Local Modified Cramér–von Mises Distance for Uncertainty Calibration Assessment in Regression

Markus Walker*, **Simon Klaus***, **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, simon.klaus@student.kit.edu, marcel.reith-braun@kit.edu, uwe.hanebeck@kit.edu

***Abstract*—Uncertainty quantification for regression models is crucial for reliable decision-making in safety-critical applications, yet global calibration metrics such as the mean squared error (MSE) overlook local differences in calibration quality in different input-space regions. Existing local calibration assessment methods rely on heuristic data structure choices to partition the input space before evaluating calibration in the output space within each identified partition. We introduce a principled local distance measure that operates directly in the joint input-output space by representing test data as a Dirac mixture probability density function (PDF) and model predictions as a hybrid PDF, enabling the application of the localized cumulative distribution (LCD) integral transform and the modified Cramér–von Mises (mCvM) distance. This formulation avoids heuristic partitioning choices, as locality emerges naturally from kernel-based bandwidth integration in both input and output dimensions. Synthetic regression experiments demonstrate that our method reveals local miscalibration in Bayesian neural network (BNN) predictions with consistent behavior across different inference methods.**

***Index Terms*—Bayesian neural networks, uncertainty quantification, calibration testing, localized cumulative distributions.**

I. Introduction

In many safety-critical applications, effectively quantifying the uncertainty of predictive models is crucial for building trustworthy systems. Uncertainty-aware regression models such as BNNs offer a powerful means of capturing predictive uncertainty, with the promise that a model will indicate when it is uncertain about its predictions, e.g., by increasing the variance of the output distribution.

In practice, however, the quality of uncertainty estimates varies across the input space. Despite advances in approximate inference techniques such as Markov chain Monte Carlo [1], variational inference [2], or expectation propagation [3], a model may yield well-calibrated predictions in some regions while producing poorly calibrated outputs in others. For instance, regions densely covered by training data often yield well-calibrated predictions, whereas regions with sparse or absent training data may produce over- or underconfident estimates. In this context, *calibration* refers to the consistency between the predicted uncertainty and the actual uncertainty inherent in the data-generating process, typically assessed using test samples. Identifying input-space regions of suboptimal calibration is therefore critical to ensuring trustworthy predictions, particularly when these predictions inform safety-critical decisions. However, assessing local calibration remains challenging. Architecture and likelihood choices induce structural assumptions, and approximate inference can lead to local miscalibration even when global metrics appear satisfactory. Standard calibration measures such as the MSE or uncertainty calibration error (UCE) [4] typically focus on global evaluations and overlook input-dependent calibration discrepancies.

To address this limitation, our previous work [5]–[9] proposed local calibration assessment methods that evaluate calibration at arbitrary input locations. These approaches rely on specific data structures for organizing or weighting test samples, such as intervals [5], Voronoi tessellations [6], kd-trees [7], ball trees [8], or spherical input-space kernels [9]. After grouping test data according to these structures, a calibration measure is applied within each region or neighborhood, such as the averaged normalized estimation error squared, UCE [4], or its multivariate extension [10]. While these methods enable local assessment, they follow a two-step procedure: first, defining local regions in the input space using a chosen data structure, then evaluating calibration in the output space within those regions. This approach works well in practice, but the choice of data structure for defining locality remains heuristic, as different structures can yield different assessments for the same model.

In this paper, we take a fundamentally different approach by deriving a distance measure directly in the joint input-output space using the LCD [11] integral transform, as shown in Fig. 1. Our key insight is to represent test data as a Dirac mixture and model predictions as a hybrid PDF, both in the joint input-output space, where the hybrid PDF has continuous output components anchored at discrete input locations. This formulation allows us to apply the LCD transform. By integrating over kernel bandwidths in both input and output dimensions, we obtain a principled distance measure $D(\underline{m}_x)$ that naturally captures locality through the kernel-based integral transform itself.

Our contributions are threefold. First, we formulate the local calibration testing problem in terms of hybrid densities and derive the LCD transforms for both test data (represented by a Dirac mixture) and model predictions (represented by a hybrid PDF), enabling principled distance computation in joint input-output space. Second, we develop the local

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.

*These authors contributed equally.

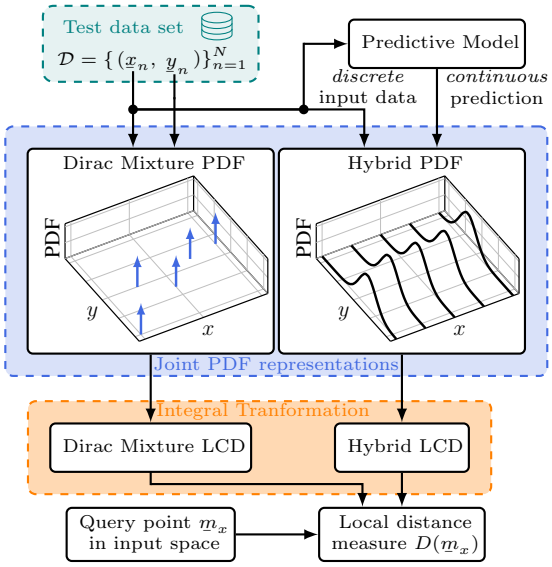


Fig. 1: Overview of the proposed method, in which test data and predictions are represented as Dirac mixture and hybrid PDF, respectively, in the joint input-output space. Their LCD transformations are then compared using a local distance measure for a given input query point $\underline{m}_x$.

distance measure with explicit bandwidth integration and analytical/semi-analytical solutions, including a penalty term to handle kernel boundary effects. Third, we demonstrate on synthetic regression examples that our method enables local calibration assessment *without heuristic spatial partitioning* and reveals local miscalibration patterns in BNN predictions.

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., $\mathcal{D}$. Expected values are denoted by $\hat{\cdot}$, and covariance matrices by $\mathbf{C}$. In case of scalar variance, we use σ^2.

II. Problem Statement

Given a trained model and test data, our goal is to develop a local distance measure $D(\underline{m}_x)$ that quantifies the discrepancy between predicted and observed distributions in the neighborhood of a query point $\underline{m}_x$. For this, we aim to derive a distance measure that operates directly on the combined input-output space, where locality emerges naturally from the distance computation itself.

Consider a regression model that quantifies uncertainty, e.g., a BNN, characterized by a predictive distribution $f(\underline{y} \mid \underline{x})$, where $\underline{x} \in \mathbb{R}^{d_x}$ is the input and $\underline{y} \in \mathbb{R}^{d_y}$ is the output. We evaluate the predictive distributions on a test data set $\{(\underline{x}_n, \underline{y}_n)\}_{n=1}^N$ with N test data points, where $\underline{x}_n$ is the n-th input and $\underline{y}_n$ is the corresponding observed output.

