SAND2022-13964C

The Evaluation and Calibration of Epistemic Uncertainty Estimates

Anonymous Author

Anonymous Institution

Abstract

Recently, a surge of interest has been given to quantifying epistemic uncertainty (EU), the reducible portion of uncertainty due to lack of data. However, few studies attempt to directly evaluate the quality of EU estimates. We adopt the definition of EU as the difference in accuracy between the optimal prediction and the current prediction at a given point, and we propose to evaluate the quality of EU estimates based on how well they agree with the observed accuracy gain when more data is added. Our proposed evaluation procedure also gives rise to a way of improving EU estimates by learning a calibration mapping. We demonstrate our evaluation and calibration method on real and simulated datasets, where we assess and compare the quality of two standard EU estimators obtained from a Gaussian process classifier.

1 INTRODUCTION

Uncertainty quantification has received widespread attention because of its importance in evaluating and improving the reliability of machine learning models. There are two sources of uncertainty: the uncertainty due to the inherently random effects of the data generating process (i.e., *aleatoric*) and the uncertainty that can be reduced with the addition of more data (i.e., *epistemic*).

Recently, efforts have been made to quantify the contribution of epistemic uncertainty (EU), due to its natural application in active learning, out-of-distribution (OOD) detection (Mukhoti et al., 2021), and classification with delay (Senge et al., 2014). However, there have been much fewer attempts to evaluate the quality of EU estimates. Current studies that propose new EU estimators typically validate the quality of their estimates via downstream tasks such as active learning and OOD detection (Postels et al., 2020). Proper evaluation of EU estimates is further complicated by the fact that there is no consensus on the precise definition of epistemic uncertainty; see Postels et al. (2021) and Lahlou et al. (2021) for two competing definitions.

In this work, we define the epistemic uncertainty in a classification setting to be the difference in accuracy between the current prediction and the optimal prediction at a particular point. We develop a procedure to measure the accuracy gain at each point between our prediction using the current train data and our prediction using a larger dataset, where the accuracy gain serves as a proxy for the epistemic uncertainty. We use it to introduce two ways to measure the quality of the EU estimates: the epistemic expected calibration error (EECE) and the rank correlation between the EU estimates and the accuracy gain. Moreover, this also enables us to develop a procedure for calibrating our EU estimates to reduce the EECE. Using this evaluation procedure, we compare and evaluate the quality of two standard EU estimators from a Gaussian process classifier. Additionally, we empirically show that we can improve the quality of our EU estimates through calibration on simulated and real data.

1.1 Related Work

There is a great deal of work on evaluating the quality of predictive uncertainty estimates, the sum of the aleatoric and epistemic uncertainties. These include the expected calibration error (ECE), negative log likelihood, Brier score (Ovadia et al., 2019),

Preliminary work. Under review by AISTATS 2023. Do not distribute.

OOD detection (Maddox et al., 2019), area under the curve (Chen et al., 2019), area under the risk curve (Brosse et al., 2020), and the performance of these under dataset shift (Ovadia et al., 2019).

In contrast, there is much less work on evaluating the quality of EU estimates. Frequently, no definition of EU is provided, so the evaluation is based on either the usual predictive uncertainty metrics, or simply the eye test. For example, Kendall and Gal (2017) evaluate their EU estimates using the standard ECE. Nilsen et al. (2022) display the images that have the highest and lowest estimated EU and argue that the images with higher estimated EU are visually harder to classify. Recently, OOD detection has been a popular approach to evaluate the quality of EU estimates (Postels et al., 2020, 2021; Mukhoti et al., 2021). Although OOD detection is a quality that good EU estimates should have, this evaluation is incomplete since it does not evaluate the quality of the EU estimates on the in-domain data. In any case, it is apparent that a formal definition of epistemic uncertainty is important to meaningfully evaluate the quality of its estimates.

Our work is more closely related to Lahlou et al. (2021), who propose the definition of EU that we use. However, they use the negative log-likelihood as their loss for classification instead of the 0/1 loss. Using the 0/1 loss enables us to develop a more direct approach of evaluating EU estimates via the EECE. They also propose using the rank correlation between the EU estimates and the accuracy, but note that this evaluation is only valid in the absence of aleatoric uncertainty. We extend their approach by taking the rank correlation between the EU estimates and the accuracy *gain*.

2 DEFINITIONS OF UNCERTAINTY

We focus on the binary classification setting. Suppose we are given a training dataset $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathcal{X}$ and $y_i \in \{0, 1\}$, for some input space $\mathcal{X}$. For any point $\boldsymbol{x} \in \mathcal{X}$, we assume the corresponding label $y \in \{0, 1\}$ follows some distribution denoted by $P(y|\boldsymbol{x})$. The Bayes optimal classifier under the 0/1 loss is $f^*(\boldsymbol{x}) = \arg\max_{y \in \{0,1\}} P(y|\boldsymbol{x})$.

