Calibration of uncertainty in the active learning of machine learning force fields

Publication Status:
Published online: 08/11/2023

DOI:
10.1088/2632-2153/ad0ab5

Document Version
Author Accepted version

Calibration of uncertainty in the active learning of machine learning force fields

To cite this article before publication: Adam Thomas-Mitchell *et al* 2023 *Mach. Learn.: Sci. Technol.* in press https://doi.org/10.1088/2632-2153/ad0ab5

Calibration of uncertainty in the active learning of machine learning force fields

Adam Thomas-Mitchell[1], Glenn Hawe[1], and Paul Popelier[2].

[1] School of Computing, Ulster University, 2-24 York Street, BT15 1AP, Belfast, Northern Ireland

[2] Department of Chemistry, The University of Manchester, Oxford Road, M13 9PL, Manchester, Great Britain

E-mail: thomas_mitchell-a@ulster.ac.uk, gi.hawe@ulster.ac.uk, paul.popelier@manchester.ac.uk

Abstract.

FFLUX is a Machine Learning Force Field that uses the Maximum Expected Prediction Error (MEPE) active learning algorithm to improve the efficiency of model training. MEPE uses the predictive uncertainty of a Gaussian Process to balance exploration and exploitation when selecting the next training sample. However, the predictive uncertainty of a Gaussian Process is unlikely to be accurate or precise immediately after training. We hypothesize that calibrating the uncertainty quantification within MEPE will improve active learning performance. We develop and test two methods to improve uncertainty estimates: post-hoc calibration of predictive uncertainty using the *CRUDE* algorithm, and replacing the Gaussian Process with a Student-t Process. We investigate the impact of these methods on MEPE for single sample and batch sample active learning. Our findings suggest that post-hoc calibration does not improve the performance of active learning using the MEPE method. However, we do find that the Student-t Process can outperform active learning strategies and random sampling using a Gaussian Process if the training set is sufficiently large.

1. Introduction

Molecular dynamics (MD) simulations are widely used in fields such as biochemistry and drug design. *Ab initio* methods, which involve solving the Schrödinger Equation, allow one to attain very accurate results on small, isolated molecular systems. Unfortunately, these methods scale extremely poorly with system size and rapidly become infeasible due to computational cost [1]. In contrast, classical force fields are used to quickly generate approximate solutions to MD simulations. Despite their advantage in terms of speed, these methods suffer from a range of limitations including the point charge description of electrostatics [2]. As a result, these methods show lower accuracy compared to their counterparts that rely on *ab initio* methods.

Machine Learning Force Fields are increasingly being used to bridge the gap between these methods due to accuracy approaching that of *ab initio* methods at speeds on a par with classical force fields [3]. FFLUX is a state-of-the-art Machine Learning Force Field that has shown success in various contexts, including small clusters of molecules, and ions [4]. This force field employs a Gaussian Process to determine the mapping between the molecular geometry of a system and atomic properties, specifically the Interacting Quantum Atoms (IQA) energy or multipole moments [5]. Whilst theoretical results have shown that kernel methods require substantially

more data compared to neural networks for certain classes of functions [6], experimental results indicate that Gaussian Processes can perform better with fewer data in the interpolation of potential energy surfaces. It is for this reason that a Gaussian Process was chosen for use in FFLUX [7].This is important as the labelling of training data for FFLUX (and Machine Learning Force Fields more generally) is computationally expensive.

Active learning has also been used by Machine Learning Force Fields to reduce the amount of training data needed to achieve a given level of accuracy. Active learning is the process of iteratively growing a training set by adding those points in feature space deemed most useful to label. Force-fields based on neural networks have used a range of active learning strategies [8], one of the most common being to select those points which have large discrepancy in their predictions across a set (committee) of models [9, 10]. A similar approach to active learning has been taken with Gaussian Process based force fields [11]. More commonly, the predictive uncertainty of a Gaussian Process is used to select points for labelling, either at the very outset of training [12] or 'on-the-fly' during a simulation to improve accuracy of a deployed model [13].

FFLUX uses the Maximum Expected Prediction Error (MEPE) active learning algorithm [14] to select data for labelling. In their comparative review of adaptive sampling methods for Gaussian Process Regression, Fuhg et al. [15] concluded that MEPE was the most well rounded method as it outperformed other algorithms in a range of benchmark tests of various dimensionality and complexity. Indeed, this method has already seen success with FFLUX in producing accurate and efficient models able to cope with large distortions for molecules such as water, ammonia, and methanol [16]. More recently, it has been used to produce chemically accurate models for high dimensional systems such as peptide-capped glycine in less time than before [17].

Despite utilising this state-of-the-art algorithm, *active learning in FFLUX has thus far not consistently outperformed models with randomly initialised training sets*. Similar observations have been made in recent work applying active learning to improve the efficiency of density functional theory predictions, where it was found that the predictive performance resulting from random sampling was nearly identical to that from the advanced EMOC algorithm [18]. The same authors also identified situations in which random sampling can significantly outperform against EMOC. Similar findings have been observed in other contexts outside of computational chemistry [19, 20, 21].

Many state-of-the-art active learning algorithms, including MEPE, make use of the predictive uncertainty of a Gaussian Process to balance the exploration of unobserved regions of feature space with the exploitation of regions known to be highly unpredictable, when selecting the next training point. However the predictive uncertainty of a Gaussian Process is rarely accurate or precise immediately after training [22]. In this paper it is hypothesized that improving the uncertainty estimates used in the MEPE algorithm will improve the performance of active learning in FFLUX.

To improve the uncertainty quantification two potential solutions are proposed:

(i) *post-hoc calibration* of predictive uncertainty;

(ii) replacing Gaussian Processes with *Student-t Processes*.

Post-hoc calibration is one method that has shown success in rectifying the issue of miscalibrated uncertainty intervals [23, 24] but it requires more labeled data in the form of a calibration set. This is problematic because, as previously stated, data generation for FFLUX is computationally expensive. Without sufficient additional data, post-hoc calibration may have limited effect in active learning, as has been shown to be the case in the related field of Bayesian optimization [25]. This motivates the pursuit of a model that provides reliable uncertainty estimates without the need for an additional calibration set. It is for this reason Student-t Processes are considered as an alternative to Gaussian Processes; they have shown superior uncertainty estimates to

Gaussian Processes given the same data [26].

In this paper we aim to:

(i) Compare the impact of post-hoc calibration and Student-t Processes on uncertainty quantification. Specifically, we aim to establish whether reliable uncertainty estimates can be obtained without the need for additional calibration data, using a Student-t Process.

(ii) Determine whether the performance of the MEPE active learning algorithm can be enhanced by improving the uncertainty estimates of the underlying model.

(iii) Develop a learning strategy for the FFLUX force field that can outperform random sampling, using the methods discussed or a combination thereof.

To the best of our knowledge this is the first time that post-hoc calibration and Student-t Processes have been used in the context of Machine Learning Force Fields. Moreover, post-hoc calibration has not been applied to Student-t Processes in any context prior to this work.

We begin by introducing the relevant methods in Section 2. This section details the theoretical background relating to FFLUX, Gaussian Processes, active learning, and uncertainty quantification including the two proposed methods: post-hoc calibration, and Student-t Processes. The results are presented and then discussed in Sections 3 and 4, respectively. Finally, we present our conclusions in Section 5.

2. Methods

2.1. FFLUX

FFLUX is an atomistic, flexible, polarisable, multipolar force field based on quantum topological atoms [27, 28]. The strategy [29] behind its construction is *ab ovo* in order to avoid undesirable limitations of more established force fields such as AMBER. For example, AMBER's point-charge description of electrostatic interaction is inherently less accurate than a multipolar one. Secondly, AMBER's artificial distinction between bonded and non-bonded potentials precludes a robust treatment of hydrogen bonds and solvation effects. The quantum topological energy partitioning method IQA [30] provides the various atomic energies that the machine learning trains for FFLUX. It is important to realise that both energies and multipole moments are associated with the same atomic partitioning, which guarantees a future-proof architecture that other post-AMBER force fields lack.

The Gaussian Process in FFLUX maps features describing the geometry of a molecular system to some atomic property, in this case the IQA energy. These features are based upon the Atomic Local Frame (ALF) [31]. In the ALF approach each atom is assigned a local coordinate system which is specified using the origin atom, an atom defining the x-axis, and an atom defining the xy-plane. The atoms with the highest and second highest priority, as determined using the Cahn-Ingold-Prelog rules, are selected to define the x-axis and xy-plane, respectively. The z-axis is then defined using the right-hand rule.

With the ALF established, the features can be determined. The initial three features in a dataset pertain to the atoms that define the ALF: the first feature is the distance from the origin atom to the atom defining the x-axis, the second feature is the distance from the origin atom to the atom defining the xy-plane, and the third feature is the angle subtending the vector from the x-axis atom to the origin atom and from the origin atom to the xy-plane atom. The remainder of the features describe the spherical polar coordinates of every other atom based on the coordinate system defined by the ALF. The spherical coordinates are given by (r, θ, ϕ), where r is the distance between a given atom, A_n, and the origin atom, A_o, θ is the polar angle of atom A_n, and ϕ is azimuthal angle of atom A_n. The ranges for the polar and azimuthal angles are $\theta \in [0, \pi]$ and $\phi \in [-\pi, \pi)$.

2.2. Dataset Generation

The experiments herein are performed on a water dimer system. For each of the 6 atoms in the system, there is a corresponding dataset. Each dataset contains 12 features, as described in the previous section, and the target variable is the IQA energy for that atom.

The datasets were constructed by randomly generating water dimer geometries and calculating the corresponding IQA energies for each atom. One of the water molecules is taken as a central water molecule and the other is then translated and rotated randomly. This procedure constructs a full sphere around the central water molecule, such that all possible configurations are considered. This is necessary as the ALF input features used in this work treat each atom uniquely. For example, exchanging the hydrogen atoms on a water molecule leads to a different feature vector that the machine learning model treats as different to the original feature vector. The goal of this dataset was to cover the first peak of the radial distribution function that is obtained experimentally [32], as well as to have full rotation of the water molecules.