The test data represent empirical samples from an unknown true joint input-output distribution $f_{\text{true}}(\underline{x}, \underline{y})$. To formulate a distance measure in the joint input-output space, we view the test data as Dirac mixture approximation of the true joint PDF,

$$f_{\text{DM}}(\underline{x}, \underline{y}) = \sum_{n=1}^{N} w_n \delta(\underline{x} - \underline{x}_n)\delta(\underline{y} - \underline{y}_n) \ , \tag{1}$$

where δ denotes the Dirac delta distribution and w_n are weights, typically chosen as uniform $w_n = 1/N$. Using the continuous predictive distribution $f(\underline{y} \mid \underline{x})$, we can construct the model's joint PDF $f_{\text{H}}(\underline{x}, \underline{y}) = f(\underline{y} \mid \underline{x}) f(\underline{x})$, by approximating the test input PDF as $f(\underline{x}) = \sum_{n=1}^{N} w_n \delta(\underline{x} - \underline{x}_n)$. Therefore, the model's joint becomes a hybrid PDF

$$f_{\text{H}}(\underline{x}, \underline{y}) = \sum_{n=1}^{N} w_n \delta(\underline{x} - \underline{x}_n) f(\underline{y} \mid \underline{x}_n) \ , \tag{2}$$

which is discrete in the input space (point masses at test locations) but continuous in the output space, e.g., a Gaussian $\mathcal{N}(\underline{y}; \hat{\underline{y}}(\underline{x}_n), \mathbf{C}(\underline{x}_n))$ with mean $\hat{\underline{y}}(\underline{x}_n)$ and covariance $\mathbf{C}(\underline{x}_n)$.

The task of assessing local calibration now translates to deriving a principled distance measure between these two representations that naturally captures locality around arbitrary query points $\underline{m}_x$ in the input space, without imposing heuristic data structures for defining neighborhoods. To address this, we leverage the localized cumulative distribution (LCD), a multivariate generalization [11] of the univariate cumulative density function (CDF).

III. Background and Related Work

A. Localized Cumulative Distribution

The LCD is defined as

$$F(\underline{m}, b) = \int_{\mathbb{R}^{d_\xi}} f(\underline{\xi}) K(\underline{\xi}, \underline{m}, b) \, \mathrm{d}\underline{\xi} \ ,$$

where $f(\underline{\xi})$ is a PDF over $\mathbb{R}^{d_\xi}$, and $K(\underline{\xi}, \underline{m}, b)$ is a kernel function centered at $\underline{m} \in \mathbb{R}^{d_\xi}$ with bandwidth $b \in \mathbb{R}_+$. The kernel function is typically chosen as a Gaussian kernel [11]

$$K(\underline{\xi}, \underline{m}, b) = \exp\left(-\frac{\|\underline{\xi} - \underline{m}\|_2^2}{2b^2}\right) \ .$$

The LCD provides a smoothed, localized representation of a distribution, where the kernel center $\underline{m}$ determines the location and the bandwidth b controls the degree of localization.

For specific distribution types, the LCD can be computed analytically. For a Gaussian distribution $f(\underline{\xi}) = \mathcal{N}(\underline{\xi}; \hat{\underline{\xi}}, \mathbf{C})$, the LCD has the closed-form expression

$$F_{\text{G}}(\underline{m}, b) = (2\pi)^{\frac{d_\xi}{2}} b^{d_\xi} \mathcal{N}(\underline{m} - \hat{\underline{\xi}}, \mathbf{C} + b^2 \mathbf{I}) \ . \tag{3}$$

For a Dirac mixture $f(\underline{\xi}) = \sum_{n=1}^{N} w_n \delta(\underline{\xi} - \underline{\xi}_n)$, LCD reads

$$F_{\text{DM}}(\underline{m}, b) = \sum_{n=1}^{N} w_n K(\underline{\xi}_n, \underline{m}, b) \ .$$

B. Modified Cramér–von Mises Distance

To compare distributions via their LCDs, the mCvM distance

$$d_{\text{CM}} = \int_0^{b_{\max}} w(b) \int_{\mathbb{R}^{d_\xi}} \left(\tilde{F}(\underline{m}, b) - F(\underline{m}, b)\right)^2 \mathrm{d}\underline{m} \, \mathrm{d}b \ , \tag{4}$$

integrates the squared difference of two LCDs $\tilde{F}(\underline{m}, b)$ and $F(\underline{m}, b)$ over both the spatial domain and the bandwidth parameter, where $b_{\max}$ is the maximum bandwidth, and $w(b)$ is a weighting function, typically chosen as $w(b) = 1/b^{d_\xi - 1}$.

This distance metric has been used for various tasks, including finding optimal sample positions for Dirac mixture approximations of continuous distributions [11] and sample reduction of Dirac mixtures [12].

IV. LCD-Based Local Distance Measure

The standard mCvM distance integrates over the entire spatial domain, making it inherently global rather than local. We therefore extend the mCvM framework to enable local calibration assessment in the joint input-output space. Our approach consists of three main components: First, we derive LCD transforms for both test data (a Dirac mixture) and model predictions (a hybrid PDF). Second, we develop a localized version of the modified Cramér–von Mises distance that depends on a query point in the input space. Third, we provide analytical and semi-analytical solutions for efficient computation.

A. LCD Transforms

Since test data and model predictions may differ in scale across the input and output dimensions, we employ an anisotropic kernel with separate bandwidths for each dimension. We denote the joint variable as $\underline{\xi} = [\underline{x}^\top, \underline{y}^\top]^\top \in \mathbb{R}^{d_x+d_y}$, the kernel center as $\underline{m} = [\underline{m}_x^\top, \underline{m}_y^\top]^\top$, and the bandwidth vector as $\underline{b} = [b_x, b_y]^\top$, where $b_x \in \mathbb{R}_+$ and $b_y \in \mathbb{R}_+$ control localization in input and output space respectively. The anisotropic product kernel is defined as

$$K(\underline{\xi}, \underline{m}, \underline{b}) = K(\underline{x}, \underline{m}_x, b_x) K(\underline{y}, \underline{m}_y, b_y) \ , \tag{5}$$

where each factor is a Gaussian kernel in its respective dimension. This separable structure is crucial for computational efficiency and allows independent integration over input and output components.