Next, we give the following definitions of predictive, aleatoric, and epistemic uncertainty of a classifier $\hat{f}$ at a specific point. These definitions are a special case of those given in Lahlou et al. (2021), modified for the binary classification setting with the 0/1 loss:

Definition 1. *The **predictive uncertainty** of classifier $\hat{f}$ at point $\boldsymbol{x}$ is given by*

$$\mathcal{U}(\boldsymbol{x}) = 1 - P(\hat{f}(\boldsymbol{x})|\boldsymbol{x})$$

Definition 2. *The **aleatoric uncertainty** at a point $\boldsymbol{x}$ is given by*

$$\mathcal{A}(\boldsymbol{x}) = 1 - P(f^*(\boldsymbol{x})|\boldsymbol{x})$$

Definition 3. *The **epistemic uncertainty** of a classifier $\hat{f}$ at a point $\boldsymbol{x}$ is given by*

$$\begin{aligned}\mathcal{E}(\boldsymbol{x}) &= \mathcal{U}(\boldsymbol{x}) - \mathcal{A}(\boldsymbol{x}) \\ &= P(f^*(\boldsymbol{x})|\boldsymbol{x}) - P(\hat{f}(\boldsymbol{x})|\boldsymbol{x})\end{aligned}$$

The epistemic uncertainty is the difference between the accuracy of the optimal classifier with your current classifier at a given point. This definition of epistemic uncertainty is attractive for the following reasons:

- It conforms to the idea that EU should represent the uncertainty that results from a lack of data. Large epistemic uncertainty at a point indicates that the optimal classifier is much better than your current classifier at the given point. As you gain more and more data, the classifier will converge to the optimal classifier at that point, causing the epistemic uncertainty to decrease to zero.
- It produces the identity $\mathcal{U}(\boldsymbol{x}) = \mathcal{A}(\boldsymbol{x}) + \mathcal{E}(\boldsymbol{x})$; that is, the predictive (total) uncertainty can be decomposed into the irreducible aleatoric uncertainty and the reducible epistemic uncertainty.
- This definition has a straightforward, practical application. Similar to how a good estimate of predictive uncertainty is useful when it is possible to reject giving a predicction, a good estimate of epistemic uncertainty is useful when it is possible to delay giving a prediction (until more data is collected so that our prediction is improved).

3 EPISTEMIC UNCERTAINTY ESTIMATORS

A classifier learning algorithm $\mathcal{L}$ is a function that takes in a dataset $\mathcal{D}_{\text{train}}$ and outputs a function $\hat{f} = \mathcal{L}(\mathcal{D}_{\text{train}})$ where $\hat{f} : \mathcal{X} \to \{0, 1\}$. Likewise, an EU learning algorithm $\mathcal{L}_{\text{epi}}$ is a function that takes in a dataset $\mathcal{D}_{\text{train}}$ and outputs a function $\hat{\mathcal{E}} = \mathcal{L}_{\text{epi}}(\mathcal{D}_{\text{train}})$ where $\hat{\mathcal{E}} : \mathcal{X} \to [0, 1]$. The output $\hat{\mathcal{E}}(\boldsymbol{x})$ should be an estimate of the EU associated with prediction $\hat{f}(\boldsymbol{x})$.

3.1 Estimation via Gaussian Processes

In this work, we will use a Gaussian process classification model to provide EU estimates, since Gaussian processes are widely believed to provide high-quality uncertainty estimates, which distinguishes it from other powerful models such as deep neural networks (Ober et al., 2021). In binary Gaussian process classification, given $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, we assume that the outputs are generated from latent values which are Gaussian distributed; that is, for $i \in 1:N$,

$$y_i = \mathbb{1}(g_i + \epsilon_i > 0), \quad \epsilon_i \sim \text{logistic}(0,1),$$
$$\boldsymbol{g} = (g_1, \ldots, g_N) \sim \mathcal{N}(\boldsymbol{0}_{N\times 1}, \boldsymbol{K}_{N\times N}),$$
$$\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{for some kernel function } k(\cdot,\cdot)$$

We approximate the distribution of the training latent values given the observed labels as $p(\boldsymbol{g}|\boldsymbol{y}) \approx \mathcal{N}(\boldsymbol{g}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ via variational inference and predict the latent value g_* of a new point $\boldsymbol{x}_*$ as

$$\begin{aligned} p(g_*|\boldsymbol{y}) &= \int p(g_*|\boldsymbol{g}) p(\boldsymbol{g}|\boldsymbol{y}) d\boldsymbol{g} \\ &\approx \int p(g_*|\boldsymbol{g}) \mathcal{N}(\boldsymbol{g}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\boldsymbol{g}, \\ &= \mathcal{N}(g_*|\mu_*, \sigma_*^2) \end{aligned}$$

This yields the classifier

$$\hat{f}_{\text{GP}}(\boldsymbol{x}_*) = \arg\max_{y_* \in \{0,1\}} p(y_*|\boldsymbol{y})$$

where

$$\begin{aligned} p(y_*|\boldsymbol{y}) &= \int p(y_*|g_*) p(g_*|\boldsymbol{y}) dg_* \\ &= \int \sigma(g_*) \mathcal{N}(g_*|\mu_*, \sigma_*^2) dg_*, \\ \sigma(a) &= (1+\exp(-a))^{-1} \end{aligned}$$

We will consider two EU estimators:

Mutual Information:

$$\hat{\mathcal{E}}_{\text{M}}(\boldsymbol{x}_*) = H(p(y_*|\boldsymbol{y})) - \int H(\sigma(g_*)) \mathcal{N}(g_*|\mu_*, \sigma_*^2) dg_*$$

Entropy:

$$\hat{\mathcal{E}}_{\text{E}}(\boldsymbol{x}_*) = H(p(y_*|\boldsymbol{y}))$$

where $H(p) = -p\log p - (1-p)\log(1-p)$. We will use $\mathcal{L}_{\text{GP}}$, $\mathcal{L}_{\text{epi},M}$ and $\mathcal{L}_{\text{epi},E}$ to denote the learning algorithms that produce $\hat{f}_{\text{GP}}$, $\hat{\mathcal{E}}_{\text{M}}$, and $\hat{\mathcal{E}}_{\text{E}}$, respectively.

