathcal{D})\, \mathrm{d}\underline{w} \quad .$$

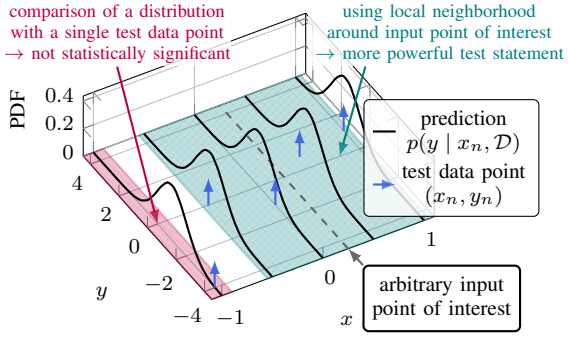


Fig. 2. Local test problem for a single-input, single-output system (adapted version from [8]). Within the purple-colored input space region, only one test point is used, which is not statistically significant. Adjacent predictions can be combined, as shown in the teal-colored region, to increase the effectiveness of the test statement, and allow arbitrary input points to be tested.

In general, exact inference for the weight posterior or prediction is intractable, hence, approximation methods (as discussed in Sec. II-A) are applied in practice.

Note that predictive distributions can also be obtained using heteroscedastic models with deterministic weights, where the noise variance is a function of the input data (i.e., $\sigma^2 = \sigma^2(\underline{x}_n)$), though this approach does not consider parameter uncertainties. Although not the focus of this paper, the proposed method can also be applied to such cases.

IV. Test Problem

Since the learning process relies on approximate inference, assumptions, and hyperparameters, the resulting model is inherently error-prone. Therefore, it is crucial to evaluate the quality of the learned model's predictive distributions. However, evaluating the quality of the learned model's predictive distributions is challenging since the true data-generating process is unknown. In particular, the true output distribution $p(\underline{y} \mid \underline{x})$ cannot be directly compared with the predictive distribution $p(\underline{y} \mid \underline{x}, \mathcal{D})$. Therefore, prediction quality is assessed using a test data set $\mathcal{D}_{\text{Test}} = \{(\underline{x}_n, \underline{y}_n)\}_{n=1}^{N_{\text{Test}}}$.

To assess the local quality of a single prediction, i.e., $p(\underline{y} \mid \underline{x}_n, \mathcal{D})$, for a specific input $\underline{x}_n$, given the corresponding output realization $\underline{y}_n$, a distance or calibration measure, such as the negative log-likelihood, can be employed. However, drawing definitive conclusions from a single sample is statistically unsound. E.g., consider a one-dimensional realization y_n that deviates by more than three standard deviations from its predicted mean, as highlighted in the purple region in Fig. 2. Although such an event is unlikely, its probability is not zero, which underscores the limitation of single-point assessments. In addition, traditional evaluation assumes that an input point of interest matches a known input in the test data set. In real-world scenarios, however, a model is often queried with *arbitrary input points* that do not coincide with any test sample.

To overcome these challenges, local neighborhoods can be defined around any given input point by aggregating the predictions from multiple nearby test points. This addresses several challenges: it increases statistical significance, provides

a representation of local prediction quality, and enables assessment at any point in the input space—not just at predefined test inputs, as illustrated by the teal region ▫ in Fig. 2.

Consequently, the test problem can be articulated in two questions: 1) How can such local neighborhoods be defined? 2) How can multiple predictions and test data points from these neighborhoods be integrated into a statistically meaningful test?

V. Key Idea

To address the test problem outlined in Sec. IV, in [5] we introduced the concept of candidate regions—fixed regions in input space that contain multiple test points and are tested using statistical tests or calibration measures. However, this requires a global partitioning of the input space.

The key idea of this paper is to extend candidate regions to an input kernel-based approach, where the quality of predictions is evaluated using multiple kernel widths centered around the input point of interest, to address the first test problem, as shown in Fig. 3. By utilizing multiple kernel widths, we can assess the quality of predictions comprehensively, as prediction quality may vary depending on the kernel width. Using this approach, trustworthy predictions should remain consistent across different kernel widths, while less trustworthy predictions may exhibit significant variation. Moreover, this approach enables the evaluation of arbitrary input points *without relying on predefined global partitions*, such as the interval-based candidate regions in [5] or hyperrectangles obtained from a k-d tree in [7]. As a result, our method can be used to assess *online* whether the input of a model leads to a trustworthy prediction by evaluating the quality in the kernel-defined neighborhood of a point of interest. However, in contrast to our previous methods based on global partitioning, at this point, we do not impose explicit requirements on the number of test samples within each kernel, but implicitly assume that enough data points are available to ensure statistical significance.