1) LCD of Test Data: The LCD of the test data Dirac mixture (1) is obtained by integrating with the anisotropic kernel and exploiting the separability of the product kernel, resulting in

$$\begin{aligned} F_{\mathrm{DM}}(\underline{m}, \underline{b}) &= \int_{\mathbb{R}^{d_x}} \int_{\mathbb{R}^{d_y}} f_{\mathrm{DM}}(\underline{x}, \underline{y}) K(\underline{\xi}, \underline{m}, \underline{b}) \, \mathrm{d}\underline{y} \, \mathrm{d}\underline{x} \\ &= \sum_{n=1}^{N} w_n K(\underline{x}_n, \underline{m}_x, b_x) K(\underline{y}_n, \underline{m}_y, b_y) \\ &= \sum_{n=1}^{N} w_n F_{\mathrm{D}}^{(n)}(\underline{m}_x, b_x) F_{\mathrm{D}}^{(n)}(\underline{m}_y, b_y) \ , \end{aligned}$$

where $F_{\mathrm{D}}^{(n)}(\cdot) = K(\cdot)$ is the LCD of the n-th Dirac component.

2) Hybrid LCD of Model Predictions: The LCD of the hybrid distribution (2) can be derived by exploiting the separability of the product kernel. This leads to

$$\begin{aligned} F_{\mathrm{H}}(\underline{m}, \underline{b}) &= \int_{\mathbb{R}^{d_x}} \int_{\mathbb{R}^{d_y}} f_{\mathrm{H}}(\underline{x}, \underline{y}) K(\underline{\xi}, \underline{m}, \underline{b}) \, \mathrm{d}\underline{y} \, \mathrm{d}\underline{x} \\ &= \sum_{n=1}^{N} w_n K(\underline{x}_n, \underline{m}_x, b_x) \underbrace{\int_{\mathbb{R}^{d_y}} f(\underline{y} \mid \underline{x}_n) K(\underline{y}, \underline{m}_y, b_y) \, \mathrm{d}\underline{y}}_{F_{\mathrm{C}}(\underline{m}_y, b_y)} \\ &= \sum_{n=1}^{N} w_n F_{\mathrm{D}}^{(n)}(\underline{m}_x, b_x) F_{\mathrm{C}}^{(n)}(\underline{m}_y, b_y) \ , \end{aligned}$$

where $F_{\mathrm{D}}^{(n)}(\cdot)$ represents the LCD of the input space Dirac component, and $F_{\mathrm{C}}^{(n)}(\cdot)$ represents the LCD of the continuous predictive distribution in output space. For Gaussian predictive distributions, the output LCD $F_{\mathrm{C}}^{(n)}(\cdot)$ has the closed-form expression (3), enabling efficient computation.

B. Local Modified Cramér-von Mises Distance

Having derived the LCD transforms for both test data and model predictions, we now define a localized distance measure that quantifies their discrepancy at arbitrary query points in the input space.

1) Formulation and Query Point Dependence: Unlike the global mCvM distance (4) which integrates over the entire spatial domain, we propose a local variant that explicitly depends on a query location $\underline{m}_x$ in the input space. This is achieved by retaining the kernel center dependence in the input dimension, yielding

$$\begin{aligned} D(\underline{m}_x) = \int_0^{b_x^{\max}} \int_0^{b_y^{\max}} & w(b_x) w(b_y) \\ & \cdot \int_{\mathbb{R}^{d_y}} \left(F_{\mathrm{H}}(\underline{m}, \underline{b}) - F_{\mathrm{DM}}(\underline{m}, \underline{b}) \right)^2 \mathrm{d}\underline{m}_y \, \mathrm{d}b_x \, \mathrm{d}b_y \ , \end{aligned}$$

where $b_x^{\max}$ and $b_y^{\max}$ are chosen based on the scale of input and output spaces.

2) Weighting Functions: The bandwidth weighting functions $w(b_x)$ and $w(b_y)$ modulate the contribution of different kernel scales to the distance measure. We choose the weighting functions as

$$w(b_x) = \frac{1}{b_x^{d_x - c_x}} \quad \text{and} \quad w(b_y) = \frac{1}{b_y^{d_y - c_y}} \ , \tag{6}$$

where $c_x, c_y > 0$ are hyperparameters controlling the influence of bandwidth across scales. This is a generalized version of the standard LCD weighting: when $c_x = c_y = 1$, we recover the classical weighting function.

3) Local Distance Expansion and Separability: To facilitate computation, we expand the squared difference of LCDs $(F_{\mathrm{H}} - F_{\mathrm{DM}})^2 = F_{\mathrm{H}}^2 - 2F_{\mathrm{H}}F_{\mathrm{DM}} + F_{\mathrm{DM}}^2$, which leads to the sum of three distance terms $D(\underline{m}_x) = D_1(\underline{m}_x) - 2D_2(\underline{m}_x) + D_3(\underline{m}_x)$. These terms are then given by

$$\begin{aligned} D_1(\underline{m}_x) = & \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \\ & \cdot \underbrace{\int_0^{b_x^{\max}} w(b_x) F_{\mathrm{D}}^{(i)}(\underline{m}_x, b_x) F_{\mathrm{D}}^{(j)}(\underline{m}_x, b_x) \, \mathrm{d}b_x}_{I_{1,x}^{(i,j)}} \\ & \cdot \underbrace{\int_0^{b_y^{\max}} w(b_y) \int_{\mathbb{R}^{d_y}} F_{\mathrm{C}}^{(i)}(\underline{m}_y, b_y) F_{\mathrm{C}}^{(j)}(\underline{m}_y, b_y) \, \mathrm{d}\underline{m}_y \, \mathrm{d}b_y}_{I_{1,y}^{(i,j)}} \ , \end{aligned} \tag{7}$$

$$D_2(\underline{m}_x) = \sum_{i=1}^{N}\sum_{n=1}^{N} w_i w_n \cdot \underbrace{\int_0^{b_x^{\max}} w(b_x) F_{\mathrm{D}}^{(i)}(\underline{m}_x, b_x) F_{\mathrm{D}}^{(n)}(\underline{m}_x, b_x)\,\mathrm{d}b_x}_{I_{2,x}^{(i,n)}} \cdot \underbrace{\int_0^{b_y^{\max}} w(b_y) \int_{\mathbb{R}^{d_y}} F_{\mathrm{C}}^{(i)}(\underline{m}_y, b_y) F_{\mathrm{D}}^{(n)}(\underline{m}_y, b_y)\,\mathrm{d}\underline{m}_y\,\mathrm{d}b_y}_{I_{2,y}^{(i,n)}}\ , \tag{8}$$