The mutual information, $\hat{\mathcal{E}}_M$, is widely used as a way to quantify EU (Hüllermeier and Waegeman, 2021). The first term in $\hat{\mathcal{E}}_{\text{M}}$, the entropy, represents the predictive uncertainty, while the second term, the expected entropy, represents the aleatoric uncertainty. Note that $\hat{\mathcal{E}}_{\text{E}}$ makes no effort to account for the aleatoric uncertainty; however, it may be more efficient than $\hat{\mathcal{E}}_{\text{M}}$ when the aleatoric uncertainty is low. In practice, we compute these integrals using Monte Carlo integration.

4 EVALUATION AND CALIBRATION

4.1 Evaluation Metrics

Suppose we have a training dataset $\mathcal{D}_{\text{train}}$, which we use to train a classifier $\hat{f} = \mathcal{L}(\mathcal{D}_{\text{train}})$ and an EU estimator $\hat{\mathcal{E}} = \mathcal{L}_{\text{epi}}(\mathcal{D}_{\text{train}})$. We are asked to provide predictions and EU estimates for test points from $\mathcal{D}_{\text{test}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^T$, and we desire to evaluate the quality of our EU estimates. Based on Definition 3, the direct way to evaluate the quality of our EU estimates is to measure the discrepancy between $\hat{\mathcal{E}}(\boldsymbol{x}_i)$ and $\mathcal{E}(\boldsymbol{x}_i)$.

We will not have access to $\mathcal{E}(\boldsymbol{x}_i)$ since it depends on the true distribution $P(y_i|\boldsymbol{x}_i)$; instead, we must approximate it using the observed labels y_i. In this section, we assume that we know $f^*(\boldsymbol{x}_i)$ for each test point $\boldsymbol{x}_i$. This allows us to approximate $\mathcal{E}(\boldsymbol{x}_i)$ with the empirical improvement of using f^* over $\hat{f}$.

Definition 4. *Given a classifier $\hat{f}$ and a pair $(\boldsymbol{x}, y)$, we define the **gain** of the point $\boldsymbol{x}$ by*

$$\begin{aligned} \text{gain}(\boldsymbol{x}, y) &= \mathbb{1}(f^*(\boldsymbol{x}) = y) - \mathbb{1}(\hat{f}(\boldsymbol{x}) = y) \\ &= \begin{cases} 1 & \text{if } \hat{f}(\boldsymbol{x}) \neq y, f^*(\boldsymbol{x}) = y \\ -1 & \text{if } \hat{f}(\boldsymbol{x}) = y, f^*(\boldsymbol{x}) \neq y \\ 0 & \text{else} \end{cases} \end{aligned}$$

The gain is 1 if the optimal prediction improves the current prediction, -1 if worsens, and 0 if stays the same. Notice that the gain is an empirical version of the EU in the sense that

$$\begin{aligned} \mathbb{E}\left[\text{gain}(\boldsymbol{x}, y)\right] &= \mathbb{E}\left[\mathbb{1}(f^*(\boldsymbol{x}) = y)\right] - \mathbb{E}\left[\mathbb{1}(\hat{f}(\boldsymbol{x}) = y)\right] \\ &= P(f^*(\boldsymbol{x})|\boldsymbol{x}) - P(\hat{f}(\boldsymbol{x})|\boldsymbol{x}) = \mathcal{E}(\boldsymbol{x}) \end{aligned}$$

where the expectation is taken over y with the true conditional distribution $P(y|\boldsymbol{x})$.

This motivates our first evaluation metric: we partition the test points into M bins based on their estimated EU and find the average gain and estimated EU for each bin. We then take the average of

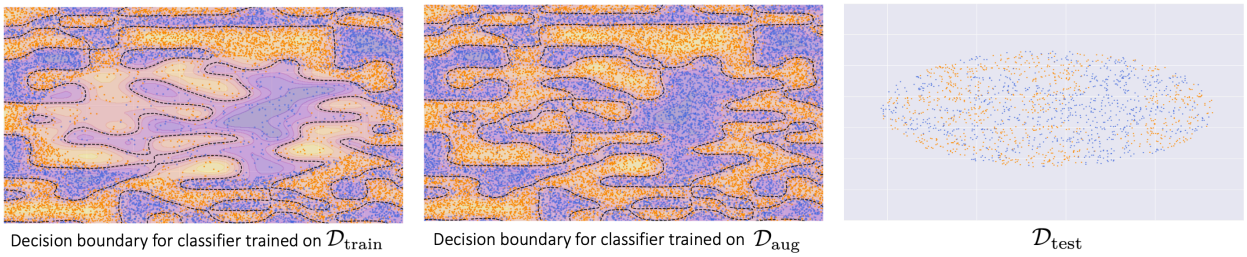

Figure 1: Our larger dataset $\mathcal{D}_{\text{aug}}$ fills in the circular region carved out in $\mathcal{D}_{\text{train}}$, allowing us compute the gain on $\mathcal{D}_{\text{test}}$.

the absolute differences, weighted by the number of points in each bin.