It should be noted that the common method of constructing datasets by performing MD simulations was not applied here. The water dimer is a very flexible system as the hydrogen bond can be broken easily compared to a fully covalent bond, thus allowing for many possible configurations not seen in a system with only covalent bonds. Conducting MD simulations for the water dimer may not always produce a good training set as some regions of space may not be sufficiently covered. Additionally, increasing the temperature to overcome local energy minima without using any constraints often leads to the two water molecules dissociating away from each other.

2.3. Gaussian Processes

Gaussian Process (GP) regression is a probabilistic approach to modelling the relationship between input and output variables. It is typically used to estimate an unknown function $f(\mathbf{x})$, given a training set of noisy observations, $\mathcal{D} = \{(\mathbf{x_i}, y_i)\}_{i=1}^{n}$ where

$$y_i = f(\mathbf{x}_i) + \epsilon_i. \tag{1}$$

Gaussian noise is assumed, such that the noise term is sampled from a Normal distribution $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$, and a GP prior is placed on the function $f(\mathbf{x})$.

GPs are Bayesian models that describe a probability distribution over functions. It is a generalisation of the Gaussian distribution, often used to model the distribution of a single random variable. In the case of a GP, the distribution is defined over an infinite number of values that can be thought of as the values of a function at an infinite number of points.

A GP is specified by two functions: the mean function $m(\mathbf{x})$, and the kernel function $k(\mathbf{x}, \mathbf{x}')$. The mean function sets the prior expected value for any input location $\mathbf{x}$ and is often set as zero or constant for simplicity. The kernel function describes the dependence between the function inputs at different points and contains parameters that must be learnt during training. Combined, these two functions provide a probabilistic description of the objective function;

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{2}$$

One way to estimate the noise variance and the kernel hyperparameters is to maximise the negative log marginal likelihood:

$$-\ln p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}(\mathbf{y} - \mathbf{m})^\top \mathbf{\Sigma}^{-1}(\mathbf{y} - \mathbf{m}) + \frac{1}{2}\ln|\mathbf{\Sigma}| + \frac{n}{2}\ln(2\pi), \tag{3}$$

where $\mathbf{m}$ is the mean vector, $\mathbf{\Sigma} = \mathbf{K} + \sigma_n^2\mathbf{I}$, and $\mathbf{K}$ is the covariance matrix with $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Each component of the mean vector is the scalar mean function $m(\mathbf{x})$, and $\cdot^\top$ represents the transpose of a matrix or vector.

Given the initial training set and learnt hyperparameters, one can make predictions on an unseen test point $\mathbf{x}_*$ using the posterior predictive density

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(f_*|\hat{\mu}(\mathbf{x}_*), \hat{\sigma}^2(\mathbf{x}_*)), \tag{4}$$

where the predictive mean, $\hat{\mu}(\mathbf{x}_*)$, and variance $\hat{\sigma}^2(\mathbf{x}_*)$ are given by

$$\hat{\mu}(\mathbf{x}_*) = \mathbf{k}_*^\top \mathbf{\Sigma}^{-1} \mathbf{y}, \tag{5}$$

$$\hat{\sigma}^2(\mathbf{x}_*) = k_{**} - \mathbf{k}_*^\top \mathbf{\Sigma}^{-1} \mathbf{k}_*. \tag{6}$$

In these equations, $\mathbf{k}_* = [k(\mathbf{x}_*, \mathbf{x}_1), ..., k(\mathbf{x}_*, \mathbf{x}_n)]$, and $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$. The variance $\hat{\sigma}^2(\mathbf{x}_*)$ represents the predictive uncertainty of a GP at the point $\mathbf{x}_*$: the higher the value of $\hat{\sigma}^2(\mathbf{x}_*)$, the less certain we are about the prediction $\hat{\mu}(\mathbf{x}_*)$. Notably, the predictive uncertainty for a GP depends only on the observed locations $\mathbf{x}$, and not the function values $f(\mathbf{x})$.

For a more detailed explanation of Gaussian Processes, the reader is referred to [33, 34].

2.3.1. Kernel Function The RBF kernel is a common choice when constructing GPs. This kernel function models the distance between two inputs as a Gaussian, and is given by

$$k_{RBF}(\mathbf{x}, \mathbf{x}') = l \exp\left(-\sum_i \frac{(x_i - x_i')^2}{2\lambda_i}\right). \tag{7}$$

In this equation l is a scaling parameter, and λ is the lengthscale parameter.

Given the nature of the dataset described previously, this kernel is insufficient to describe all dimensions of the input. Specifically, the features corresponding to the azimuthal angle in the spherical polar coordinates for the non-ALF atoms are cyclic in the range $[-\pi, \pi)$. As such, using the RBF kernel for these features would model a linear relationship, which would not accurately describe the feature space. To address this issue, the periodic kernel is used for the cyclic dimensions and the RBF kernel is used for the remaining dimensions. The periodic kernel is described by the equation

$$k_{periodic}(\mathbf{x}, \mathbf{x}') = l \exp\left(-\sum_i \frac{2\sin^2(\frac{\pi}{p}(x_i - x_i'))}{\lambda_i}\right), \tag{8}$$

Here, the l and λ parameters are the same as for the RBF kernel but there now exists the additional periodicity parameter p. As the period of the azimuthal angle is known, this parameter can be set to $p = 2\pi$.

The final kernel is the product of the RBF kernel over the non-cyclic dimensions, and the periodic kernel over the cyclic dimensions;

$$k(\mathbf{x}, \mathbf{x}') = k_{RBF}(\mathbf{x}_{non-cyclic}, \mathbf{x}'_{non-cyclic}) \times k_{periodic}(\mathbf{x}_{cyclic}, \mathbf{x}'_{cyclic}). \tag{9}$$

2.4. Active Learning

Active learning is a subfield of machine learning concerned with the development of algorithms to actively select the data used to train a model [35]. This is in contrast to passive learning, where the labelling of data has already been performed, with no option to request further data for labelling to improve model performance. Generally, an active learning algorithm uses an *acquisition function* to quantify how attractive candidate points are for labelling. The point(s) that maximize the acquisition function are then labelled and added to the training data. By judiciously selecting training points based on their potential impact on model accuracy, the idea

is that a model can be trained to a desired level of accuracy with fewer data than would otherwise be required with passive learning. In some applications passive learning may be unavoidable, but with Machine Learning Force Fields we are generally free to select which data to label for training.

There are many approaches to active learning but here the focus is on pool-based sampling. With this approach, there exists a model, a small initial training set, and a large set of unlabelled data samples referred to as the candidate pool. First the model is trained on the initial data set. Then, the active learning algorithm selects the most useful data point from the candidate pool based on some refinement criterion. The true target value is generated for the selected point, and this labelled data sample is then added to the training set. This is repeated until some stopping criterion is met, with the model being trained on the updated training set at each iteration. The process is illustrated in figure 1.

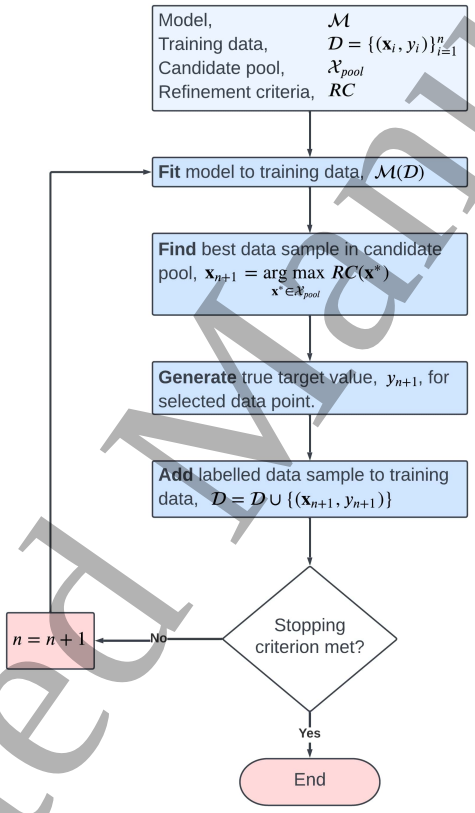


Figure 1: General pipeline for pool-based adaptive sampling.

2.4.1. MEPE As previously stated, this research concentrates on the MEPE method. At each iteration of the active learning process the 'expected prediction error' (EPE) acquisition function is calculated for each point in the candidate pool, and the point with the maximum value is selected to be added to the training set. Like most acquisition functions, expected prediction error comprises an exploration term and an exploitation term. The exploration term aims to evenly sample the entire feature space to gain a general understanding of the mapping from input to output. The exploitation term relies on knowledge extracted from available observations to identify subregions that require more data for accurate prediction. A balance factor is used to balance the contribution between the two terms.

In this instance, the predicted variance is used for the exploration term while the leave-one-out cross-validation (LOOCV) error is used for the exploitation term. The LOOCV error is computationally demanding given that a new model must be trained on many subsets of the training data. To overcome this the fast approximation [36] is used, given by

$$e_{CV}^{approx}(\mathbf{x}_i) = \frac{\left[\boldsymbol{\Sigma}^{-1}(\mathbf{y} - \mathbf{1}m)\right]_i}{(\boldsymbol{\Sigma}^{-1})_{ii}}, \tag{10}$$

where m is the prior estimate for the global mean. In order to use this approximation for points in the candidate pool, it is assumed that the LOOCV error at an unobserved point is equal to the error at the closest point in the training set.

Integrating all components returns a measure of the expected prediction error for each sample in the candidate pool as

$$\mathrm{EPE}(\mathbf{x}) = \alpha\, e_{CV}^{approx^2}(\mathbf{x}) + (1 - \alpha)\, \hat{\sigma}^2(\mathbf{x}). \tag{11}$$

The balance factor is given by

$$\alpha = \begin{cases} 0.5 & \text{if } q = 1 \\ 0.99 \min\left[0.5 \frac{e_{CV}^2(\mathbf{x}_m)}{e_{CV}^{approx^2}(\mathbf{x}_m)}, 1\right] & \text{else,} \end{cases} \tag{12}$$

where q represents the iteration number for the adaptive sampling process. The true value is calculated for the point in the candidate pool with the maximum expected prediction, and this sample is added to the training set.