$$D_3(\underline{m}_x) = \sum_{n=1}^{N}\sum_{m=1}^{N} w_n w_m \cdot \underbrace{\int_0^{b_x^{\max}} w(b_x) F_{\mathrm{D}}^{(n)}(\underline{m}_x, b_x) F_{\mathrm{D}}^{(m)}(\underline{m}_x, b_x)\,\mathrm{d}b_x}_{I_{3,x}^{(n,m)}} \cdot \underbrace{\int_0^{b_y^{\max}} w(b_y) \int_{\mathbb{R}^{d_y}} F_{\mathrm{D}}^{(n)}(\underline{m}_y, b_y) F_{\mathrm{D}}^{(m)}(\underline{m}_y, b_y)\,\mathrm{d}\underline{m}_y\,\mathrm{d}b_y}_{I_{3,y}^{(n,m)}}\ , \tag{9}$$

where the pairs (i,j), (i,n), (n,m) index the pairwise products. Due to the product kernel structure (5), the resulting bandwidth integrals (underbraced terms) are separable into input and output space components. Notably, since predictions and test data are evaluated at the *same discrete input locations*, all input bandwidth integrals are identical, i.e., $I_{1,x}^{(i,j)} = I_{2,x}^{(i,n)} = I_{3,x}^{(n,m)}$.

C. Computational Solutions

Having obtained the expanded terms (7) to (9), we now derive closed-form or low-dimensional numerical solutions for their integral components.

1) Integration over Input Bandwidth (b_x): As noted in Sec. IV-B3, all input bandwidth integrals $I_{\cdot,x}$ have the identical form. Therefore, w.l.o.g., we derive the solution for $I_{1,x}^{(i,j)}$. Inserting the LCDs parts and the weighting function (6) yields

$$\int_0^{b_x^{\max}} \frac{1}{b_x^{d_x - c_x}} \exp\left(-\frac{1}{b_x^2} c_{i,j}\right) \mathrm{d}b_x\ , \tag{10}$$

with $c_{i,j} = \frac{1}{2}\|\underline{x}_i - \underline{m}^x\|_2^2 + \frac{1}{2}\|\underline{x}_j - \underline{m}^x\|_2^2$. Using the substitution $u = \frac{1}{b_x^2}$ and $\mathrm{d}b_x = -\frac{1}{2}u^{-\frac{3}{2}}\,\mathrm{d}u$ (limits $b_x = 0 \to u = \infty$, $b_x = b_x^{\max} \to u = \frac{1}{(b_x^{\max})^2}$) and flipping the limits gives

$$\frac{1}{2}\int_{\frac{1}{(b_x^{\max})^2}}^{\infty} u^{\frac{d_x - c_x - 3}{2}} \exp\left(-u\, c_{i,j}\right) \mathrm{d}u$$

Next, we substitute $t = c_{i,j}u$, $\mathrm{d}u = \frac{1}{c_{i,j}}\,\mathrm{d}t$ and adjusting the limits gives

$$\frac{1}{2}\cdot\frac{1}{(c_{i,j})^{\frac{d_x - c_x - 1}{2}}}\int_{\frac{c_{i,j}}{(b_x^{\max})^2}}^{\infty} t^{\frac{d_x - c_x - 3}{2}} \exp\left(-t\right) \mathrm{d}t$$

Transforming the power of t to a more familiar form, by using t^{s-1} with $s = \frac{d_x - c_x - 1}{2}$, the integral can now be identified as the upper incomplete gamma function

$$\Gamma(s,x) = \int_x^{\infty} t^{s-1} e^{-t}\,\mathrm{d}t\ ,$$

with $x = \frac{c_{i,j}}{(b_x^{\max})^2}$.

2) Integration over Output Bandwidth (b_y): The output bandwidth integrations, i.e., $I_{1,y}, I_{2,y}$, depend on the specific distribution types. If we assume that predictive distributions are Gaussian (common in regression models), we can exploit the closed-form expression for the Gaussian LCD and further simplify $I_{1,y}$ and $I_{2,y}$. In total, we have to consider three combinations: Gaussian-Gaussian, Gaussian-Dirac, and Dirac-Dirac products.

a) Gaussian-Gaussian Product ($I_{1,y}$): The integral part over $\underline{m}_y$ in $I_{1,y}$ using Gaussian LCDs (3) reads

$$\int_{\mathbb{R}^{d_y}} F_{\mathrm{G}}^{(i)}(\underline{m}_y, b_y) F_{\mathrm{G}}^{(j)}(\underline{m}_y, b_y)\,\mathrm{d}\underline{m}_y\ ,$$

and can be solved analytically using Lemma IX.2. Inserting this result into $I_{1,y}$ gives

$$I_{1,y}^{i,j} = \int_0^{b_y^{\max}} \frac{1}{b_y^{d_y - c_y}} \cdot \frac{(2\pi)^{\frac{d_y}{2}} b_y^{2d_y}}{\sqrt{\det(\mathbf{C}_i + \mathbf{C}_j + 2b_y^2\mathbf{I})}} \cdot \exp\left(-\frac{1}{2}(\underline{\hat{y}}_i - \underline{\hat{y}}_j)^\top (\mathbf{C}_i + \mathbf{C}_j + 2b_y^2\mathbf{I})^{-1}(\underline{\hat{y}}_i - \underline{\hat{y}}_j)\right)\mathrm{d}b_y$$

which can be solved numerically by *one-dimensional integration*, as it only depends on a scalar b_y.

b) Gaussian-Dirac Product ($I_{2,y}$): As in *a)*, the integral part over $\underline{m}_y$ can be solved analytically using Lemma IX.2 (by setting $\mathbf{C}_j = \mathbf{0}$). Inserting this result into $I_{2,y}$ gives

$$I_{2,y}^{(i,n)} = \int_0^{b_y^{\max}} \frac{1}{b_y^{d_y - c_y}} \cdot \frac{(2\pi)^{\frac{d_y}{2}} b_y^{d_y}}{\sqrt{\det(\mathbf{C}_i + 2b_y^2\mathbf{I})}} \cdot \exp\left(-\frac{1}{2}(\underline{\hat{y}}_i - \underline{y}_n)^\top (\mathbf{C}_i + 2b_y^2\mathbf{I})^{-1}(\underline{\hat{y}}_i - \underline{y}_n)\right)\mathrm{d}b_y\ ,$$

which is solved using one-dimensional numerical integration.