To address the second testing problem, we combine results from multiple kernel widths using two schemes to achieve a robust evaluation of prediction quality:

1) Utilize calibration measures for each kernel and aggregate their results with weighted averaging to obtain a *continuous measure* of local prediction quality.
2) Conduct statistical tests for each kernel and combine the resulting p-values to yield a *binary decision* on prediction trustworthiness.

Both schemes integrate the evaluation at a single input point (the kernel center) with the assessment of adjacent data and prediction, offering a comprehensive view of local prediction quality.

VI. Input Space Kernel

As illustrated in Fig. 3, a d_x-dimensional sphere is used as the input space kernel, centered at the point $\underline{x}_\mathrm{c}$ that is to be tested. The kernel is defined by

$$K(\underline{x}_\mathrm{c}, b) = \begin{cases} 1 & \text{if } \|\underline{x} - \underline{x}_\mathrm{c}\|_2 \leq \frac{b}{2} \\ 0 & \text{otherwise} \end{cases} ,$$

where $\|\underline{x} - \underline{x}_\mathrm{c}\|_2$ is the Euclidean distance between the point $\underline{x}$ and the kernel center $\underline{x}_\mathrm{c}$, and b is the nonnegative kernel width, i.e., the diameter of the kernel. Consequently, the set of indices of test points falling within the kernel is defined as

$$\mathcal{I}_K(\underline{x}_\mathrm{c}, b) = \left\{ n \middle| \|\underline{x}_n - \underline{x}_\mathrm{c}\|_2 \leq \frac{b}{2},\, n = 1, 2, \ldots, N_\mathrm{Test} \right\} ,$$

where $\underline{x}_n$ are test input points from the test data set $\mathcal{D}_\mathrm{Test}$. Therefore, the set of input points within the kernel is given by $\mathcal{X}_K(\underline{x}_\mathrm{c}, b) = \{\underline{x}_n\}_{n \in \mathcal{I}_K(\underline{x}_\mathrm{c}, b)}$, the set of corresponding output data points is $\mathcal{Y}_K(\underline{x}_\mathrm{c}, b) = \{\underline{y}_n\}_{n \in \mathcal{I}_K(\underline{x}_\mathrm{c}, b)}$, and the set of predictions is $\mathcal{P}_K(\underline{x}_\mathrm{c}, b) = \{p(\underline{y} \mid \underline{x}_n, \mathcal{D})\}_{n \in \mathcal{I}_K(\underline{x}_\mathrm{c}, b)}$. For brevity, we refer to these sets simply as $\mathcal{I}_K$, $\mathcal{X}_K$, $\mathcal{Y}_K$, and $\mathcal{P}_K$ throughout the paper. To determine the input points within a kernel, we need to find all points within a certain radius of the kernel center point. This can be done numerically, e.g., by a naïve search where the distance from the kernel center point to all test points is calculated, or more efficiently by using data structures such as k-d trees [24].

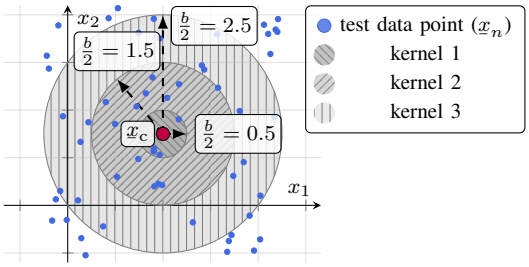


Fig. 3. Illustration of spherical input space kernels in a 2-dimensional input space using three different kernel widths. All kernels are centered around the same point ($\underline{x}_\mathrm{c}$), i.e., the point in the input space that should be tested.

VII. Kernel Statistic

Given the input space kernel and the corresponding output points $\mathcal{Y}_K$ and predictions $\mathcal{P}_K$, evaluation is performed by applying an arbitrary statistical test, distance metric, or calibration measure, resulting in a kernel statistic $T_K(\underline{x}_\mathrm{c}, b)$.

A. Example: UCE as Kernel Calibration Measure

As an example, consider the case of *univariate* and normally distributed predictions $p(y \mid x_n, \mathcal{D}) = \mathcal{N}(\mu_n^{\boldsymbol{y}}, (\sigma_n^{\boldsymbol{y}})^2)$ for which we choose the UCE (1) (without binning scheme) as a calibration measure. Given the set of predictions $\mathcal{P}_K$, the kernel statistic $T_{K,\mathrm{UCE}}(\underline{x}_\mathrm{c}, b)$ is then defined as

$$T_{K,\mathrm{UCE}}(\underline{x}_\mathrm{c}, b) = \left| \mathrm{MSE}(\mathcal{Y}_K, \mathcal{P}_K) - \mathrm{MV}(\mathcal{P}_K) \right| ,$$
$$\mathrm{MSE}(\mathcal{Y}_K, \mathcal{P}_K) = \frac{1}{|\mathcal{I}_K|} \sum_{n \in \mathcal{I}_K} (y_n - \mu_n^{\boldsymbol{y}})^2 ,$$
$$\mathrm{MV}(\mathcal{P}_K) = \frac{1}{|\mathcal{I}_K|} \sum_{n \in \mathcal{I}_K} (\sigma_n^{\boldsymbol{y}})^2 ,$$

where $|\mathcal{I}_K|$ is the number of points within the kernel.