Definition 5. *Given a test dataset* $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^T$*, a classifier* $\hat{f}$*, an EU estimator* $\hat{\mathcal{E}}$*, and a partition* $\{I_1, \ldots, I_M\}$ *of intervals of [0,1], the* ***epistemic expected calibration error*** *is given by*

$$\text{EECE} = \frac{1}{M} \sum_{m=1}^{M} |B_m||\text{gain}(B_m) - \hat{\mathcal{E}}(B_m)|$$

where $B_m = \left\{ i \in 1:T \mid \hat{\mathcal{E}}(\boldsymbol{x}_i) \in I_m \right\}$,

$$\text{gain}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \text{gain}(\boldsymbol{x}_i, y_i),$$

$$\hat{\mathcal{E}}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{\mathcal{E}}(\boldsymbol{x}_i), \quad m \in 1:M$$

Notice the similarity with the definition of ECE (Guo et al., 2017); the only difference is that we replace the empirical version of predictive uncertainty the empirical version of EU. The EECE provides useful information when we want to know how much the predictions on a set of observations can be improved. For example, if our EU estimator is calibrated, we can make an informed decision about whether or not it is worth it to delay making predictions until we collect more data.

In some situations, we only desire that our EU estimates are ordered correctly: $\hat{\mathcal{E}}(\boldsymbol{x}_1) > \hat{\mathcal{E}}(\boldsymbol{x}_2)$ implies $\mathcal{E}(\boldsymbol{x}_1) > \mathcal{E}(\boldsymbol{x}_2)$. For example, we might be given a fixed number of observations that we are allowed to delay. This leads to our second evaluation metric: the Spearman rank correlation coefficient between the EU estimates and the gain:

Definition 6. *Given a test dataset* $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^T$*, a classifier* $\hat{f}$*, and an EU estimator* $\hat{\mathcal{E}}$*, the* ***epistemic correlation*** *of* $\hat{\mathcal{E}}$ *is given by*

$$\rho_{\text{epi}} = r_s \left(\left\{ \hat{\mathcal{E}}(\boldsymbol{x}_i), \text{gain}(\boldsymbol{x}_i, y_i) \right\}_{i=1}^T \right)$$

where r_s *is the Spearman rank correlation coefficient.*

The epistemic correlation also has the advantage that it directly compares the quality of two EU estimators without the choice of the partition of intervals. In addition, we also find that a higher epistemic correlation also tends to result superior *calibrated* EU estimates, which we will discuss in Section 5.

4.2 Computing the Evaluation Metrics

We can only evaluate the gain on test points for which we know the optimal prediction. To give an example where this is true, suppose we are given the training dataset $\mathcal{D}_{\text{train}}$ shown on the left of Fig. 1, and we are asked to provide predictions and EU estimates for $\mathcal{D}_{\text{test}}$ shown on the right of Fig. 1. The dotted lines show the decision boundary of our learned classifier. Notice that our predictions can be significantly improved, since $\mathcal{D}_{\text{train}}$ does not have much data in the circular region that we are asked to predict. If we had access to the larger dataset $\mathcal{D}_{\text{aug}}$, shown on the middle of Fig. 1, then we could compute the evaluation metrics defined in the previous section by replacing f^* with $\hat{f}_{\text{aug}} = \mathcal{L}(\mathcal{D}_{\text{aug}})$ in Definition 4, since is $\hat{f}_{\text{aug}}$ is close to optimal on $\mathcal{D}_{\text{test}}$. We can think of $\mathcal{D}_{\text{aug}}$ as the hypothetical future data we would have if we were to delay making our predictions.

4.2.1 Data Splitting Procedure

Drawing inspiration from the above illustration, for our experiments, given a dataset $\mathcal{D}$, we *provide* ourselves with $\mathcal{D}_{\text{aug}}$ by removing ball-shaped regions of a fixed radius from the high density regions of $\mathcal{D}$. Specifically, in each iteration i, we identify a center $\boldsymbol{c}_i$ and obtain the set of points within a radius R of $\boldsymbol{c}_i$, which we distribute in the following way:

- $p_i\%$ of the points are added to $\mathcal{D}_{\text{train}}$.
- $q_i\%$ of the points are added to $\mathcal{D}_{\text{test}}$.

- $(1 - p_i - q_i)\%$ of the points are combined with $\mathcal{D}_{\text{train}}$ to form $\mathcal{D}_{\text{aug}}$.

Here, q_i should be selected to be small enough so that the assumption that $\hat{f}_{\text{aug}} \approx f^*$ on $\mathcal{D}_{\text{test}}$ holds, but large enough that there are enough points in $\mathcal{D}_{\text{test}}$ for a proper evaluation. Full detail of this procedure is given in Algorithm 1 in the Supplementary Material. See the left part of Fig. 2 for an illustration of what the resulting sets can look like in a simple case. On real data, we find data-rich regions by sampling points and computing the number of neighbors in the ball-shaped region around them; more details are given in Section 5.4.

Note that we specifically choose the data-rich regions of $\mathcal{D}$ to subset the ball-shaped regions, since those are the regions of the input space that we are most likely to be able to learn a near-optimal predictor. However, even if the predictions are not optimal, this experimental procedure is still meaningful in the sense that measuring how well the EU estimates match up with the improvement on a larger dataset is still an informative procedure to assess their performance.