c) Dirac-Dirac Product ($I_{3,y}$): Analogously to *a)*, the integral over $\underline{m}_y$ of the product of two Dirac LCDs can be solved analytically using Lemma IX.2 (setting $\mathbf{C}_i = \mathbf{C}_j = \mathbf{0}$). The integral $I_{3,y}$ is then given by

$$I_{3,y}^{(n,m)} = \int_0^{b_y^{\max}} \frac{\pi^{\frac{d_y}{2}} b_y^{d_y}}{b_y^{d_y - c_y}} \exp\left(-\frac{1}{4b_y^2}\|\underline{y}_n - \underline{y}_m\|_2^2\right)\mathrm{d}b_y\ ,$$

which has a closed-form solution that can be derived using the substitution $t = \frac{c_{n,m}}{b_y^2}$, with $c_{n,m} = \frac{1}{4}\|\underline{y}_n - \underline{y}_m\|_2^2$. Recognizing the integral as an upper incomplete gamma function with $s = -\frac{c_y + 1}{2}$ gives us

$$I_{3,y}^{(n,m)} = \frac{1}{2}\pi^{\frac{d_y}{2}} c_{n,m}^{\frac{c_y+1}{2}} \Gamma\left(-\frac{c_y+1}{2}, \frac{c_{n,m}}{(b_y^{\max})^2}\right)\ .$$

In case of the diagonal terms, where $n = m$, the exponential term vanishes, and the integral can be solved directly by

$$I_{3,y}^{(n,n)} = \pi^{\frac{d_y}{2}} \int_0^{b_y^{\max}} b_y^{c_y}\,\mathrm{d}b_y = \pi^{\frac{d_y}{2}}\left[\frac{b_y^{c_y+1}}{c_y+1}\right]_0^{b_y^{\max}} = \pi^{\frac{d_y}{2}}\frac{(b_y^{\max})^{c_y+1}}{c_y+1}\ ,$$

provided that $c_y > -1$, which is satisfied for typical choices $c_y \geq 0$.

D. Handling Boundary Effects in the Input Space

In kernel-based methods, e.g., in kernel density estimation, boundary effects are a well-known issue [13]. When a kernel is centered near the edge of the data region, it partially covers areas where no observations exist, leading to biased estimates. Existing corrections typically involve modifying the kernel shape or renormalizing near boundaries [13], [14]. However, these approaches often require explicit knowledge of boundary locations and thus would not fit well with our framework, which is designed to evaluate the distance at arbitrary query points without assuming known boundaries.

We therefore pursue a different approach: we keep the distance computation unchanged and instead penalize distances evaluated near the data boundaries, thereby preserving the (semi-)analytical integral solutions. This penalty is motivated by the observation that when a kernel center $\underline{m}_x$ lies near the edge of the test data, data are distributed highly asymmetrically around $\underline{m}_x$, which causes the distance measure to collapse to zero. Thus, we detect boundary effects by checking whether the mean of nearby test data deviates from the kernel center. This is possible without knowing exactly where the boundaries, edges, or holes in the data are. Define the set of nearby points as

$$\mathcal{I}_{\text{near}}(\underline{m}_x, b) = \{i \in \{1, \dots, N\} \mid \|\underline{x}_i - \underline{m}_x\| \leq \alpha b_x^{\max}\} \ ,$$

where $\alpha > 0$ determines the proximity threshold (e.g., $\alpha = 2$). The data average $\bar{\underline{x}}_{\text{near}} = \frac{1}{|\mathcal{I}_{\text{near}}|} \sum_{i \in \mathcal{I}_{\text{near}}} \underline{x}_i$, where $|\mathcal{I}_{\text{near}}|$ is the number of elements in $\mathcal{I}_{\text{near}}$, satisfies $\bar{\underline{x}}_{\text{near}} \approx \underline{m}_x$ at interior points due to symmetric data distribution, while at boundaries asymmetry causes deviation.

We quantify this through $\Delta_x(\underline{m}_x) = \|\underline{m}_x - \bar{\underline{x}}_{\text{near}}\|_2^2$ and apply a threshold τ to distinguish genuine boundary effects from natural fluctuations

$$\tilde{\Delta}_x(\underline{m}_x) = \begin{cases} 0 & \text{if } \Delta_x(\underline{m}_x) \leq \tau \\ \Delta_x(\underline{m}_x) & \text{otherwise} \end{cases} \ ,$$

where $\tilde{\Delta}_x(\underline{m}_x)$ is the clipped deviation used for penalty weighting. Note that this threshold loosens the implicit assumption of uniform input data distribution, allowing for asymmetry without triggering penalties. The penalized distance measure

$$D_{\text{pen}}(\underline{m}_x) = w_{\text{pen}}(\underline{m}_x)\, D(\underline{m}_x)$$

uses the penalty weight

$$w_{\text{pen}}(\underline{m}_x) = \exp\left(\lambda\left(1 + \tilde{\Delta}_x(\underline{m}_x)\right)^2 - \lambda\right) \ ,$$

where $\lambda > 0$ controls penalty strength. This yields $w_{\text{pen}} = 1$ at interior points and exponential growth near boundaries.

We set the proximity threshold $\alpha = 2$ based on the effective support of Gaussian kernels (extending approximately to distance $2b$ from the center), and the boundary threshold $\tau = 0.75\, b_x^{\max}$ to separate genuine asymmetry from sampling noise while scaling with the spatial scale. The penalty strength λ controls the aggressiveness of boundary rejection, with typical values $\lambda \in [1, 20]$ providing moderate to strict penalization.

V. Implementation and Computational Aspects

A. Numerical Implementation

Computing the distance measure involves evaluating the upper incomplete gamma function for various parameter combinations. Many scientific libraries implement this function efficiently, but careful attention to edge cases is needed.

a) Handling Non-positive Indices: The upper incomplete gamma function $\Gamma(s, x)$ in standard library implementations typically requires $s > 0$. However, depending on the input dimension d_x and weighting exponent c_x, we may encounter negative values of $s = \frac{d_x - c_x - 1}{2}$. For negative s, we use the recurrence relation $\Gamma(s+1, x) = s\,\Gamma(s, x) + x^s e^{-x}$ which allows computing $\Gamma(s, x)$ for $s < 0$ by recursively increasing the argument until reaching positive values.

When $s = 0$ (which occurs with the standard mCvM weighting $c_x = 1$ in univariate input space), the incomplete gamma function reduces to the exponential integral $\Gamma(0, x) = -\operatorname{Ei}(-x)$ available in scientific libraries such as SciPy.

b) Numerical Stability: When $c_{i,j} = 0$ (arising when $\underline{x}_i = \underline{x}_j = \underline{m}_x$), the integral (10) over b_x diverges. Numerical stability is ensured by adding a small regularization constant $\epsilon \approx 10^{-15}$ to $c_{i,j}$ in this case.