B. Kernel Statistical Test

Analogously, a statistical test, e.g., the ANEES test, can be used for each kernel to obtain the kernel statistic $T_{K,\mathrm{ANEES}}(\underline{x}_\mathrm{c}, b)$ based on the test statistic (2). Additionally, each statistical test returns a p-value $p^\mathrm{val}(\underline{x}_\mathrm{c}, b)$ for each kernel, enabling kernel-level decision-making. In case of the ANEES test, the p-value is calculated according to (3).

C. Trade-off Between Local and Global Assessment

Observing the kernel statistics reveals that for small kernel widths, the local behavior is captured, while for larger widths the global behavior emerges. In fact, as $b \to \infty$ (i.e., when the kernel encompasses all test points), the kernel statistic equals the global statistic. This is evident, e.g., in the ANEES test plot in Fig. 6. Above a certain kernel width, the ANEES converges to the global ANEES value, as the kernel includes all input points. Thus, analyzing kernel statistics over multiple widths provides insights into both the *local* and *global* quality.

VIII. Combined Statistic

We now introduce two approaches—continuous and binary assessment—to combine kernel statistics for evaluating local prediction quality. For both approaches, test results are aggregated over L kernels (with widths b_l, for $l = 1, \ldots, L$) centered at the same input point, with each kernel assigned a nonnegative normalized weight w_l reflecting its contribution.

A. Continuous Assessment

For continuous assessment, we define the combined statistic as the weighted sum of the kernel statistics

$$T(\underline{x}_\mathrm{c}) = \sum_{l=1}^{L} w_l \cdot T_K(\underline{x}_\mathrm{c}, b_l) \ ,$$

where $T_K(\underline{x}_\mathrm{c}, b_l)$ is the kernel statistic for a single kernel.

B. Binary Assessment

For binary assessment, a statistical test is performed for each kernel width, yielding a p-value $p_l^{\mathrm{val}}(\underline{x}_\mathrm{c}, b_l)$ per kernel. These p-values must then be combined to obtain a single p-value that summarizes the evidence against the null hypothesis at the given center point. Standard methods for combining p-values, such as Fisher's method [27] assume independence among the p-values. In our case, however, kernels centered at the same input point share common test points (e.g., in Fig. 3, the test points in kernel 1 also appear in kernels 2 and 3), resulting in correlated p-values. To address this issue, we adopt the Cauchy combination test [28], which is robust to correlations and avoids explicitly estimating correlations between p-values. The combined statistic is defined as [28]

$$T_\mathrm{C}(\underline{x}_\mathrm{c}) = \sum_{l=1}^{L} w_l \cdot \tan\left(\pi\left(\frac{1}{2} - p_l^{\mathrm{val}}(\underline{x}_\mathrm{c}, b_l)\right)\right) \ .$$

Under the null hypothesis of the test (when p-values are uniformly distributed), each term $\tan\big(\pi\big(\frac{1}{2} - p_l\big)\big)$ follows a standard Cauchy distribution, and hence their weighted sum T_C is approximately standard Cauchy distributed. Due to the heavy-tailed nature of the Cauchy distribution, the combined statistic is robust to the correlation between the p-values [29], and the p-value of the combined statistic can be approximated using the upper tail probability of the standard Cauchy CDF which is given by [28]

$$p_\mathrm{C}^{\mathrm{val}}(\underline{x}_\mathrm{c}) = 1 - F_\mathrm{C}(T_\mathrm{C}) = \frac{1}{2} - \frac{1}{\pi}\arctan(T_\mathrm{C}) \ .$$

A binary decision is reached by comparing $p_\mathrm{C}^{\mathrm{val}}$ to a predefined significance level α. E.g., if $p_\mathrm{C}^{\mathrm{val}} < \alpha$, the null hypothesis is rejected, and the predictions at the input point $\underline{x}_\mathrm{c}$ are deemed locally inaccurate or untrustworthy. The simplicity of this approach, which avoids the explicit computation of correlations, makes the Cauchy combination test particularly suitable for large-scale applications with many kernels and data points.