2.4.2. Batch MEPE The standard MEPE algorithm updates the training set with a single sample at each iteration of the active learning process. This may be sufficient for small systems but as the dimensionality grows more training points are required to construct an accurate model. As a result, single sample adaptive sampling may be an inefficient way of training a model on larger systems. Batch-mode active learning aims to overcome this issue by selecting multiple samples at each iteration of the active learning process.

There are a number of active learning algorithms developed specifically for batch sampling [47] but in this research we investigate a variation of the MEPE algorithm: Batch MEPE. This variation has been implemented into the FFLUX pipeline and has shown success in efficiently predicting atomic energies of molecules such as peptide-capped glycine [17]. Batch MEPE involves selecting the points with the n_b largest EPE values from the candidate pool to add to the training set at each iteration, where n_b is the batch size. For this variation the balance factor must be adjusted to

$$\alpha = \frac{1}{n_b} \sum_{i=1}^{n_b} \alpha_i, \tag{13}$$

where α_i is the ith data point added to the training set in the previous iteration. As the true values for each point in the batch are calculated in parallel, the time taken to complete the active learning process is reduced by a factor of n_b.

2.5. Metrics

2.5.1. Uncertainty Quantification A model with reliable uncertainty quantification is able to accurately quantify the likelihood of outcomes associated with a predicted quantity [37]. In regression problems this effectively means that the model can produce a useful and

trustworthy confidence interval that captures the true outcome a specified percentage of the time. Uncertainty quantification can be assessed by considering both calibration and sharpness.

Calibration refers to the degree to which the uncertainty estimates provided by a model are accurate. A model that is well calibrated will provide uncertainty estimates that reflect the true underlying uncertainty in the data, such that the predicted probabilities of events match the observed frequency of those events. A model is considered to be well calibrated if it predicts that there is an $x\%$ chance of an event occurring, and the event occurs in $x\%$ of cases, over the range $x \in (0, 100)$.

We use miscalibration area as a metric to assess the calibration of a model. This is illustrated using a calibration curve as shown in figure 2. A calibration curve displays the true frequency of points in each confidence interval against the predicted proportion of points in that interval, and a well calibrated model will follow the line $y = x$. This metric measures the deviation from this ideal, such that a greater miscalibration area corresponds to a more poorly calibrated model. It should be noted that this quantity evaluates the absolute miscalibration, such that overconfidence does not correct underconfidence to produce a lower miscalibration area.

Sharpness refers to the degree to which model predictions are concentrated around the true values. A model that is sharp will have narrow uncertainty intervals, which means the predictions are more confident and precise. We measure the sharpness using the average width of the 50th and 90th percentile confidence intervals. A lower value is better as this corresponds to a more precise model with tighter confidence intervals.

It is generally desirable for a model to be both well calibrated and sharp, as this means that it is both accurate and precise in terms of predictive uncertainty. However, it is also important to consider the trade-off between these two properties, as increasing the sharpness may come at the cost of decreased calibration. Additionally, it should be noted that the uncertainty quantification metrics discussed are independent of accuracy metrics; a sharp and well calibrated model can still show poor predictive performance.

2.5.2. Continuous Ranked Probability Skill Score The Continuous Ranked Probability Score (CRPS) is a proper scoring rule, which can be used to evaluate the accuracy and reliability for a continuous variable where the model predicts a distribution, rather than a point estimate. It is a generalisation of the Mean Absolute Error (MAE) for distributional predictions. Given a predictive cumulative density function $F(x)$ and an observed value y, the CRPS can be computed as follows:

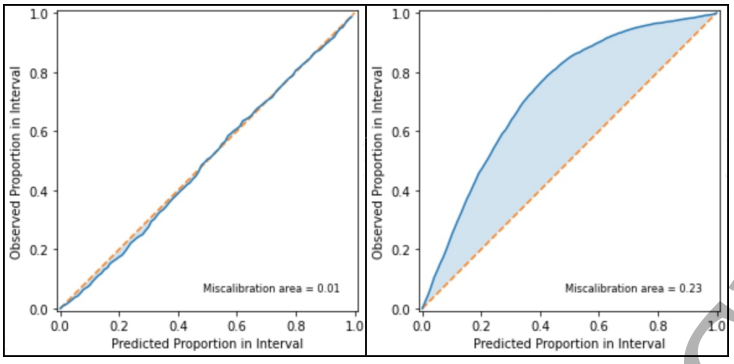
$$CRPS(F, y) = \int_{-\infty}^{\infty} \left(F(x) - \Theta(x - y)\right)^2 dx, \tag{14}$$

where Θ is the Heaviside function that denotes a step function along the real line. As a proper scoring rule this metric provides a summary assessment of accuracy, calibration, and sharpness. A lower value indicates a more accurate prediction and a more reliable measure of uncertainty.

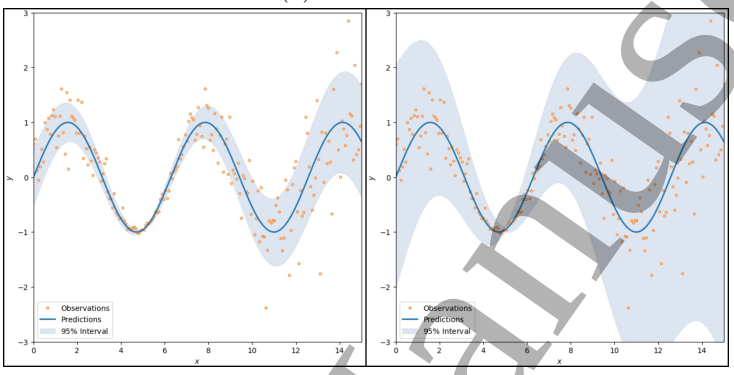
The skill score is used to calculate the performance of a model relative to some reference model. The skill score provides a benchmark for assessing the improvement or degradation in model performance while considering accuracy, calibration, and sharpness. This metric is calculated as

$$skill = \frac{CRPS_{ref} - CRPS_{target}}{CRPS_{ref}}, \tag{15}$$

where $CRPS_{ref}$ and $CRPS_{target}$ are the CRPS scores for the reference model and target model, respectively. A positive skill indicates that a model has performed better than the reference model whereas a negative skill score indicates that is has performed worse. The skill score is used to evaluate the performance of the proposed active learning strategies at each iteration of the adaptive sampling process.



(a) Calibration



(b) Sharpness

Figure 2: (a) shows calibration curves for a well-calibrated model (left) and a poorly calibrated model (right). In this case, the poorly calibrated model is underconfident as the observed proportion in an interval is greater than what is predicted. For an overconfident model, the curve would be below the ideal diagonal. (b) shows the 90% confidence intervals for a sharp model (left) and a dull model (right).

2.6. Improving Uncertainty Quantification

2.6.1. Post-Hoc Calibration Post-hoc calibration is the process of improving the uncertainty quantification of a model after it has been trained. There are various methods but in this research we focus on the state-of-the-art method known as *CRUDE* (Calibrating Regression Uncertainty Distributions Empirically) [23]. This method has shown leading performance, overcoming the trade-off between calibration and sharpness exhibited by other algorithms.

CRUDE is designed for probabilistic regression and as such it requires a model that returns a shift and scale value for a given input: $\mathcal{M}(\mathbf{x}) = (\mu(\mathbf{x}), \sigma(\mathbf{x}))$. Typically these predicted values will pertain to the mean and standard deviation for a Gaussian distribution but in the general case any assumptions of normality can be disregarded. It is assumed that each observed output is noisy with additive noise z which follows an unspecified error distribution $\mathcal{E}$. It is then the case that each observation y is a scaled and shifted version of this noise:

$$y = \mu + z \cdot \sigma, \text{ with } z \sim \mathcal{E}. \tag{16}$$

The main objective then is to estimate the distribution $\mathcal{E}$, with $\hat{\mathcal{E}}$. This is done by first using the trained model to predict the shift and scale value for each sample in a held-out calibration set $(\mathbf{X}_C, Y_C)$ to collect the z-scores. The empirical distribution $\hat{\mathcal{E}}$ is then constructed from the

collection of these z-scores as follows:

$$Z_C = \left\{ \frac{y - \mu(\mathbf{x})}{\sigma(\mathbf{x})} \middle| (\mathbf{x}, y) \in (\mathbf{X}_C, Y_C) \right\}, \tag{17}$$

$$\hat{\mathcal{E}} := \text{Empirical}(Z_C).$$

Following this, the predictive distribution on an unseen data point $\mathbf{x}_*$ is the estimated noise distribution scaled and shifted by the predicted values obtained from the model,

$$y_* | \mathbf{x}_* \sim \mu(\mathbf{x}_*) + \sigma(\mathbf{x}_*) \cdot \hat{\mathcal{E}}. \tag{18}$$

The calibrated variance is then given by

$$\mathbb{V}\left[y_* | \mathbf{x}_*\right] = \sigma^2(\mathbf{x}_*) \cdot \mathbb{V}\left[\hat{\mathcal{E}}\right], \tag{19}$$

with

$$\mathbb{V}\left[\hat{\mathcal{E}}\right] = \frac{1}{|Z_C|} \sum_{z_c \in Z_C} \left(z_c - \mathbb{E}\left[\hat{\mathcal{E}}\right] \right)^2, \tag{20}$$

$$\mathbb{E}\left[\hat{\mathcal{E}}\right] = \frac{1}{|Z_C|} \sum_{z_c \in Z_C} z_c, \tag{21}$$

where $|Z_C|$ is the number of elements in the set Z_C.

We propose *calibrated MEPE* in which the calibrated variance obtained from the CRUDE method is used in place of the predicted variance from the GP in equation 11.

2.6.2. Student-t Processes Student-t Processes (TPs) are a generalisation of GPs and indeed the two methods share many similarities. A TP is obtained by placing an *inverse Wishart process* prior over the kernel function and marginalising. A function with a Student-t Process prior is then denoted by

$$f(\mathbf{x}) \sim \mathcal{TP}(\nu, m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{22}$$

where ν is the degrees of freedom parameter, m is the mean function, and k is the kernel function. The parameter ν controls how heavy-tailed the process is. A small value corresponds to a heavy tail but the TP tends to a GP as $\nu \to \infty$.