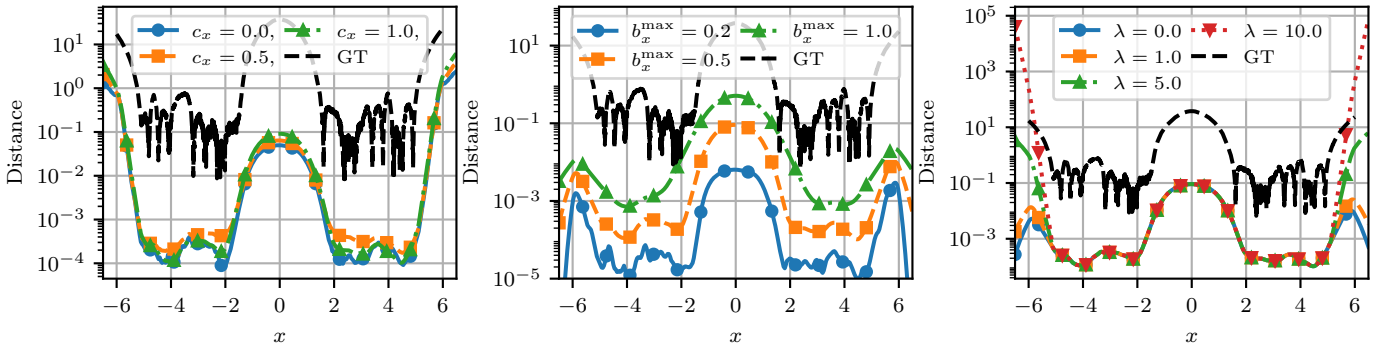
B. Computational Complexity and Efficiency

Computing $D(\underline{m}_x)$ has time complexity $\mathcal{O}(N^2)$, where N is the number of hybrid PDF components (equal to the number of test data points). Quadratic complexity arises from the double summation in the three distance terms. To reduce computation, we leverage the following strategies ($\mathcal{O}(N^2)$ remains):

1) Since only $I_{\cdot,x}$ depend on the specific query point $\underline{m}_x$, the integrals $I_{1,y}, I_{2,y}, I_{3,y}$ can be pre-computed for all component pairs (i, j), (i, n), (n, m) before querying any input point $\underline{m}_x$. This, in particular, includes all terms for which numerical integration is needed.
2) The $\underline{m}_x$ dependent integrals $I_{\cdot,x}$ can be computed efficiently using the closed-form solutions derived in Sec. IV-C1, and are equal for all three distance terms (see, Sec. IV-B3).
3) The symmetric structure of the summands in D_1 and D_3 terms allows computing only $N(N+1)/2$ unique pairwise integrals rather than N^2.
4) Points far away from the query point $\underline{m}_x$ contribute negligibly due to Gaussian kernel decay. Setting a distance threshold beyond which contributions are ignored reduces N for each query point, significantly lowering the quadratic complexity. We found that a threshold of $\geq 4b_x^{\max}$ does not influence results while substantially reducing computation.

VI. Numerical Evaluation

We evaluate our method on two synthetic regression scenarios, one with a one-dimensional input space and one with a two-dimensional input space. To assess local calibration quality, we use a reference distance between predictive distributions and the true data-generating process. In all experiments, this



(a) weight coeff. c_x (b) max. input kernel width $b_x^{\max}$ (c) penalty factor λ

Fig. 2: Distance measure $D_{\text{pen}}(\underline{m}_x)$ sensitivity to the hyperparameters $c_x = c_y$, $b_x^{\max}$ and λ. As a reference, the 1-Wasserstein distance between the predictive distributions and the true data-generating process (ground truth (GT) distance) is shown as black dashed line.

reference is the 1-Wasserstein distance. The 1-Wasserstein distance between the normally distributed outputs of the true process, $p(y \mid x_n) = \mathcal{N}\left(\hat{y}_{\text{GT}}, \sigma_{\text{GT}}^2\right)$ and the predictions $p(y \mid \underline{x}_n, \mathcal{D}) = \mathcal{N}\left(\hat{y}_{\text{Pred}}, \sigma_{\text{Pred}}^2\right)$ is given by [15], [16]

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

where $\mu = \hat{y}_{\text{GT}} - \hat{y}_{\text{Pred}}$, $\sigma^2 = (\sigma_{\text{GT}} - \sigma_{\text{Pred}})^2$, and $F_{\mathcal{N}}(\cdot)$ is the standard normal CDF.

Our evaluation proceeds in two stages. First, we examine parameter sensitivity and consistency with ground truth distance in a single-input scenario where behavior can be clearly interpreted. Then, we validate functionality on a two-dimensional input example.

A. Single-Input Regression Scenario

In our first evaluation scenario, we use the same single-input cubic regression example as in [5]. This scenario provides a controlled setting for examining parameter sensitivity and consistency with ground truth distance. The setup contains 2000 training points and 2400 test points generated from $\boldsymbol{y} = x^3 + \boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 9)$. Training inputs $x_n \in \mathcal{X}_{\text{Train}}$ are drawn uniformly from $[-5, 5]$ and test inputs $x_n \in \mathcal{X}_{\text{Test}}$ 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 No-U-Turn Sampler (NUTS) [17], Stochastic Variational Inference (SVI) [18], and probabilistic backpropagation (PBP) [19] following the settings of [5]. For evaluation, 300 kernel centers ($\underline{m}_x$) are equally spaced over $[-6.5, 6.5]$. The mCvM parameter sensitivity analysis is performed on NUTS predictions, while the consistency check with different inference methods is performed across all three inference techniques.

Weighting Function: The influence of the weighting function parameters c_x and c_y is shown in Fig. 2a. With $\lambda = 5$ and fixed, the overall trends are similar across weights, but the degree of smoothing changes. In particular, $c_x = c_y = 0.5$ yields a smoother curve, e.g., in the input range $[2, 5]$, whereas $c_x = c_y = 0$ and $c_x = c_y = 1$ retain more local variation.