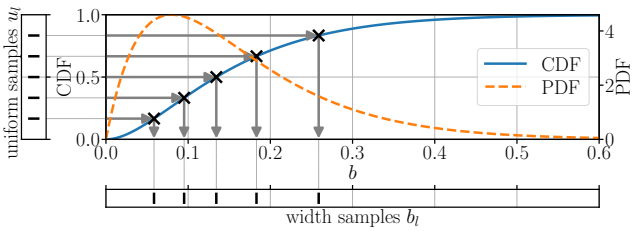


Fig. 4. Example of inverse sampling (4), with $\alpha_{\mathrm{Gam}} = 2$ and $\beta_{\mathrm{Gam}} = 12.5$.

C. Kernel Weight and Width Selection

Selecting appropriate kernel weights and widths is a crucial aspect of our kernel-based method, as it directly influences the balance between local and global quality assessments. A kernel that is too narrow may yield unreliable statistics due to insufficient data, while an overly broad kernel may obscure important local variations. For weighting, one may choose a function that emphasizes local behavior. E.g., unnormalized weights $\tilde{w}_l = b_l^{-d_x+1}$ has been used in localized cumulative distribution-based methods [30]. However, such weights must be balanced with the observation that very small kernel widths contain fewer test points, reducing statistical reliability.

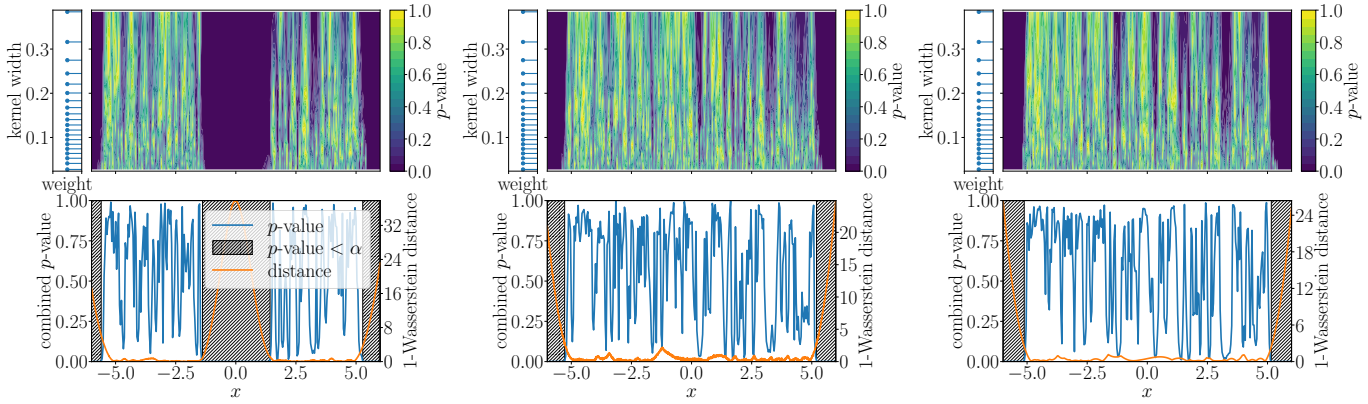
From a practical perspective, domain knowledge can guide the selection of kernel widths. To automate this process, we propose a heuristic for sampling kernel widths from a PDF that reflects domain knowledge. In our experiments, we use the gamma distribution, $\mathrm{Gam}(\alpha_{\mathrm{Gam}}, \beta_{\mathrm{Gam}})$, where the shape parameter α_{Gam} and inverse scale β_{Gam} are user-defined hyperparameters. The gamma distribution is well suited because it is defined over the nonnegative reals, matching the requirement for kernel widths. Deterministic kernel widths can be generated via *inverse transform sampling*

$$b_l = F_{\mathrm{Gam}}^{-1}(u_l) \ , \tag{4}$$

where F_{Gam}^{-1} is the inverse CDF of the gamma distribution and u_l are uniformly distributed values in $[0, 1]$ (e.g., equidistant points). This approach ensures that kernel widths are densely sampled in regions where the gamma PDF is high. The resulting weights are then equally weighted by $w_l = \frac{1}{L}$. An example of deterministic inverse sampling from a gamma distribution is shown in Fig. 4. Note that by using this scheme, only the number of kernel widths and the width distribution have to be specified and an arbitrary number of deterministic kernel widths can be generated.

D. Boundary Effects

In kernel-based methods with finite samples, boundary effects are a well-known challenge that can lead to biased results and can be addressed by using methods such as boundary correction [31]. However, how to properly address these boundary effects in the combined statistic remains an open question for future work.