Although both models are similar, regression must be treated differently in the case of a TP. For GP regression it is assumed that the output is the sum of the latent function and independent Gaussian noise. This is analytically tractable as the Gaussian distribution is closed under addition, however, this is not the case for noise sampled from a Student-t distribution. To overcome this, the output is simply modelled as $y = f(\mathbf{x})$, and the noise is incorporated into the kernel function, $k = k_\theta + \delta$. In doing this the noise is no longer independent as the degrees of freedom parameter scales both the parametric kernel and the noise kernel. Despite this, the TP with noise incorporated into the kernel behaves similarly to a TP with independent Student-t noise [26].

As with the GP, the hyperparameters can be estimated by maximising the negative log marginal likelihood. For a TP this now takes the form

$$-\ln p(\mathbf{y}|\nu, k) = \frac{1}{2} \ln |\mathbf{K}| + \frac{n}{2} \ln((\nu - 2)\pi) + \ln\left(\frac{\Gamma(\frac{\nu+n}{2})}{\Gamma(\frac{\nu}{2})} \right) + \left(\frac{\nu + n}{2} \right) \ln\left(1 + \frac{\beta}{\nu - 2} \right), \tag{23}$$

where $\beta = (\mathbf{y} - \mathbf{m})^\top \mathbf{K}_\theta^{-1} (\mathbf{y} - \mathbf{m})$ and $\Gamma(\cdot)$ is the Gamma function.

For making predictions on an unseen test point $\mathbf{x}_*$ in the case of TP regression, the posterior predictive density is given by

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \text{MVT}(\nu + n_1, \hat{\mu}(\mathbf{x}_*), \hat{\sigma}^2(\mathbf{x}_*)), \tag{24}$$

where MVT refers to a multivariate Student-t distribution, and n_1 is the size of the training set. The predictive mean and variance are given by

$$\hat{\mu}(\mathbf{x}_*) = \mathbf{k}_*^\top \mathbf{K}^{-1} \mathbf{y}, \tag{25}$$

$$\hat{\sigma}^2(\mathbf{x}_*) = \frac{\nu + \beta - 2}{\nu + n_1 - 2} \cdot (k_{**} - \mathbf{k}_*^\top \mathbf{\Sigma}^{-1} \mathbf{k}_*) \tag{26}$$

The main difference between the GP and TP is the predictive covariance, which is used to quantify the uncertainty. For TPs this covariance now depends on the training observations. Shah *et al.* [26] indicated that TPs are especially useful in situations where accurate predictive covariances are critical for good performance. In the regression experiments detailed in their paper, the TPs outperformed the GPs in terms of both predictive mean and uncertainty. It was evaluated that the TPs had all of the benefits of the GPs, but with increased modelling flexibility and no further computational cost. In fact, the authors concluded "that it could be useful to replace GPs with TPs in almost any application". This has been corroborated by [38] and [39] where TPs were shown to outperform GPs in both regression and Baysesian optimisation settings. This success was in part attributed to the ability of the TP to better handle outliers in the dataset.

We propose the method *TP-MEPE* in which the predicted mean and variance used in the calculation of the expected prediction error is obtained from a TP rather than a GP.

2.6.3. Calibrated Student-t Processes We combine TPs and CRUDE calibration to investigate whether this results in additional benefits. This involves using a TP to obtain the predicted mean and variance of the IQA energy for a given input and then performing CRUDE calibration on this predicted variance as described in equation 19. The calibrated TP variance is then used in place of the predicted variance in equation 11 to calculate the expected predicted error for each data point in the candidate pool at each step of the active learning process. We refer to this method as *calibrated TP-MEPE*.

3. Results

3.1. Uncertainty Quantification

We first investigate the uncertainty quantification of GPs and TPs both with and without CRUDE post-hoc recalibration. For both the GP and TP models, the accuracy and precision of the confidence intervals are measured while varying the size of the training set n_{train}, and the size of the calibration set n_{cal}. Specifically, the miscalibration area and sharpness are recorded against a test set of 4000 randomly selected points for each atom in the water dimer system. The size of the training set ranges between 300 and 3000, and the size of the calibration set ranges from 0 to 1000 where $n_{cal} = 0$ corresponds to a model that does not undergo recalibration. Each training set consists of points sampled at random. The size of the training set and calibration set are increased in intervals of 300 and 100, respectively. The GP and TP models were implemented using the GPyTorch Python package [43].

Figure 3 shows the mean miscalibration area as it varies with the size of the training set n_{train} for uncalibrated models. The miscalibration area is averaged over the hydrogen atoms, oxygen atoms, and all atoms in the water dimer system to provide an overview of the effect of training set size on the calibration of different groups of atoms.

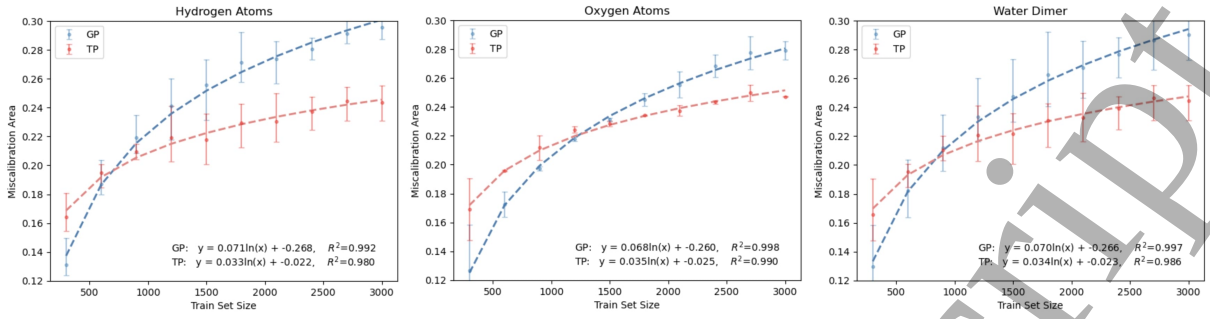
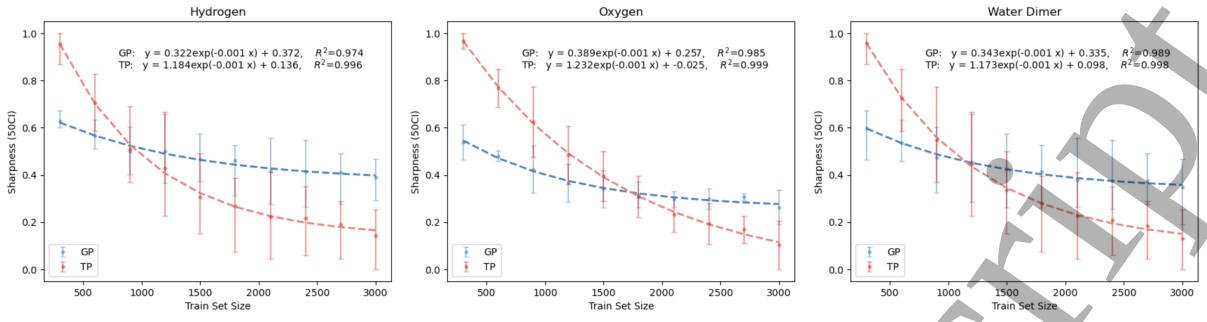
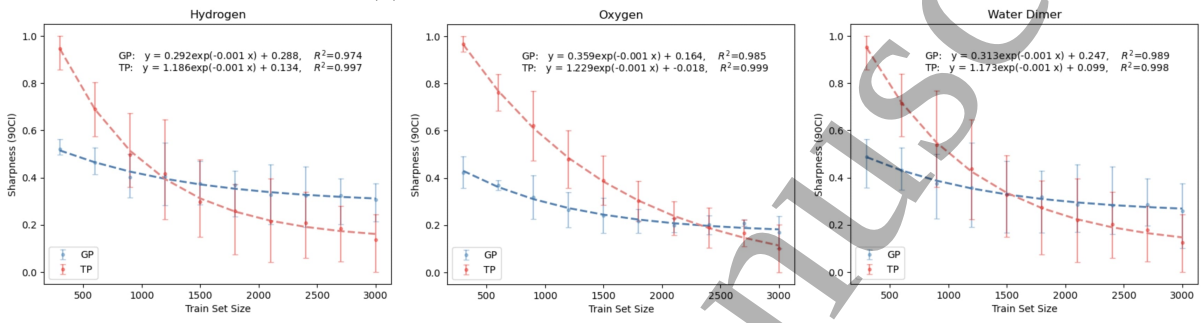


Figure 3: Mean miscalibration area against training set size for hydrogen atoms, oxygen atoms, and all atoms in the water dimer system using uncalibrated GP and TP models. The train set size ranges from $n_{train} = 300$ to $n_{train} = 3000$ with a step size of 300. The performance of each trained model is evaluated against a test set of 4000 random samples. The error bars show the minimum and maximum miscalibration area values for a given set of atoms.

Figure 4 shows the mean normalised sharpness of the 50% and 90% confidence intervals as a function of the size of the training set n_{train} for uncalibrated models. Similar to figure 3, the sharpness is averaged over the hydrogen atoms, oxygen atoms, and all atoms in the water dimer system. Unlike for miscalibration area, the scale of the sharpness differs for each type of atom therefore it is necessary to normalise the sharpness values for hydrogen atoms and oxygen atoms separately to lie between 0 and 1 before taking the mean.