4.3 Calibration Method

Our approach to splitting the datasets suggests an approach to improve our EU estimates: we take $\mathcal{D}_{\text{train}}$ and similarly create three *calibration* datasets: $\mathcal{D}_{\text{train},2}$, $\mathcal{D}_{\text{aug},2}$, $\mathcal{D}_{\text{test},2} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{T}$. We then train a classifier $\hat{f}_{\text{tr},2}$ on $\mathcal{D}_{\text{train},2}$ and another classifier $\hat{f}_{\text{aug},2}$ on $\mathcal{D}_{\text{aug}}, 2$, calculate the epistemic uncertainty estimates and gain in accuracy on $\mathcal{D}_{\text{test},2}$ and use those results to learn a *calibration mapping* $\gamma : [0, 1] \to [0, 1]$, which we can then apply to our EU estimates on $\mathcal{D}_{\text{test}}$. Specifically, for each interval I_m, we obtain the average accuracy gain of the points whose estimated EU lie in the interval:

$$\overline{\gamma}_m = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(\hat{f}_{\text{tr},2}(\boldsymbol{x}_i) \neq y_i) - \mathbb{1}(\hat{f}_{\text{aug},2}(\boldsymbol{x}_i) \neq y_i)$$

$$B_m = \left\{ i \in 1 : T \mid \hat{\mathcal{E}}(\boldsymbol{x}_i) \in I_m \right\}, \quad m \in 1 : M$$

Then for each test observation, we define $\hat{\gamma}_i$ to be the average accuracy gain of the points in the interval it belongs to:

$$\hat{\gamma}_i = \overline{\gamma}_{m_i}, \quad m_i = \prod_{m=1}^{M} m^{\mathbb{1}(i \in B_m)}, \quad i \in 1 : T$$

We then obtain our calibration mapping γ by performing an isotonic regression where the inputs are $\hat{\mathcal{E}}(\boldsymbol{x}_i)$ and the outputs are $\hat{\gamma}_i$:

$$\gamma = \text{IsotonicRegression}\left(\left\{ (\hat{\mathcal{E}}(\boldsymbol{x}_i), \hat{\gamma}_i) \right\}_{i=1}^{T} \right)$$

We use isotonic regression to learn the calibration mapping simply because its popularity and ease of implementation. See Guo et al. (2017) for other methods.

4.3.1 Creating the Calibration Datasets

An open question is how exactly to partition $\mathcal{D}_{\text{train}}$ to create the calibration datasets without information about the evaluation procedure. One approach is to search for a set of inputs to the procedure described in Section 4.2.1. to partition $\mathcal{D}_{\text{train}}$ such that the distribution of the EU estimates on $\mathcal{D}_{\text{test},2}$ and $\mathcal{D}_{\text{test}}$ are similar, a process that would likely be quite time-consuming. We leave the study on such delicate splitting procedures for future work. Throughout this paper, we will sidestep this issue by using the procedure used to create the evaluation datasets as the splitting procedure for the calibration datasets. Nevertheless, in Section 5.3.2 and the Supplementary Material, we will explore two scenarios where this assumption is not satisfied (i.e., the calibration datasets differ significantly from the evaluation datasets). We also note that when we perform calibration on the real data in Section 5.4, the pattern carved out in the evaluation and calibration datasets will likely be quite different, despite the same splitting procedure being used.

5 EXPERIMENTS

5.1 Large Scale Gaussian Process Classification

We use the Gaussian process classification model to make predictions and obtain uncertainty estimates, since Gaussian Processes are widely held to produce the high quality uncertainty estimates. However, the traditional GP Classifier model does not scale well with large amounts of data, necessitating the use of large scale GPs. The most scalable approach is through variational inference with inducing points. This approach reduces the training cost from $\mathcal{O}(N^3)$ to $\mathcal{O}(m^3)$ per minibatch, where m is the number of inducing points (Liu et al., 2020). We use the GPyTorch implementation of the method in (Wenzel et al., 2019).

5.2 Experimental Methodology

For each dataset $\mathcal{D}$, we follow the following steps in our experiments:

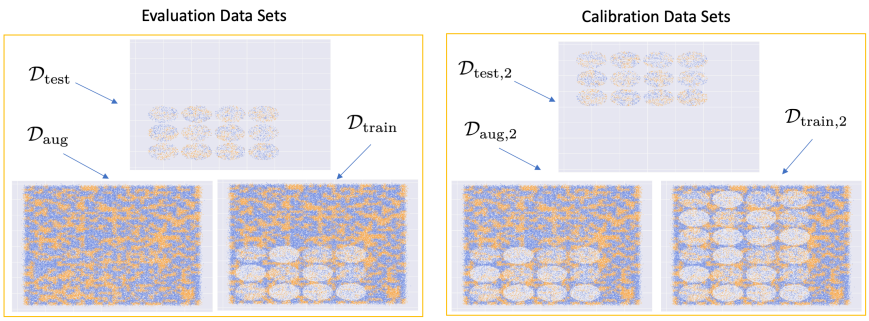

Figure 2: Toy Data

1. We create our evaluation datasets $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{test}}$, and $\mathcal{D}_{\text{aug}}$ using the procedure described in Section 4.2.
2. We train our base Gaussian process classifier $\hat{f}_{\text{tr}} = \mathcal{L}_{\text{GP}}(\mathcal{D}_{\text{train}})$, our two EU estimators $\hat{\mathcal{E}}_{\text{M}} = \mathcal{L}_{\text{epi,M}}(\mathcal{D}_{\text{train}})$ and $\hat{\mathcal{E}}_{\text{E}} = \mathcal{L}_{\text{epi,E}}(\mathcal{D}_{\text{train}})$ and our enhanced Gaussian process classifier $\hat{f}_{\text{aug}} = \mathcal{L}_{\text{GP}}(\mathcal{D}_{\text{aug}})$.
3. For each EU estimator, we compute a partition of twenty intervals $I_1, \ldots, I_{20}$ over the range of the EU estimates on $\mathcal{D}_{\text{test}}$. The intervals are made so that each interval contains roughly the same number of points.
4. For each EU estimator, we use the partition $I_1, \ldots, I_{20}$ to obtain a calibration mapping γ using the procedure described in section 4.3.
5. Having obtained our base classifier $\hat{f}_{\text{tr}}$, our improved classifier $\hat{f}_{\text{aug}}$, our two EU estimators $\hat{\mathcal{E}}_{\text{M}}$ and $\hat{\mathcal{E}}_{\text{E}}$, and our two calibrated EU estimators $\gamma_{\text{M}}(\hat{\mathcal{E}}_{\text{M}})$ and $\gamma_{\text{E}}(\hat{\mathcal{E}}_{\text{E}})$, we can now calculate our evaluation metrics.