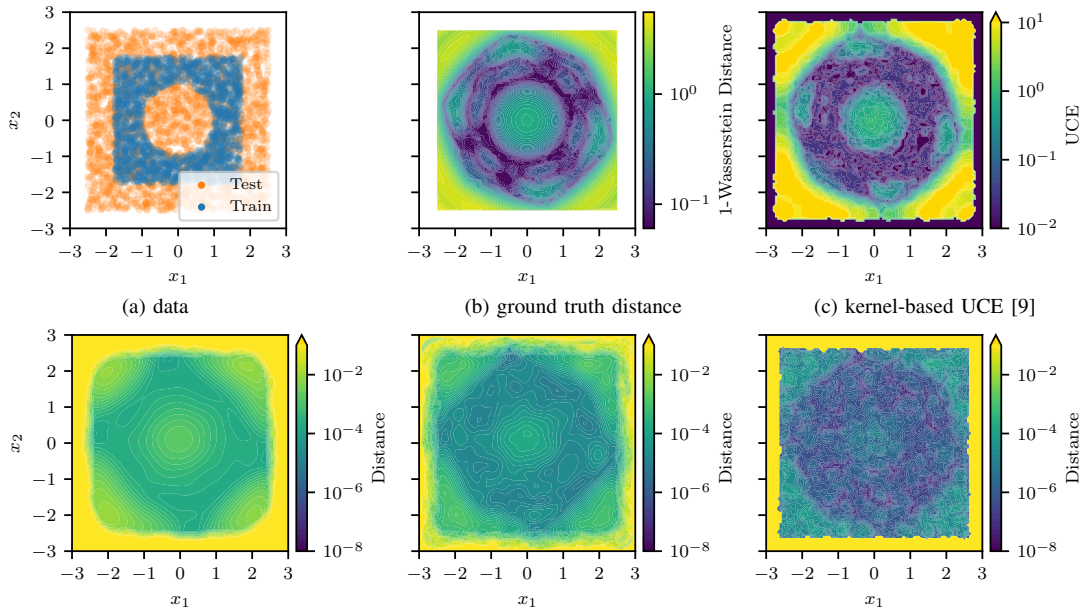
Maximum Kernel Width: The influence of the maximum kernel width $b_x^{\max}$ is shown in Fig. 2b. With $\lambda = 0$ and $c_x = c_y = 1.0$ fixed, larger $b_x^{\max}$ values yield smoother distance curves, while smaller $b_x^{\max}$ capture more local variation at the cost of increased noise. We also observe an upward shift of the distance as $b_x^{\max}$ increases, which is expected since larger kernel widths integrate over a larger region.

Penalty Parameter: The influence of the boundary penalty parameter λ is shown in Fig. 2c. With $c_x = c_y = 1.0$ and $b_x^{\max} = 0.5$ fixed, setting $\lambda = 0$ yields no penalty and causes the distance to collapse toward zero near the boundaries, despite errors indicated by the ground truth distance. Increasing λ counteracts this collapse by penalizing boundary effects. In data-rich regions, the distance remains largely insensitive to λ, since the penalty mainly affects regions with scarce or missing training data. Note that points with $|x| > 6$ lie outside the data range. Here, the penalty term dominates, corresponding to a pessimistic assumption about predictive uncertainty.

Inference Methods: In Fig. 4, we apply the method to predictions from SVI and PBP to assess whether the distance depends on the inference technique. With $\lambda = 5$, $b_x^{\max} = 0.5$, and $c_x = c_y = 1.0$ fixed, the local mCvM distance ($D_{\text{pen}}(\underline{m}_x)$) follows the same overall tendency as the 1-Wasserstein reference but is smoother, reflecting the kernel-based construction. The two distances are not identical, yet low Wasserstein values correspond to low mCvM values, and vice versa, across both SVI and PBP. This supports the view that the proposed measure is not specific to a single inference method.

B. Multiple-Input Regression Scenario

In this scenario, we evaluate on the same two-dimensional nonlinear regression task as in [6]. The data is generated according to $\boldsymbol{y} = \sin\left(x_1^2 + x_2^2\right) + \boldsymbol{\epsilon}$ 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. 3a. The BNN is trained using SVI [18] following the settings of [6]. The kernel center points $\underline{m}_x$, where the predictive distributions are evaluated, are



(a) data (b) ground truth distance (c) kernel-based UCE [9]

(d) $D_{\text{pen}}(\underline{m}_x)$, $b_x^{\max} = 0.5$ (e) $D_{\text{pen}}(\underline{m}_x)$, $b_x^{\max} = 0.25$ (f) $D_{\text{pen}}(\underline{m}_x)$, $b_x^{\max} = 0.1$

Fig. 3: Results of the multiple-input scenario. 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 (ground truth distance) is shown. In (d)–(f) the local mCvM distance $D_{\text{pen}}(\underline{m}_x)$ is shown for different values of $b_x^{\max}$. For comparison, the results from our prior region-based method [9] are shown in (c). Note that all plots share the same range for the vertically displayed x_2-axis.

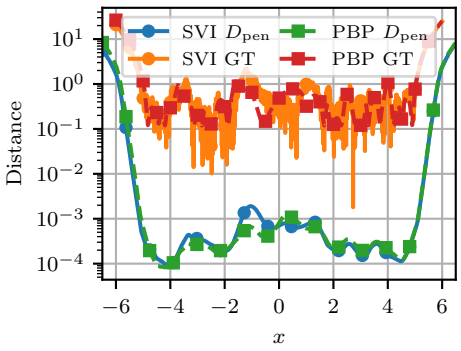


Fig. 4: Distance measure $D_{\text{pen}}(\underline{m}_x)$ for different inference methods (SVI, and PBP) compared to the ground truth distance.

placed on an equally spaced grid over $[-3, 3]^2$. The mCvM parameters are fixed to $\lambda = 20$, $b_x^{\max} \in \{0.1, 0.15, 0.25, 0.5\}$, and $c_x = c_y = 1.0$. For comparison, we include the closest related approach [9], which assesses local calibration using predefined spherical kernel regions and the UCE as calibration measure.

For a quantitative comparison, Spearman's correlation coefficients between the local measures and the ground truth distance evaluated at the 3000 test points are shown in Tab. I. Spearman correlation is used since the measures differ in scale but should ideally be monotonically related, i.e., a low ground truth distance should correspond to a low local distance and vice versa.

For $b_x^{\max} = 0.5$, the LCD yields a smoother distance (see Fig. 3) than both the ground truth in Fig. 3b and the baseline [9] in Fig. 3c. The latter captures local variations more sharply but exhibits pronounced boundary effects, whereas our penalty term mitigates these effects when λ is chosen appropriately. As $b_x^{\max}$ decreases, the local distance captures finer local

TABLE I: Spearman's correlation coefficients (with confidence intervals (CIs)).

Method	Spearman correlation	95 % CI
$D_{\text{pen}}(\underline{m}_x), b_x^{\max} = 0.5$	0.734	$[0.717, 0.750]$
$D_{\text{pen}}(\underline{m}_x), b_x^{\max} = 0.25$	0.846	$[0.836, 0.856]$
$D_{\text{pen}}(\underline{m}_x), b_x^{\max} = 0.15$	0.893	$[0.885, 0.900]$
$D_{\text{pen}}(\underline{m}_x), b_x^{\max} = 0.1$	**0.905**	$[0.898, 0.911]$
kernel-based UCE [9]	0.900	$[0.892, 0.906]$

variations, resulting in higher correlation with the ground truth distance, with the best correlation achieved at $b_x^{\max} = 0.1$. However, $b_x^{\max}$ cannot be chosen arbitrarily small. Although Gaussian kernels always assign nonzero weight to every data point, data points far from the kernel center are exponentially downweighted. Hence, a kernel center with no nearby samples effectively receives near-zero contribution from all data points.