(a) Sharpness of 50% confidence interval



(b) Sharpness of 90% confidence interval

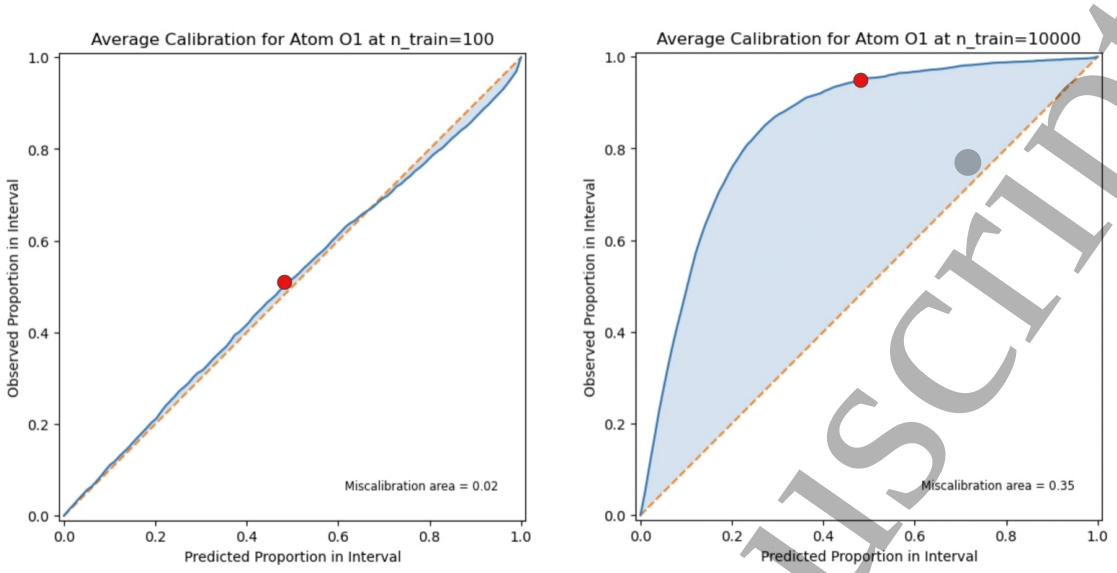
Figure 4: Mean normalised sharpness against training set size for hydrogen atoms, oxygen atoms, and all atoms in the water dimer system using uncalibrated GP and TP models. Sharpness is shown as measured by the average width of (a) the 50% confidence interval and (b) the 90% confidence interval. The train set size ranges from $n_{train} = 300$ to $n_{train} = 3000$ with a step size of 300. The performance of each trained model is evaluated against a test set of 4000 random samples. The error bars show the minimum and maximum normalised sharpness values for a given set of atoms.

We first consider the uncertainty quantification of uncalibrated models. As seen in figure 3 and figure 4, the miscalibration area grows logarithmically with the size of the training set whereas the sharpness decays exponentially. It appears that a model becomes increasingly underconfident as the training set grows, hence the increase in miscalibration area. This is a surprising result as it has been shown that a well-specified model should be calibrated in the limit where the training set size is much larger than the number of features [41]. Nonetheless, similar behaviour has been reported with GP-deep neural network hybrids [40], and overparameterised deep neural networks [42] in which calibration error increases with training set size and training time, respectively.

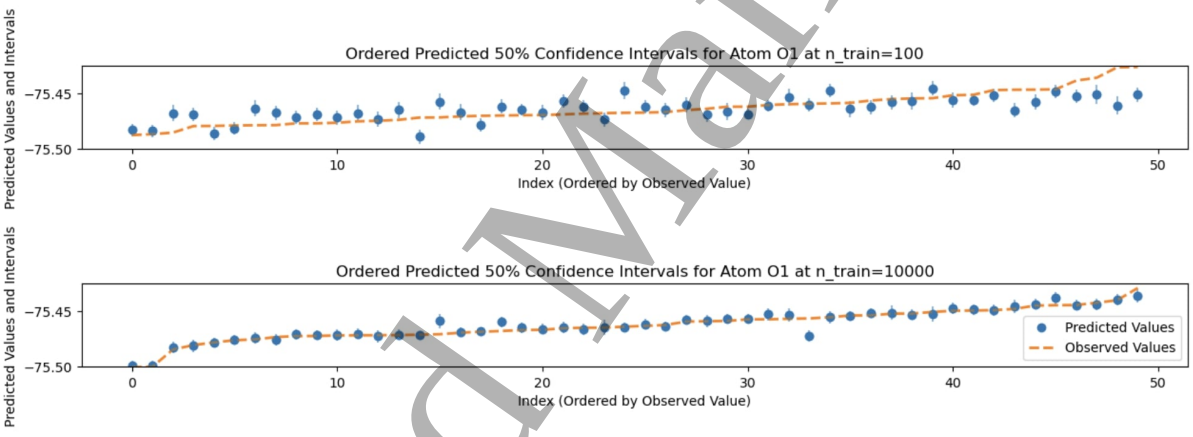
In the context of classification it has been shown that there is an intermediate range where calibration error increase when the number of features is comparable to the number of samples [44]. Our results indicate that in contrast to these reports, calibration increases monotonically with training set size even when there are 1000 times more training samples than features. This behaviour is not a result of implementation or kernel selection as it has been replicated using the GPy package, and using both RBF and Matern kernels in the GPyTorch package.

Figure 5b and figure 5a illustrate how the accuracy, calibration and sharpness evolve from a small training set of $n_{train} = 100$ to a large training set of $n_{train} = 10,000$ for a Gaussian Process trained on the O1 atom dataset. Figure 5b shows the ordered prediction intervals for a subset of 50 samples in the test set, and figure 5a shows the calibration curve obtained using a test set of

4,000 samples. These results are representative of both GP and TP models trained on any of the atoms in the water dimer system. As the size of the training set increases, the predicted mean values approach the true observed values and the predicted variances decrease. As a result, both the accuracy and sharpness improve. As the accuracy improves, an increasing proportion of the true values fall within the confidence intervals as the sharpness does not improve sufficiently fast. Therefore, the model becomes more underconfident as it ingests more training data, seemingly due to the fact that the width of the confidence intervals does not decrease quickly enough. When fewer training data are available the GP produces better calibrated and sharper confidence intervals compared to the TP. With larger training sets, however, the TP outperforms in terms of both calibration and sharpness.



(a) Calibration Curves



(b) Ordered Prediction Intervals

Figure 5: An uncalibrated Gaussian Process was trained on the O1 atom dataset using a training set of size $n_{train} = 100$ and $n_{train} = 10,000$ to obtain (a) calibration curves, and (b) ordered prediction intervals for a subset of 50 points in a test set. The predicted intervals show the sharpness of the 50% confidence interval. The test set consists of $4,000$ randomly selected samples. The red marker in (a) indicates the point that corresponds to the ordered prediction intervals in (b).

Considering the entire water dimer system we can show analytically that the rate of change of the average miscalibration area for the GP is 2.06 times that of the TP. While these models become less calibrated with the addition of more training data, the TP scales better with the size of the training set. In contrast, the sharpness improves with the addition of more training data but it does so at a faster rate for the TP. In summary, the uncertainty quantification of the TP improves at a faster rate than that of the GP. Additionally, while the general trend remains the same for all atoms in this system, more training data is required for the oxygen atoms before the TP has the advantage.

Figure 6 shows the miscalibration area and sharpness averaged over every atom in the

water dimer system for both the uncalibrated GP and uncalibrated TP. There appears to be competition between calibration and sharpness, given that the miscalibration area increases with the size of the training set whereas sharpness decreases. This supports the idea of a trade-off between the two attributes as described in the literature [45, 46]. The trade-off between calibration and sharpness is more severe for the TP. As illustrated in figure 6, the same decrease in miscalibration area will result in a much greater increase in the sharpness compared to the GP because the slopes of the TP are larger than those of the GP. This suggests that it may be more difficult to balance the accuracy and the precision of confidence intervals for this proposed TP alternative.

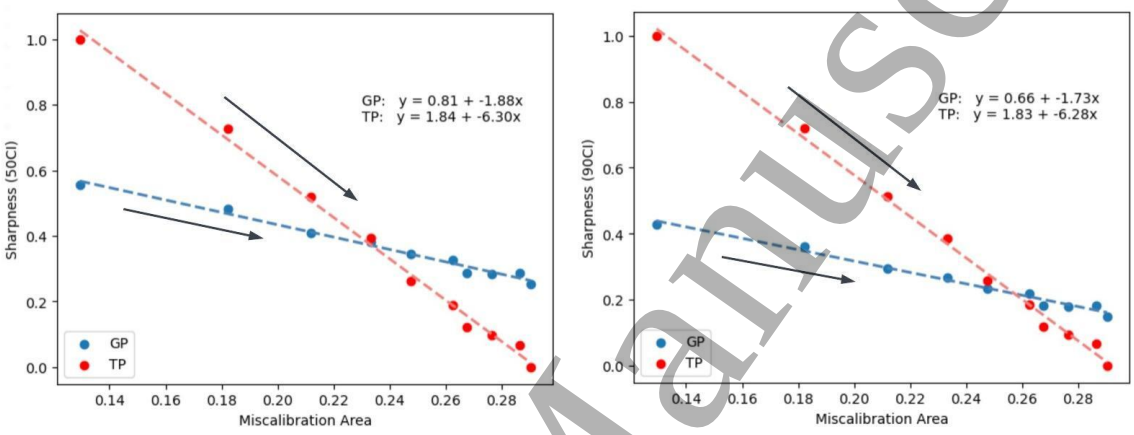


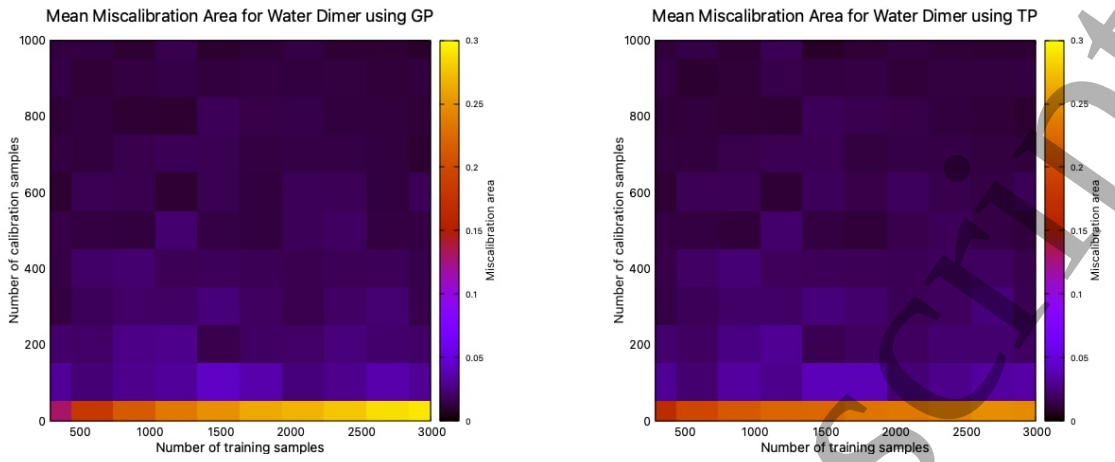
Figure 6: Mean miscalibration area against normalised mean sharpness for uncalibrated GP and TP models on the water dimer system. The train set size ranges from $n_{train} = 300$ to $n_{train} = 3000$ with a step size of 300. The arrows point in the direction of increasing n_{train}. The performance of each trained model is evaluated against a test set of 4000 random samples.