Further implementation details, such as the hyperparameters chosen to train the large scale Gaussian process classifier, are given in the supplementary material.

5.3 Toy Data

First, we will illustrate our evaluation method on toy data. To generate our toy data, we first sample $N = 142,693$ points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ from the following Gaussian mixture model with $K = 38 \cdot 46$ groups:

$$p(\boldsymbol{x}_i | \boldsymbol{z}_i = [j,k]) = \mathcal{N}\left(\boldsymbol{x}_i \mid [j,k], \boldsymbol{I}_2\right)$$
$$p(\boldsymbol{z}_i) = \frac{1}{K} \mathbb{1}(\boldsymbol{z}_i \in \{1, \ldots, 38\} \times \{1, \ldots, 46\})$$

We generate the corresponding labels by randomly partitioning each cluster into equal groups. That is, the labels $y_1, \ldots, y_N$ are generated according to

$$p(y_i | \boldsymbol{z}_i = (j,k), w_1, \ldots, w_K) = \begin{cases} \mathbb{1}(w_{j+46k} \leqslant K/2) & \text{if } y_i = 1 \\ \mathbb{1}(w_{j+46k} > K/2) & \text{if } y_i = 0 \end{cases}$$

where $w_1, \ldots, w_K$ are a random permutation of the index set $\{1, \ldots, K\}$:

$$p(w_1, \ldots, w_K) = \frac{1}{K!} \mathbb{1}(w_i \in \{1, \ldots, K\}, w_i \neq w_j)$$

This results in a dataset $\mathcal{D}$ which consists of a dense grid of points with a random pattern for the decision boundary, shown in Fig. 2. We follow the methodology described in Section 5.2 to calculate our evaluation metrics. In the first step, we carve out non-overlapping discs arranged in a grid, as shown in the left portion of Fig. 2. Note that some of the discs have fewer points removed than others; this is done so that the test points have a wide range of EU estimates. We can then repeat this process on the untouched upper region of $\mathcal{D}_{\text{train}}$ to create the calibration datasets, $\mathcal{D}_{\text{train},2}$, $\mathcal{D}_{\text{test},2}$ and $\mathcal{D}_{\text{aug},2}$ as shown in the right portion of Fig. 2.

5.3.1 Results

In Table 1, we provide the evaluation metrics for both methods, averaged over 6 seeds. The standard deviation is given after the $\pm$ symbol. We see that $\hat{\mathcal{E}}_{\text{E}}$ has superior epistemic correlation than $\hat{\mathcal{E}}_{\text{M}}$, likely because the test data does not contain many points with high aleatoric uncertainty, which are the points on the boundary between the two classes. We also observe that $\hat{\mathcal{E}}_{\text{E}}$, before calibration, has an extremely high EECE, making it unsuitable as a direct estimator for EU. However, after calibration, $\hat{\mathcal{E}}_{\text{E}}$ has a lower EECE than $\hat{\mathcal{E}}_{\text{M}}$. In general, we would expect the method with the higher epistemic correlation to be better suited to calibration, since calibration only depends on the quality of the ranking produced by the EU estimator.

Next, we will analyze the EECE in more detail. For

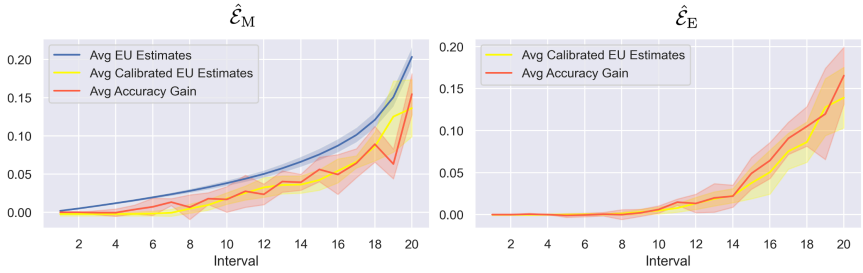

Figure 3: Average Epistemic Uncertainty for each Interval on Toy Data.

Table 1: Evaluation Metrics on Toy Data

Method	ρ_{epi}	EECE	Calib. EECE
$\hat{\mathcal{E}}_{\mathrm{M}}$	0.108 $\pm$0.023	**0.0249 $\pm$ 0.002**	0.0182 $\pm$ 0.005
$\hat{\mathcal{E}}_{\mathrm{E}}$	**0.121 $\pm$ 0.028**	0.393 $\pm$0.014	**0.0152 $\pm$ 0.002**

each interval, we obtain the average estimated EU, the average calibrated EU, and the average accuracy gain over the points in that interval. We visualize this by plotting these quantities over the interval number on the x-axis. We then average these values over six seeds; the results are displayed in Fig. 3; the curves are created by interpolating the points using matplotlib's default connecting method. For each curve, we also show the standard deviation over the six seeds by shading in the area with the corresponding color. We do not display the curve for the uncalibrated $\hat{\mathcal{E}}_{\mathrm{E}}$ estimates because they are too large and distort the graph.