(a) NUTS (b) SVI (c) PBP

Fig. 5. Local calibraion results for the single input cubic regression scenario S1. The p-values per kernel width are shown for (a) NUTS, (b) SVI, and (c) PBP. The combined p-values, the resulting binary decision, and the 1-Wasserstein distances (ground truth distances) are shown in the lower row of the figures.

IX. Numerical Evaluation

We evaluate our methods on two simulated single-output regression scenarios for which the data-generating processes are known. To evaluate the performance of local calibration methods, an arbitrary distance for PDFs between the predictive distributions and the true data-generating process can be used as the ground truth distance. For this purpose, we use the 1-Wasserstein distance in all experiments. The 1-Wasserstein distance between the normally distributed outputs of the true process, $p(y \mid x_n) = \mathcal{N}\big(\mu_{\mathrm{GT}}, \sigma^2_{\mathrm{GT}}\big)$ and the predictions $p(y \mid x_n, \mathcal{D}) = \mathcal{N}\big(\mu_{\mathrm{Pred}}, \sigma^2_{\mathrm{Pred}}\big)$ is given by [32], [33]

$$W_1 = |\mu|\left(1 - 2F_{\mathcal{N}}\left(-\frac{|\mu|}{|\sigma|}\right)\right) + |\sigma|\sqrt{\frac{2}{\pi}}\exp\left(-\frac{\mu^2}{2\sigma^2}\right) \ ,$$

where $\mu = \mu_{\mathrm{GT}} - \mu_{\mathrm{Pred}}$, $\sigma^2 = (\sigma_{\mathrm{GT}} - \sigma_{\mathrm{Pred}})^2$, and $F_{\mathcal{N}}(\cdot)$ is the standard normal CDF. To assess consistency with the ground truth distance, we examine whether high W_1 values correspond to regions rejected by our binary assessment, and whether our continuous measure reflects the behavior of W_1.

A. Single Input Regression Scenario S1

In our first evaluation scenario (S1), we use the same single-input cubic regression example as in [5]. In this scenario, 2000 training points and 2400 test points generated from $\boldsymbol{y} = x^3 + \boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 9)$ are used. Training inputs $x_n \in \mathcal{X}_{\mathrm{Train}}$ are drawn uniformly from $[-5, 5]$ and test inputs $x_n \in \mathcal{X}_{\mathrm{Test}}$ from $[-6, 6]$. From the training data, 30 % of the data around the origin is removed, producing a gap in the input training data approximately in the range $[-1.5, 1.5]$. We train BNNs using NUTS [13], SVI [14], and PBP [16] following the settings of [5]. For evaluation, 300 kernel centers $(\underline{x}_{\mathrm{c}})$ are equally spaced over $[-6, 6]$.

Fig. 6 shows the kernel-based ANEES for the NUTS predictions evaluated over kernel widths uniformly drawn from $[0, 25]$. As expected, for kernel widths $b \geq 24$ the statistic becomes constant across x values (since the kernel then encompasses the full test range, $x \in [-6, 6]$). In contrast, lower kernel widths ($b \ll 5$) capture interesting local characteristics.