Figure 7 shows heatmaps that describe how the mean miscalibration area and normalised mean sharpness vary with both the size of the training set n_{train}, and the size of the calibration set n_{cal}. Here, the sharpness is measured as the average width of the 90% confidence interval.

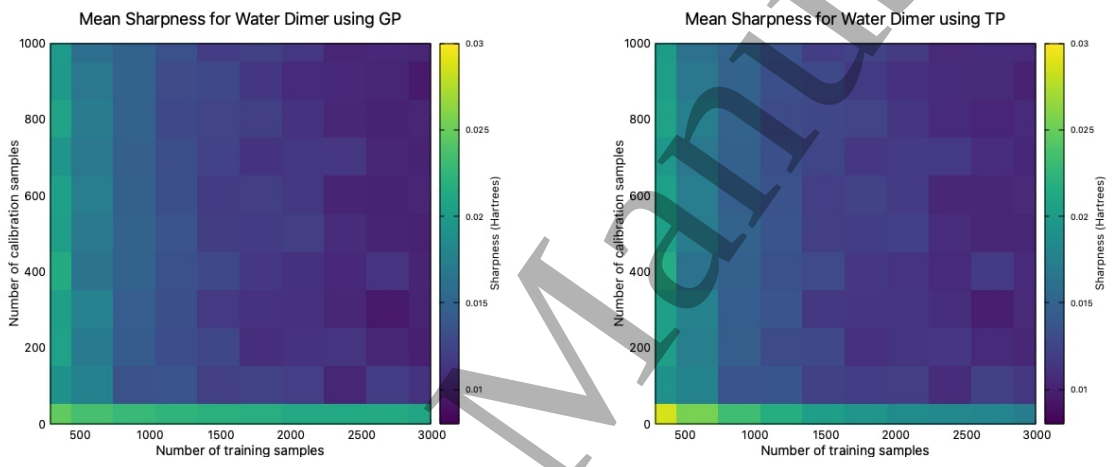
Now considering post-hoc calibration, it is clear from the figure 7 that the CRUDE method has a significant and positive impact on the uncertainty quantification, however, calibration and sharpness are affected differently. Pearson correlation coefficients were calculated to assess the linear relationship between the uncertainty quantification metrics and the size of the train and calibration sets for recalibrated models. The results can be found in table 1 and table 2. From these tables it is clear that both GPs and TPs are affected in the same way such that their results are nearly identical following recalibration.

GP	n_{train}	n_{cal}
Miscalibration Area	-0.004	-0.770
Sharpness	-0.903	-0.020

Table 1: Pearson correlation coefficient for the calibrated Gaussian Process models.



(a) Calibration



(b) Sharpness

Figure 7: (a) Average miscalibration area and (b) average sharpness for all atoms in water dimer system for different values of n_{train} and n_{cal}. The train set size ranges from $n_{train} = 300$ to $n_{train} = 3000$ with a step size of 300. The calibration set ranges from $n_{cal} = 0$ to $n_{cal} = 1000$ with a step size of 100. A value of $n_{cal} = 0$ corresponds to a model that is uncalibrated. The performance of each trained model is evaluated against a test set of 4000 random samples.

TP	n_{train}	n_{cal}
Miscalibration Area	-0.009	-0.771
Sharpness	-0.894	-0.013

Table 2: Pearson correlation coefficient for the calibrated Student-t Process models.

Considering calibration first, we can see that using even the smallest calibration set with $n_{cal} = 100$ reduced the miscalibration area to less than 0.05. The miscalibration area is further reduced as the size of the calibration set increases, such that both GP and TP models show near perfect calibration with a miscalibration area between $0.0080 - 0.0161$ using the largest calibration set of $n_{cal} = 1000$. Indeed, we can see there is a strong negative correlation between miscalibration area and n_{cal} in table 1 and table 2. The Pearson correlation coefficient for miscalibration area and n_{train} is extremely small, indicating that the size of the training set has

no substantial impact upon the calibration of a model following the application of the CRUDE method. This is despite the fact that uncalibrated models show a distinct relationship between miscalibration area and the size of the training set.

CRUDE also improves sharpness but it has less of an impact when a model is trained with fewer data. Sharpness is strongly and negatively correlated to n_{train} as can be seen in table 1 and table 2. Notably, the Pearson correlation coefficient pertaining to sharpness and n_{cal} is small enough to suggest very little to no linear dependence at all. This is in contrast to the results for miscalibration area, and implies that accurate and precise uncertainty estimates require a sufficiently large training set as well as a calibration set.

3.2. Active Learning

Following the analysis of the uncertainty quantification, we analyse the performance of the following active learning strategies:

(i) GP with random sampling (GP) as a baseline;
(ii) GP with MEPE sampling (GP-MEPE);
(iii) GP with calibrated MEPE sampling (GP-MEPE-cal);
(iv) TP with random sampling (TP);
(v) TP with MEPE sampling (TP-MEPE);
(vi) TP with calibrated MEPE sampling (TP-MEPE-cal).

This analysis is performed by measuring the CRPS on an unseen test set at every iteration of the active learning process for each atom in the water dimer system. The test set contains 4000 samples, and the new training samples are selected from an unlabelled candidate pool of $10,000$ points. When a new point is selected, the true value is generated and the labelled sample is added to the training set. When initialising a training set for active learning it is advisable to opt for a space filling method. In the FFLUX force field, the min-max-mean method is used to select the initial data points [17], and to allow for comparison that method is implemented here. With this approach, the sample with the minimum, maximum, and mean value for each feature is selected from the larger unlabelled candidate pool. This results in an initial training set size of $n_{train} = 36$.

We first perform single sample active learning where the training set is increased from $n_{train} = 36$ to $n_{train} = 1000$. Following this, the impact of batch sampling is investigated. In this case the training set is increased from $n_{train} = 36$ to $n_{train} = 2036$ in batches of 10 samples. A calibration set of 3000 randomly selected points is used for the strategies that utilise the CRUDE method. For each learning strategy, the active learning process was repeated 10 times.

Figure 8 and figure 9 represent the mean skill score over 10 experiments at every iteration of the learning process for single sample and batch sample active learning, respectively. In these experiments the GP random sampling strategy is the reference model such that the skill scores indicate improvement or degradation relative to this baseline. The shaded region represents one standard deviation in the skill score based on the results from the 10 experiments.

We first consider the GP-MEPE strategy. From figure 8 it is clear that this strategy did not perform well in the single sample active learning experiments. For all atoms in the water dimer system there is an immediate sharp decline in skill score from which this strategy cannot recover. For all atoms except O1 the skill score drops to between -0.15 and -0.35 after 100 iterations, meaning these strategies are performing $15 - 35\%$ worse than the GP baseline strategy. For the hydrogen atoms, the performance starts to increase after the sharp decline but no strategy is able to recover from the initial degradation in performance.