From the red accuracy gain curves, although there is a moderate amount of variance in the accuracy gain at each interval for both methods, the increasing trends verify their positive correlation with the accuracy gain. Both of the yellow calibrated EU curves match the red curve relatively well, but the larger intervals tend to be more difficult to estimate, as indicated by the larger standard deviation. Notably, $\hat{\mathcal{E}}_{\mathrm{E}}$ performs better than $\hat{\mathcal{E}}_{\mathrm{M}}$, with the red and yellow curves almost completely overlapping on the first 7 intervals. We also see that $\hat{\mathcal{E}}_{\mathrm{M}}$ before calibration tends to overestimate the accuracy gain, but it is a decent rough estimator for the EU.

Overall, both methods are able to be calibrated to achieve a small EECE, averaging a difference of less than 0.02. This indicates that we are able to take information about the EU of a region in the input space and *apply* it to another region of the input space. Even though the pattern of the decision boundary for each part of the input space is random, there appears to be a certain regularity to the EU.

5.3.2 Effect of Calibration Mismatch

As shown in Fig. 2, the evaluation datasets and the calibration datasets resemble each other for our experiments. Since we cannot ensure this in practice, in the Supplementary Material, we explore two situations in which there is a mismatch between the calibration and evaluation data. We consider two modifications to our experiments:

1. The evaluation datasets remain the same as in Fig. 2; for the calibration datasets, we carve three large circles instead of twelve small circles.
2. The calibration datasets remain the same as in Fig. 2; for the evaluation datasets, we modify $\mathcal{D}_{\text{test}}$ by removing points with low aleatoric uncertainty in some of the discs.

We find that in modification 1, the results do not change dramatically; both EU estimators still perform relatively well. In modification 2, the performance of $\hat{\mathcal{E}}_{\mathrm{E}}$ decreases and $\hat{\mathcal{E}}_{\mathrm{M}}$ is comparatively better. Overall, our results indicate that calibration can still be beneficial even when there is a mismatch with the evaluation data.

5.4 Real Data: EMNIST-Letters and Kuzushiji-49

We will consider two real datasets: Kuzushiji-49 and EMNIST-Letters. Kuzushiji-49 is a Japanese character classification dataset with 270,912 observations and 49 classes. EMNIST-Letters is an English letter classification dataset with 131,600 observations and 47 classes. For each dataset, we use PCA to reduce the dimensionality to 50 and convert it to a binary classification problem by setting the even classes to 0 and the odd classes to 1.

We follow the procedure from Section 5.2 to calculate our evaluation metrics. In the first step, where we create the evaluation datasets, we note that unlike the toy data where we could easily carve out non-overlapping balls, the real datasets are high di-

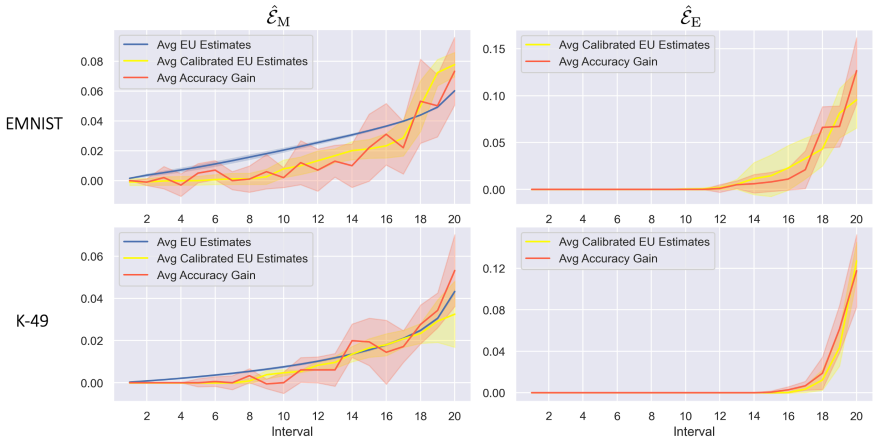

Figure 4: Average Epistemic Uncertainty for each Interval on Real Data.

Table 2: Evaluation Metrics on Real Data.

Dataset	Method	ρ_{epi}	EECE	Calib. EECE
EMNIST	$\hat{\mathcal{E}}_{\text{M}}$	0.088 $\pm$ 0.02	**0.0151 $\pm$ 0.003**	0.0122 $\pm$ 0.0027
	$\hat{\mathcal{E}}_{\text{E}}$	**0.108 $\pm$ 0.016**	0.41 $\pm$0.03	**0.0096 $\pm$ 0.005**
K-49	$\hat{\mathcal{E}}_{\text{M}}$	0.065 $\pm$0.015	**0.0063 $\pm$ 0.001**	0.0060 $\pm$ 0.0004
	$\hat{\mathcal{E}}_{\text{E}}$	**0.101 $\pm$ 0.014**	0.33 $\pm$0.019	**0.0043 $\pm$ 0.001**

mensional, making it impossible to do the same. In addition, the density varies over the input space, so we must search for the high density regions to carve out. In our implementation, we search for regions of high density by sampling random points, calculating the number of points in a ball around them, and selecting points that have a sufficient number of neighbors. As a consequence of the fact that the balls carved out will have some overlap, the evaluation and calibration datasets will not resemble each other as in the toy data. A complete description of how we create the evaluation and calibration datasets can be found in the Supplementary Material.