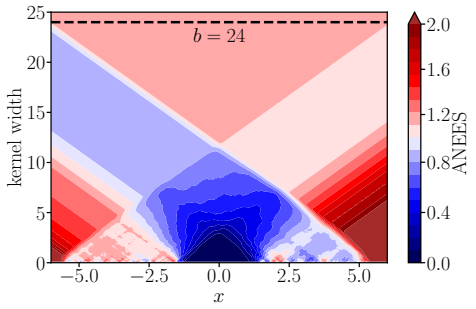


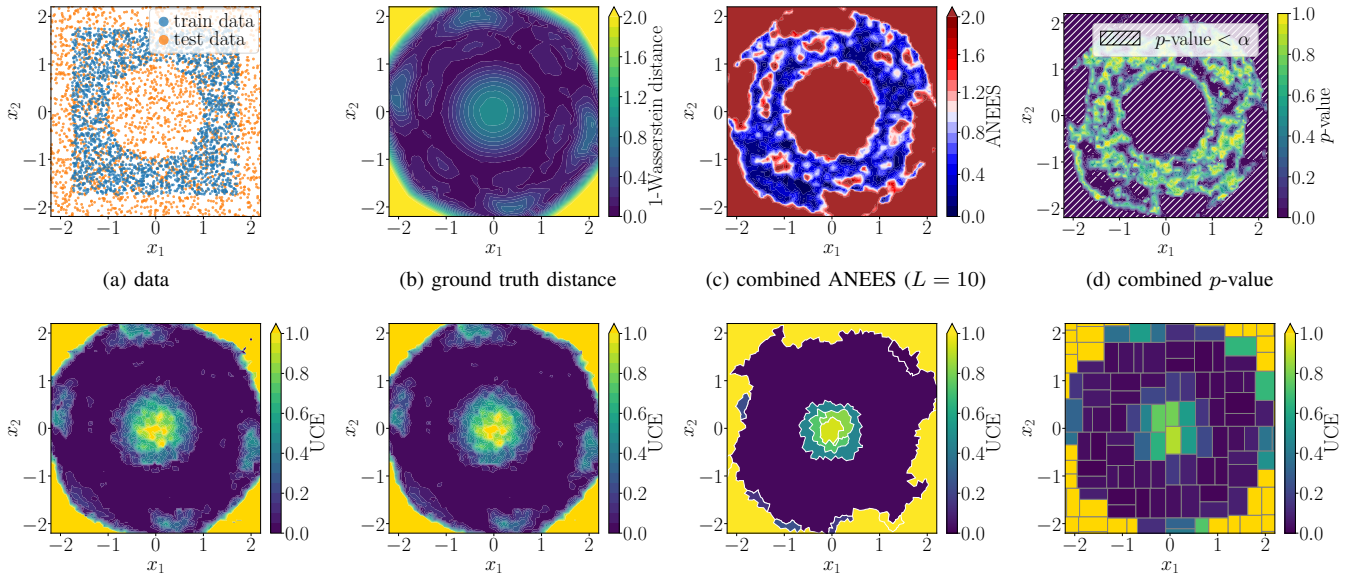
Fig. 6. Kernel statistics using the ANEES evaluated over multiple center points and kernel widths for the predictions of the NUTS for scenario S1.

For combined statistical tests with binary decisions, we deterministically sample $L = 20$ kernel widths b_l using inverse transform sampling method from $\mathrm{Gam}(\alpha_{\mathrm{Gam}} = 2, \beta_{\mathrm{Gam}} = 12.5)$. The results of the binary decision using the combined statistical test are shown in Fig. 5. It can be seen that the regions with significant errors regarding the significance level $\alpha = 0.01$ are rejected by the statistical combination test. E.g., the regions near the upper and lower bounds of the input data are rejected for all considered training methods, which aligns with the 1-Wasserstein distances in these regions.

Note that, as indicated by the 1-Wasserstein distances, the SVI and PBP predictions in Figs. 5b and 5c remain calibrated within $x \in [-1.5, 1.5]$, even though no training data are available in this region. This underscores the importance of local testing, as neither the presence nor absence of training data inherently guarantees calibration or miscalibration, respectively.

B. Multiple Input Regression Scenario S2

In this scenario, we evaluate a two-dimensional nonlinear regression task as introduced in [6]. The data is generated according to $\boldsymbol{y} = \sin\big(x_1^2 + x_2^2\big) + \boldsymbol{\epsilon}$ with 4500 training points and 3000 test points using $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 0.1)$. Training inputs are drawn uniformly from $[-1.75, 1.75]^2$, while test inputs are drawn uniformly from $[-2.5, 2.5]^2$. Around the origin, 30 % of the training data was removed, as shown in Fig. 7a. The BNN is



(a) data (b) ground truth distance (c) combined ANEES ($L = 10$) (d) combined p-value

(e) combined UCE ($L = 3$) (f) combined UCE ($L = 5$) (g) Voronoi-based UCE [6] (h) k-d tree-based UCE [7]

Fig. 7. Results of the multiple input scenario S2. In (a) the training and test data is shown. In (b) the 1-Wasserstein distance between the predictive distribution and the true data generating process, which is used as ground truth distance, is shown. In (c) and (d) the combined ANEES and p-value is shown. In (e) and (f) the UCE is shown for $L = 3$ and $L = 5$ kernel widths, respectively. For comparison, (h) and (g) display the UCE obtained using our methods from [6], [7].

trained using SVI [14] following the settings of [6]. The kernel center points $\underline{x}_c$, where the predictive distributions are evaluated, are placed on an equally spaced grid over $[-2.2, 2.2]^2$. Note that no evaluation points are placed at the boundary of the test input space to avoid potential boundary effects. The kernel widths are deterministically sampled using inverse transform sampling from $\mathrm{Gam}(\alpha_{\mathrm{Gam}} = 2, \beta_{\mathrm{Gam}} = 12.5)$.