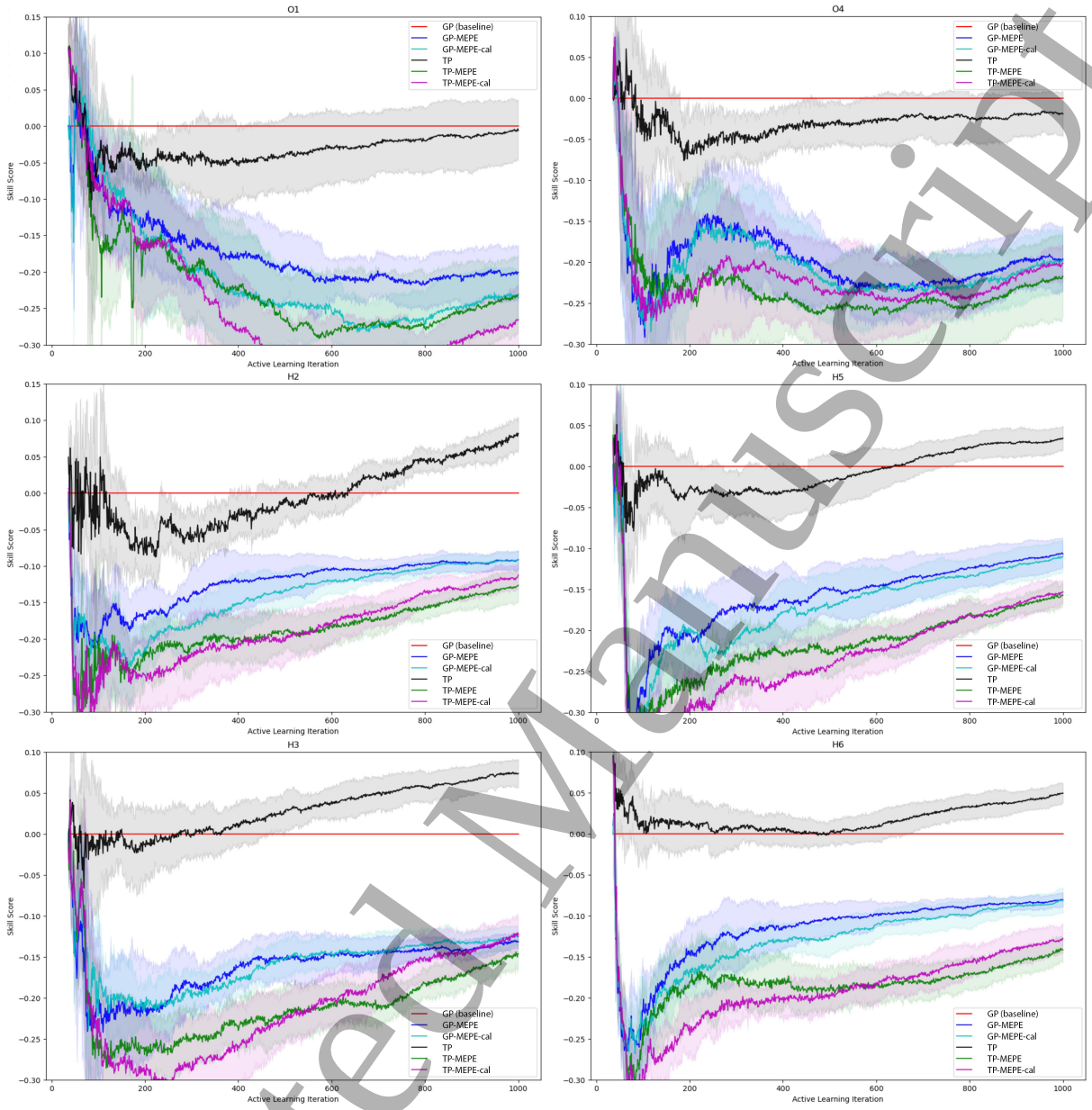


Figure 8: Mean skill score of single sample learning strategies for each atom in the water dimer system. The shaded region represents one standard deviation of the skill score based on results from 10 experiments. A calibration set of 3000 random samples is used for calibrated models. The performance of each trained model is evaluated against a test set of 4000 random samples.

These results are echoed with batch sampling, as seen in figure 9. With the hydrogen atoms there is still a sharp decline followed by a steady increase in skill score. The difference here is that the performance of the GP-MEPE strategy is seen to converge to that of the GP baseline strategy such that the skill score for each hydrogen atom is less than or equal to 0.001 by the end of the learning process. A disparity remains for the oxygen atoms, but it may be that convergence would similarly occur for larger n_{train}.

Substantial differences can be seen when considering the two passive learning methods; the GP and TP strategies that use randomly sampled training sets. With single sample selection

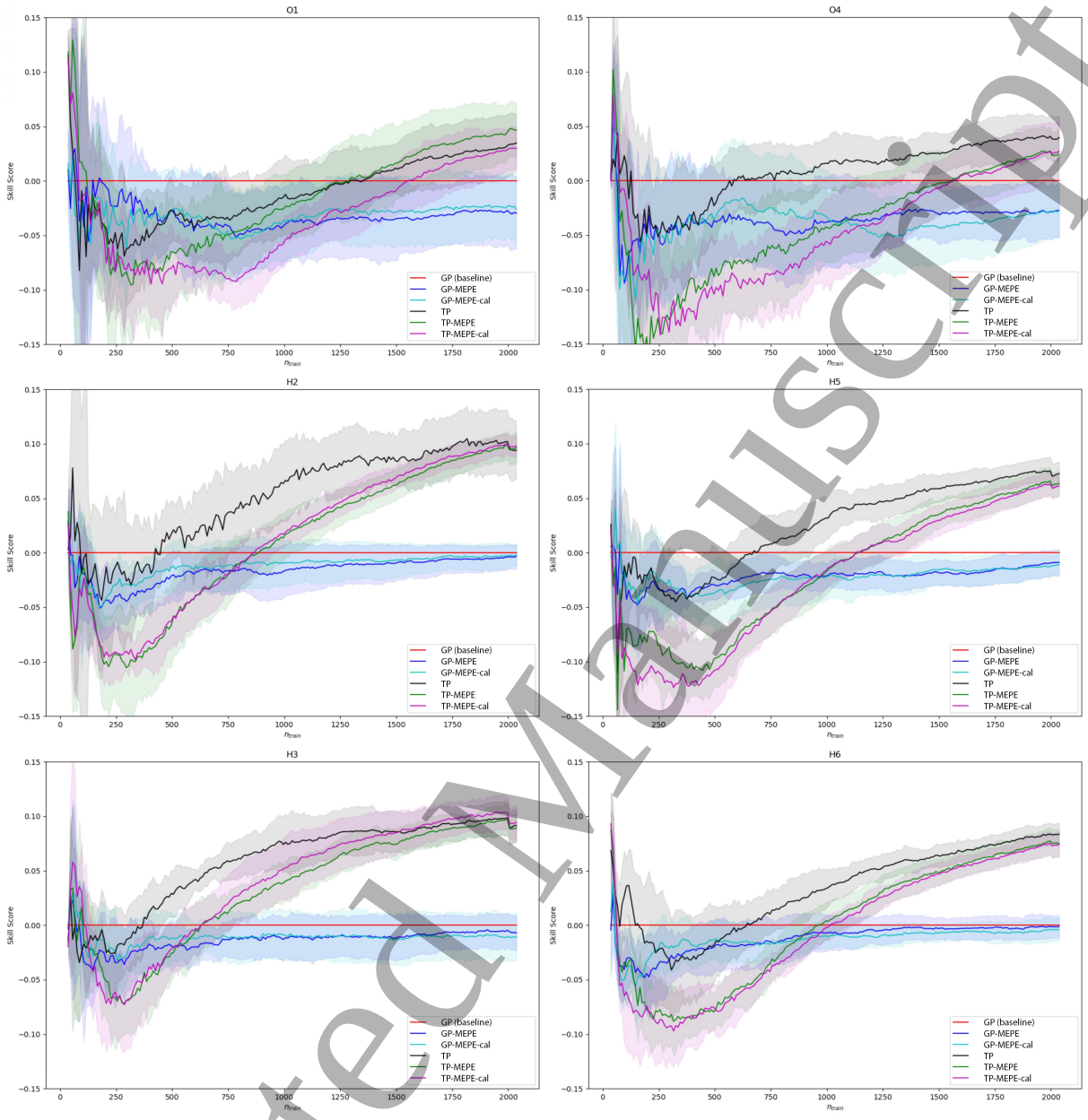


Figure 9: Mean skill score of batch sample learning strategies for each atom in the water dimer system. The shaded region represents one standard deviation of the skill score based on results from 10 experiments. A calibration set of 3000 random samples is used for calibrated models. The performance of each trained model is evaluated against a test set of 4000 random samples.

(figure 8) on the oxygen atoms the TP performs slightly worse than the GP with an average skill score of -0.03, however performance converges as the training set increases. For hydrogen atoms the GP typically performs better with fewer data but as the training set grows the TP starts to outperform the GP. By the end of the learning process the TP showed an improvement of $5-8\%$ over the GP baseline.

This trend continues with larger training sets as seen in figure 9 for the batch sampling experiments. As the size of the training set grows further so does the disparity between the performance of the GP and TP methods, such that the TP is the dominant strategy across all atoms when the training set is sufficiently large. With the largest training set of size $n_{train} = 2034$ the TP shows an average improvement of 3.75% for oxygen atoms and 8.55% for hydrogen atoms when compared to the GP baseline. The comparative performance increases logarithmically such that there will be diminishing returns with further increases in the size of the training set.

Referring to figure 8, TP-MEPE was one of the worst performing strategies across all atoms in the water dimer system. The behaviour is similar to that of GP-MEPE in that there is typically an immediate sharp decline followed by an increase in performance, but the skill score is $10-20\%$ lower than that of GP-MEPE across all atoms except O4 for $n_{train} > 200$. Interestingly, even when TP outperforms GP, TP-MEPE does not outperform GP-MEPE.

As seen in figure 9, TP-MEPE improves drastically in the batch sample setting. While performance is still extremely poor for training sets with less than 1000 points, it converges to that of the TP when more training data is available.

CRUDE calibration does not appear to improve the performance of GP-MEPE or TP-MEPE. In fact, the performance of uncalibrated and calibrated models is nearly identical with the skill score differing by no more than 0.03 across all atoms in the water dimer system for both single sample active learning (figure 8) and batch sample active learning (figure 9). Consistently, the worst performing strategies were TP-MEPE and TP-MEPE-cal, with the two being effectively equivalent. MEPE had more of a detrimental effect on the TP despite the fact that this model outperformed the GP with random sampling in many circumstances.

4. Discussion

The uncertainty of a Student-t Process is better calibrated and sharper than that of a Gaussian Process if there are sufficient training data. This is because the Student-t Process scales better with the size of the training set, in that the miscalibration area increases more slowly and the sharpness decreases more quickly. Unfortunately, in the lower data limit the uncertainty estimates become much more unreliable. The difference in performance between the Student-t Process and Gaussian Process may be due to the difference in what is considered signal and what is considered noise by the two models (a tail value considered an outlier by a Gaussian Process may not be considered an outlier by a Student-t Process).

The CRUDE method results in near perfect calibration with even a small calibration set of size $n_{cal} = 100$. While this method does also improve sharpness, this effect is weaker. It appears that if a model is at all calibrated then increasing the size of the training set is the most effective way to reduce sharpness.

While both methods have strengths, it is clear that the CRUDE algorithm is superior in the context of optimal uncertainty quantification. Returning to aim (1), we conclude that the Student-t Process is not a viable alternative to post-hoc calibration in producing reliable uncertainty estimates.

Despite the fact that the CRUDE method significantly improved the uncertainty estimates, the calibrated MEPE strategies did not substantially differ in performance compared to the standard MEPE strategies. In reference to aim (2), this implies that improving the uncertainty estimates used in the MEPE algorithm does not improve active learning.

It may be that CRUDE calibration effectively scales the predicted variance of every element

in the sample pool by the same constant. Therefore, when the balance factor α is small and the MEPE algorithm prioritises exploration, the same samples will be selected as having the maximum expected prediction error. When α is larger, the MEPE algorithm prioritises exploitation such that the variance, calibrated or not, is irrelevant. It is only when the exploration and exploitation terms are roughly balanced that there will be a discernible difference between the calibrated and uncalibrated strategies. This will rarely happen in practice given that the MEPE algorithm typically prioritises exploration in the beginning and prefers exploitation in the later stages.