VII. Discussion

Our approach yields an input-point-specific distance as a measure of uncertainty calibration by comparing test data with test predictions. Sensitivity analysis in the one-dimensional scenario shows that $b_x^{\max}$ controls the smoothness of the distance measure, trading off locality against noise. In practice, $b_x^{\max}$ can be selected based on the expected scale of local variations or tuned by cross-validation. Additional smoothing can be beneficial when the distance is used as a loss in gradient-based optimization, as it reduces local minima. The boundary penalty parameter λ mitigates boundary effects, preventing the distance from collapsing near regions with little or no test data. This addresses a known weakness of region-based approaches, such as

the spherical kernel method [9], which can exhibit strong boundary artifacts. Finally, the measure behaves consistently across inference methods, underscoring its general applicability.

VIII. Conclusion

We introduced a principled local distance for uncertainty-aware regression that operates directly in the joint input-output space. By representing test data as a Dirac mixture and model predictions as a hybrid PDF in the joint input-output space, and by using LCD-based integral transforms, we obtain a local mCvM distance that avoids heuristic partitioning of the input space. A key contribution is deriving closed-form and semi-analytical solutions for the integral components, enabling efficient computation without requiring numerical integration over the full joint input-output space. Compared to global distance metrics, our approach provides locality-aware calibration assessment, and empirical results in regression scenarios show that the proposed measure tracks the overall tendency of a ground truth reference while providing smooth, locality-aware behavior and consistent performance across inference methods.

IX. Appendix

Lemma IX.1: Integration over two Gaussians is given by

$$\int_{\mathbb{R}^{d_\xi}} \mathcal{N}(\hat{\underline{\xi}}_1, \mathbf{C}_1)\mathcal{N}(\hat{\underline{\xi}}_2, \mathbf{C}_2)\,\mathrm{d}\underline{\xi} = \frac{1}{\sqrt{\det(2\pi(\mathbf{C}_1+\mathbf{C}_2))}} \cdot \exp\left(-\frac{1}{2}\left(\hat{\underline{\xi}}_1-\hat{\underline{\xi}}_2\right)^\top (\mathbf{C}_1+\mathbf{C}_2)^{-1}\left(\hat{\underline{\xi}}_1-\hat{\underline{\xi}}_2\right)\right)$$

Proof: The product of two Gaussians can be reformulated as $c\mathcal{N}(\hat{\underline{\xi}}_3, \mathbf{C}_3) = \mathcal{N}(\hat{\underline{\xi}}_1, \mathbf{C}_1)\mathcal{N}(\hat{\underline{\xi}}_2, \mathbf{C}_2)$. Integrating both sides yields $c\int_{\mathbb{R}^{d_\xi}} \mathcal{N}(\hat{\underline{\xi}}_3, \mathbf{C}_3)\,\mathrm{d}\underline{\xi}$. Since the integral of a Gaussian is 1, the result equals c (given in [20, Sec. 8.1.8]). ■

Lemma IX.2: The integral over the product of two Gaussian LCDs, $F_1(\underline{m}, b)$ and $F_2(\underline{m}, b)$, is given by

$$\int_{\mathbb{R}^{d_\xi}} F_1(\underline{m}, b)F_2(\underline{m}, b)\,\mathrm{d}\underline{m} = \frac{(2\pi)^{\frac{d_\xi}{2}} b^{2d_\xi}}{\sqrt{\det(\mathbf{C}_1+\mathbf{C}_2+2b^2\mathbf{I})}} \cdot \exp\left(-\frac{1}{2}(\hat{\underline{\xi}}_1-\hat{\underline{\xi}}_2)^\top(\mathbf{C}_1+\mathbf{C}_2+2b^2\mathbf{I})^{-1}(\hat{\underline{\xi}}_1-\hat{\underline{\xi}}_2)\right)$$

where $\hat{\underline{\xi}}_1$, $\hat{\underline{\xi}}_2$ and covariances $\mathbf{C}_1$, $\mathbf{C}_2$ are the parameters of the Gaussians before LCD transformation.

Proof: By inserting the Gaussian LCDs (3) and completing the factors to Gaussians, we can write the integral as

$$(2\pi)^{d_\xi} b^{2d_\xi} \int_{\mathbb{R}^{d_\xi}} \mathcal{N}\left(\hat{\underline{\xi}}_1, \mathbf{C}_1+b^2\mathbf{I}\right)\mathcal{N}\left(\hat{\underline{\xi}}_2, \mathbf{C}_2+b^2\mathbf{I}\right)\mathrm{d}\underline{m}\ .$$

Solving the integral using Lemma IX.1, yields

$$\frac{(2\pi)^{d_\xi} b^{2d_\xi}}{\sqrt{\det(2\pi(\mathbf{C}_1+\mathbf{C}_2+2b^2\mathbf{I}))}} \cdot \exp\left(-\frac{1}{2}(\hat{\underline{\xi}}_1-\hat{\underline{\xi}}_2)^\top(\mathbf{C}_1+\mathbf{C}_2+2b^2\mathbf{I})^{-1}(\hat{\underline{\xi}}_1-\hat{\underline{\xi}}_2)\right)\ .$$

Simplifying the 2π factors gives the final result. ■

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] M. Walker, M. Reith-Braun, and U. D. Hanebeck, "Local calibration testing in supervised machine learning models using input space kernels," in *Proceedings of the 28th International Conference on Information Fusion (Fusion)*, Rio de Janeiro, Brazil, Jul. 2025, pp. 1–8.

[10] ——, "Weaknesses of the ANEES and new calibration measures for multivariate predictions," in *Proceedings of the 2025 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, College Station, TX, USA, 2025, pp. 1–8.

[11] 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.

[12] U. D. Hanebeck, "Optimal reduction of multivariate Dirac mixture densities," *at - Automatisierungstechnik*, vol. 63, no. 4, pp. 265–278, Apr. 2015.

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

[14] J. Dai and S. Sperlich, "Simple and effective boundary correction for kernel densities and regression with an application to the world income and Engel curve estimation," *Computational Statistics & Data Analysis*, vol. 54, no. 11, pp. 2487–2497, Nov. 2010.

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

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

[17] 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.

[18] 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.

[19] 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.

[20] K. B. Petersen and M. S. Pedersen, "The Matrix Cookbook," Nov. 2012.