The binary assessment (Fig. 7d) shows that regions with significant errors—where the combined p-value falls below the significance level $\alpha = 0.01$—are rejected by the test. In most cases, these regions align with areas of high 1-Wasserstein distance (see Fig. 7b). However, a few regions with low 1-Wasserstein distance are also rejected, which is expected given that the significance level limits false rejections to $100 \cdot \alpha\,\%$, and no absolute guarantees can be given.

The results using the kernel-based UCE are shown in Figs. 7e and 7f, and Fig. 1a for 3, 5 and 100 deterministically sampled kernel widths, respectively. These results align clearly with the 1-Wasserstein distance. Notably, the gap in training data around the origin is detected by the UCE, as predictions in that region are error-prone. Furthermore, the similarity of results across different numbers of kernel width samples confirms the robustness of the method. Although using a larger number of kernels leads to smoother results, even a small number of kernel widths captures the essential local error characteristics.

In Fig. 7h and Fig. 7g, the results obtained from our previous methods [6], [7] are shown. Both methods are capable of evaluating local calibration. However, our proposed kernel-based method provides finer-grained results in terms of locality and smoother calibration results, demonstrating greater flexibility compared to predefined k-d tree-based or Voronoi-based regions.

TABLE II
EVALUATION TIMES FOR A SINGLE KERNEL CENTER POINT.

scenario	statistic	L	run time[a]
S1	ANEES	20	$(24.82 \pm 5.43)\,\mu s$
S2	ANEES	10	$(41.36 \pm 4.75)\,\mu s$
S2	UCE	3	$(268.14 \pm 51.10)\,\mu s$
S2	UCE	5	$(321.91 \pm 72.36)\,\mu s$
S2	UCE	100	$3.40\,ms \pm 514.37\,\mu s$

[a]mean $\pm$ standard deviation

C. Implementation Details and Performance

All evaluations were conducted on a single CPU core of an Intel Core i7-1165G7 using a vectorized implementation in Python. As shown in Tab. II, our approach assesses input points of interest quickly, achieving sub-millisecond evaluation times in most settings. Notably, the computational cost for the kernel-based ANEES and its p-value is lower than that for the UCE, since our implementation reuses previously computed squared Mahalanobis distances for the ANEES. These results demonstrate that our method is efficient and promising for real-time as well as large-scale applications.

X. Discussion

Combining results from multiple kernel widths can be interpreted as a generalization of traditional calibration measures, as it enables a tunable trade-off between local and global evaluation of predictive quality. By selecting appropriate kernel widths in arbitrary-dimensional input spaces with hyperspherical kernels, our method captures local miscalibration while preserving global behavior. Notably, it is implicitly assumed that there are sufficient data points within the kernels to ensure the test results are statistically significant. In contrast, global partitioning-based

methods (e.g., k-d tree-based [7]) explicitly ensure that a minimum number of test points are present within a region. However, as our results show, this effect is partially compensated for by using a sufficiently large maximum kernel width, which indirectly ensures that numerous test points are taken into account.

The proposed framework provides both binary decisions and continuous measures that offer finer-grained insights at specific input points. It is important to note that statistical tests cannot offer absolute guarantees. The significance level controls—but does not eliminate—the probability of a Type I error.

In summary, our approach yields point-specific estimates of uncertainty calibration without relying on exact ground truth knowledge. Moreover, by utilizing inverse transform sampling for the deterministic generation of kernel widths, our method achieves efficient computation, with evaluation rates exceeding $1000\,\mathrm{Hz}$. This efficiency makes the approach attractive for real-time and large-scale applications.

XI. Conclusion

In conclusion, this paper presents a novel method for assessing the local quality of uncertainty predictions, i.e., the local calibration, in Bayesian models such as BNNs. Our approach leverages spherical kernels to represent localized input-space regions, then employs calibration metrics or statistical tests on predictions within each kernel. A key advantage is the method's ability to output either a clear binary decision or a continuous measure, indicating how well the predictive uncertainties align with the underlying data distribution at a given input point.

References

[1] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, Jun. 1953.

[2] A. Graves, "Practical variational inference for neural networks," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, vol. 24, 2011, pp. 2348–2356.

[3] T. P. Minka, "A family of algorithms for approximate Bayesian inference," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[4] M.-H. Laves, S. Ihler, J. F. Fast, L. A. Kahrs, and T. Ortmaier, "Well-calibrated regression uncertainty in medical imaging with deep learning," in *Proceedings of the Third Conference on Medical Imaging with Deep Learning*, Jul. 2020, pp. 393–412.

[5] M. Walker, M. Reith-Braun, P. Schichtel, M. Knaak, and U. D. Hanebeck, "Identifying trust regions of Bayesian neural networks," in *Proceedings of the 2023 IEEE Symposium Sensor Data Fusion and International Conference on Multisensor Fusion and Integration (SDF-MFI)*, Bonn, Germany, Nov. 2023, pp. 1–8.