5.4.1 Results

Our results are similar to those of the toy data. From Table 2, we see that $\hat{\mathcal{E}}_{\text{E}}$ has a higher correlation with the gain than $\hat{\mathcal{E}}_{\text{M}}$ on both datasets, just as in the toy data, although we note that $\hat{\mathcal{E}}_{\text{M}}$ does better on K-49 than EMNIST in terms of epistemic correlation. On the other hand, the overall EECE for both methods is lower on K-49 than EMNIST. From Fig. 4, we see the red average accuracy curve and yellow average calibrated estimated EU curve have higher variance on EMNIST than K-49, likely because K-49 has more than twice as much data. This results in the lower EECE on K-49.

We see that $\hat{\mathcal{E}}_{\text{M}}$ tends to overestimate the accuracy gain in EMNIST, and the calibrated EU curve improves the EECE significantly. On K-49, $\hat{\mathcal{E}}_{\text{M}}$ already estimates the accuracy gain curve quite well, averaging a difference of only 0.0063. The calibrated curve only averages a slightly lower EECE, but it also has a lower standard deviation. In both datasets, calibrated $\hat{\mathcal{E}}_{\text{E}}$ achieves a lower EECE than calibrated $\hat{\mathcal{E}}_{\text{M}}$. From the y-axis of Fig. 4, we can see the impact of the superior epistemic correlation of $\hat{\mathcal{E}}_{\text{E}}$ over $\hat{\mathcal{E}}_{\text{M}}$. The upper intervals for $\hat{\mathcal{E}}_{\text{E}}$ have larger accuracy gain, and the lower intervals have an accuracy gain of 0. This corroborates earlier work, such as Gal et al. (2017), who find $\hat{\mathcal{E}}_{\text{E}}$ to be a superior acquisition function than $\hat{\mathcal{E}}_{\text{M}}$ in active learning.

6 CONCLUSION

We introduced a novel, yet simple approach to assess the quality of EU estimates: remove points from the dataset and measure the change in accuracy, then measure how well the EU estimates match with this change. Our results support the claim that the uncertainty produced by GP's is reliable; we find that both EU estimators derived from the GP show a positive correlation with accuracy gain and produce low calibrated EECE's. We find that the entropy estimator outperforms the mutual information estimator, likely due to the small amount of aleatoric uncertainty in real datasets, although the latter can be used as a direct EU estimator without calibration. Finally, we discover that EU estimates can be calibrated: we can improve the EU estimates on a region of the input space by measuring the change in accuracy when we subset data from another region of the input space.

References

Brosse, N., C. Riquelme, A. Martin, S. Gelly, and É. Moulines (2020). On last-layer algorithms for classification: Decoupling representation from uncertainty estimation. *arXiv preprint arXiv:2001.08049*.

Chen, T., J. Navrátil, V. Iyengar, and K. Shanmugam (2019). Confidence scoring using whitebox meta-models with linear classifier probes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1467–1475. PMLR.

Gal, Y., R. Islam, and Z. Ghahramani (2017). Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pp. 1183–1192. PMLR.

Guo, C., G. Pleiss, Y. Sun, and K. Q. Weinberger (2017). On calibration of modern neural networks.

Hüllermeier, E. and W. Waegeman (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning 110*(3), 457–506.

Kendall, A. and Y. Gal (2017). What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems 30*.

Lahlou, S., M. Jain, H. Nekoei, V. Butoi, P. Bertin, J. Rector-Brooks, M. Korablyov, and Y. Bengio (2021). Deup: Direct epistemic uncertainty prediction.

Liu, H., Y.-S. Ong, X. Shen, and J. Cai (2020). When gaussian process meets big data: A review of scalable gps. *IEEE transactions on neural networks and learning systems 31*(11), 4405–4423.

Maddox, W. J., P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson (2019). A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems 32*.

Mukhoti, J., A. Kirsch, J. van Amersfoort, P. H. Torr, and Y. Gal (2021). Deterministic neural networks with appropriate inductive biases capture epistemic and aleatoric uncertainty. *arXiv preprint arXiv:2102.11582*.

Nilsen, G. K., A. Z. Munthe-Kaas, H. J. Skaug, and M. Brun (2022). Epistemic uncertainty quantification in deep learning classification by the delta method. *Neural Networks 145*, 164–176.

Ober, S. W., C. E. Rasmussen, and M. van der Wilk (2021). The promises and pitfalls of deep kernel learning. In *Uncertainty in Artificial Intelligence*, pp. 1206–1216. PMLR.

Ovadia, Y., E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Postels, J., H. Blum, C. Cadena, R. Siegwart, L. Van Gool, and F. Tombari (2020). Quantifying aleatoric and epistemic uncertainty using density estimation in latent space. *arXiv preprint arXiv:2012.03082*.

Postels, J., M. Segu, T. Sun, L. Van Gool, F. Yu, and F. Tombari (2021). On the practicality of deterministic epistemic uncertainty. *arXiv preprint arXiv:2107.00649*.

Senge, R., S. Bösner, K. Dembczyński, J. Haasenritter, O. Hirsch, N. Donner-Banzhoff, and E. Hüllermeier (2014). Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. *Information Sciences 255*, 16–29.

Wenzel, F., T. Galy-Fajou, C. Donner, M. Kloft, and M. Opper (2019). Efficient gaussian process classification using pòlya-gamma data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 33, pp. 5417–5424.