[6] M. Walker, P. S. Bien, and U. D. Hanebeck, "Voronoi trust regions for local calibration testing in supervised machine learning models," in *Proceedings of the 2024 IEEE Symposium Sensor Data Fusion: Trends, solutions, applications (SDF)*, Bonn, Germany, Nov. 2024, pp. 1–8.

[7] M. Walker, H. Amirkhanian, M. F. Huber, and U. D. Hanebeck, "Trustworthy Bayesian perceptrons," in *Proceedings of the 2024 27th International Conference on Information Fusion (FUSION)*, Venice, Italy, Jul. 2024, pp. 1–8.

[8] M. Walker and U. D. Hanebeck, "Multi-scale uncertainty calibration testing for Bayesian neural networks using ball trees," in *Proceedings of the 2024 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Plzeň, Czech Republic, Sep. 2024, pp. 1–7.

[9] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, "Hands-on Bayesian neural networks—a tutorial for deep learning users," *IEEE Computational Intelligence Magazine*, vol. 17, no. 2, pp. 29–48, May 2022.

[10] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, Apr. 1970.

[11] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[12] R. M. Neal, *Bayesian learning for neural networks*, ser. Lecture Notes in Statistics. New York, NY: Springer, 1996, vol. 118.

[13] M. D. Homan and A. Gelman, "The No-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1593–1623, Jan. 2014.

[14] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[15] A. Wu, S. Nowozin, E. Meeds, R. E. Turner, J. M. Hernández-Lobato, and A. L. Gaunt, "Deterministic variational inference for robust Bayesian neural networks," in *International Conference on Learning Representations*, 2019.

[16] J. M. Hernández-Lobato and R. P. Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks," in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, Jul. 2015, pp. 1861–1869.

[17] F. Küppers, J. Schneider, and A. Haselhoff, "Parametric and multivariate uncertainty calibration for regression and object detection," in *Computer Vision – ECCV 2022 Workshops*, 2023, vol. 13805, pp. 426–442.

[18] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 2796–2804.

[19] D. Levi, L. Gispan, N. Giladi, and E. Fetaya, "Evaluating and calibrating uncertainty prediction in regression tasks," *Sensors*, vol. 22, no. 15, p. 5540, 2022.

[20] E. Lehmann and J. P. Romano, *Testing statistical hypotheses*, ser. Springer Texts in Statistics. Cham: Springer International Publishing, 2022.

[21] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. New York, USA: John Wiley & Sons, Inc., 2001.

[22] N. Smirnov, "Table for estimating the goodness of fit of empirical distributions," *The Annals of Mathematical Statistics*, vol. 19, no. 2, pp. 279–281, Jun. 1948.

[23] T. W. Anderson and D. A. Darling, "Asymptotic theory of certain "Goodness of Fit" criteria based on stochastic processes," *The Annals of Mathematical Statistics*, vol. 23, no. 2, pp. 193–212, Jun. 1952.

[24] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software*, vol. 3, no. 3, pp. 209–226, Sep. 1977.

[25] S. M. Omohundro, "Five balltree construction algorithms," International Computer Science Institute, Tech. Rep. 89-063, Dec. 1989.

[26] M. Pakdaman Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using Bayesian binning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, Feb. 2015.

[27] R. A. Fisher, *Statistical methods for research workers*, 2nd ed. Edinburgh, London, Oliver and Boyd, 1928.

[28] Y. Liu and J. Xie, "Cauchy combination test: a powerful test with analytic p-value calculation under arbitrary dependency structures," *Journal of the American Statistical Association*, vol. 115, no. 529, pp. 393–402, Jan. 2020.

[29] H. Zhang and Z. Wu, "The generalized Fisher's combination and accurate p-value calculation under dependence," *Biometrics*, vol. 79, no. 2, pp. 1159–1172, 2023.

[30] U. D. Hanebeck, M. F. Huber, and V. Klumpp, "Dirac mixture approximation of multivariate Gaussian densities," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Dec. 2009, pp. 3851–3858.

[31] M. C. Jones, "Simple boundary correction for kernel density estimation," *Statistics and Computing*, vol. 3, no. 3, pp. 135–146, Sep. 1993.

[32] M. Tsagris, C. Beneki, and H. Hassani, "On the folded normal distribution," *Mathematics*, vol. 2, no. 1, pp. 12–28, Mar. 2014.

[33] S. Chhachhi and F. Teng, "On the 1-Wasserstein distance between location-scale distributions and the effect of differential privacy," *arXiv:2304.14869*, 2023.