The active learning strategies tested could not consistently outperform against the Gaussian Process random sampling baseline. In fact, these methods consistently show a significant drop in performance at the beginning of the active learning process before showing improvement. This may be a result of the small initial training set. With an initial size of $n_{train} = 36$ the underlying model may not have enough information to accurately predict the uncertainty which then results in poor selection of samples to be added to the training set. These samples may poison the training set such that the model cannot recover after their addition.

Considering the random sampling strategies, when fewer training data are available the Gaussian Process performs better. With larger training sets, however, the Student-t Process is superior as measured by the CRPS. This is a surprising result for two reasons. First, the posterior of a Student-t Process is a Student-t distribution, which is more flexible with fewer data compared to a normal distribution. Furthermore, it has been shown mathematically, that a Student-t Process tends to a Gaussian Process as $n_{train} \to \infty$ [26]. As such, it was expected that the Student-t Process would show improved predictive performance in the low data limit and that the models would converge as the training set was increased.

Regarding aim (3), replacing the Gaussian Process with a Student-t Process improves the predictive performance on a water dimer system when trained on a sufficiently large, randomly sampled training set. Further research is required to confirm whether these findings hold on larger systems such as Glycine.

5. Conclusion

In this research, we successfully implemented a Student-t Process and the CRUDE post-hoc calibration method in an attempt to improve MEPE active learning in the FFLUX force field. Our results indicate that recalibrating uncertainty is not sufficient to overcome the effectiveness of random sampling using a Gaussian Process. Nonetheless, we identified the uncalibrated Student-t Process with random sampling as an avenue to overcome this problem. When using larger labelled datasets composed of randomly sampled examples, the Student-t Process is the leading strategy for predicting the IQA energy for both hydrogen and oxygen atoms in the water dimer system. Developing this method further could result in leading performance in a wider range of problems.

6. Acknowledgements

We are grateful for the use of the computing resources from the Northern Ireland High Performance Computing (NI-HPC) service. We are also grateful to Yulian Manchev for providing the dataset. P.L.A.P is grateful to the European Research Council (ERC) for the award of an Advanced Grant underwritten by the UKRI-funded Frontier Research grant EP/XO24393/1.

References

[1] Xu, P, Guidez, E B, Bertoni, C and Gordon, M S 2018 Perspective: Ab Initio Force Field Methods Derived from Quantum Mechanics *J. Chem. Phys.* **148** (9): 090901

[2] Cardamone, S, Hughes, T J and Popelier, P L A 2014 Multipolar Electrostatics *Phys. Chem. Chem. Phys.* **16** 10367

[3] Unke, O T, Chmiela, S, Sauceda, H E, Gastegger, M, Poltavsky, I, Schütt, K T, Tkatchenko, A and Müller, K-R 2021 Machine Learning Force Fields *Chem. Rev.* **121** (16) 10142–86

[4] Di Pasquale, N, Davie, S J and Popelier, P L A 2018 The accuracy of ab initio calculations without ab initio calculations for charged systems: Kriging predictions of atomistic properties for ions in aqueous solutions *J. Chem. Phys.* **148** 241724

[5] Burn, M J and Popelier, P L A 2023 FEREBUS: A High-Performance Modern Gaussian Process Regression Engine *Digital Discovery*, **2** 152

[6] Ghorbani, B, Mei, S, Misiakiewicz, T, Montanari, A 2021 'When do neural networks outperform kernel methods?' *Journal of Statistical Mechanics: Theory and Experiment* **12** p. 124009

[7] Kamath, A, Vargas-Hernández, R A, Krems, R V, Carrington Jr, T and Manzhos, S 2018 Neural Networks vs Gaussian Process Regression for Representing Potential Energy Surfaces: A Comparative Study of Fit Quality and Vibrational Spectrum Accuracy *J. Chem. Phys.* **148** (24) 241702

[8] Miksch, A M, Morawietz, T, Kästner, J, Urban, A and Artrith, N 2021 Strategies for the construction of machine-learning potentials for accurate and efficient atomic-scale simulations. *Mach. Learn.: Sci. Technol.* **2**(3), 031001

[9] Lin, Q, Zhang, L, Zhang, Y and Jiang, B 2021 Searching configurations in uncertainty space: Active learning of high-dimensional neural network reactive potentials. *J. Chem. Theory Comput.* **17**(5), 2691-2701

[10] Schran, C, Brezina, K and Marsalek, O 2020 Committee neural network potentials control generalization errors and enable active learning. *J. Chem. Phys.* **153**(10)

[11] Uteva, E, Graham, R S, Wilkinson, R D and Wheatley, R J 2018 Active learning in Gaussian process interpolation of potential energy surfaces. *J. Chem. Phys.*, **149** (17)

[12] Guan, Y, Yang, S and Zhang, D H 2018 Construction of reactive potential energy surfaces with Gaussian process regression: active data selection. *Molecular Physics*, 116(7-8), 823-834

[13] Vandermause, J, Torrisi, S B, Batzner, S, Xie, Y, Kolpak, A M, Kozinsky, B 2020 On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events. *NPJ Comput. Mater.* 6:20

[14] Liu, H, Cai, J and Yong, Y-S 2017 An Adaptive Sampling Approach for Kriging Metamodeling by Maximizing Expected Prediction Error *Comput. Chem. Eng.* **106** 171–82

[15] Fuhg, J N, Amélie F and Nackenhorst, U 2021 State-of-the-Art and Comparative Review of Adaptive Sampling Methods for Kriging *Arch. Comput. Methods Eng.* **28** (4) 2689–2747

[16] Burn, M J and Popelier, P L A 2020 Creating Gaussian Process Regression Models for Molecular Simulations Using Adaptive Sampling *J. Chem. Phys.* **153** (5) 054111

[17] Burn, M J and Popelier, P L A 2022 Producing Chemically Accurate Atomic Gaussian Process Regression Models by Active Learning for Molecular Simulation *J. Comput. Chem.* **43** (31) 2084–98

[18] Sivaraman, G and Jackson, N E 2022 Coarse-Grained Density Functional Theory Predictions via Deep Kernel Learning *J. Chem. Theory Comput.* **18** (2) 1129–41

[19] Burbidge, R., Rowland, J.J. and King, R.D. 2007 Active Learning for Regression Based on Query by Committee *Springer eBooks* pp.209–218

[20] Bemporad, A. 2023 Active learning for regression by inverse distance weighting *Information Sciences*, **626** pp.275–292

[21] Schein, A.I. and Ungar, L.H. 2007 Active Learning for Logistic regression: an Evaluation. *Machine Learning* **68** (3) pp.235–265

[22] Laves, M-H, Ihler, S, Fast, J F, Kahrs, Lüder A and Ortmaier, T 2020 Well-Calibrated Regression Uncertainty in Medical Imaging with Deep Learning *Proceedings of the Third Conference on Medical Imaging with Deep Learning* 393–412

[23] Zelikman, E, Healy, C, Zhou, S and Avati, A 2020 arXiv:2005.12496

[24] Kuleshov, V, Fenner, N and Ermon, S 2018 arXiv:1807.00263

[25] Foldager, J, Jordahn, M, Hansen, L K, Andersen, M R 2023 arXiv:2301.05983

[26] Shah, A, Wilson, A G and Ghahramani, Z 2014 Student-t Processes as Alternatives to Gaussian Processes *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*

[27] Bader, R F W 1990 *Atoms in Molecules: A Quantum Theory* (Oxford: OUP)

[28] Popelier, P L A 2018 On topological atoms and bonds *Intermolecular Interactions in molecular Crystals* ed J Novoa (Cambridge: RSC) pp 147-177

[29] Popelier, P L A 2022 Non-covalent interactions from a Quantum Chemical Topology perspective *J. Mol. Model* **28** 276

[30] Blanco, M A, Martín Pendás, A and Francisco, E 2005 Interacting Quantum Atoms: A Correlated Energy Decomposition Scheme Based on the Quantum Theory of Atoms in Molecules *J. Chem. Theory Comput.* **1** (6) 1096-1109

[31] Mills, M J L and Popelier, P L A 2014 Electrostatic Forces: Formulas for the First Derivatives of a Polarizable, Anisotropic Electrostatic Potential Energy Function Based on Machine Learning *J. Chem. Theory Comput.*

10 (9) 3840–56
[32] Soper, A. K. *Chemical Physics* 2000 **258**(2-3) 121-137
[33] Rasmussen, C E and Williams, C K I 2005 *Gaussian Processes for Machine Learning* (Cambridge, MA: MIT Press)
[34] Murphy, K P 2022 *Probabilistic Machine Learning: An Introduction* (Cambridge, MA: MIT Press)
[35] Settles, B 2009 Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.
[36] Sundararajan, S and Keerthi, S S 2001 Predictive Approaches for Choosing Hyperparameters in Gaussian Processes *Neural Computation* **13** (5) 1103–18
[37] Tran, K, Neiswanger, W, Yoon, J, Zhang, Q, Xing, E and Ulissi, Z W 2020 Methods for Comparing Uncertainty Quantifications for Material Property Predictions *Mach. Learn.: Sci. Technol.* **1** (2) 025006
[38] Tang, Q, Niu, L, Wang, Y, Dai, T, An, W, Cai, J and Xia, S-T 2017 Student-t Process Regression with Student-t Likelihood *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence* 2822-2828
[39] Tracey, B D and Wolpert, D H 2018 Upgrading from Gaussian Processes to Student's-T Processes *AIAA Non-Deterministic Approaches Conference*
[40] Tran, G, Bonilla, E, Cunningham, J, Michiardi, P and Filippone, M 2019 Calibrating Deep Convolutional Gaussian Processes *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, PMLR* 89:1554-1563.
[41] Clarté, L, Loureiro, B, Krzakala, F, Zdeborová, L Theoretical characterization of uncertainty in high-dimensional linear classification *Machine Learning: Science and Technology* 4025029.
[42] Carrell, AM, Mallinar, N, Lucas, J and Nakkiran, P 2022) The Calibration Generalization Gap. arXiv:2210.01964
[43] Gardner, J R, Pleiss, G, Bindel D, Weinberger K Q and Wilson, A G 2018 GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration *Advances in neural information processing systems* **31** 7587–7597.
[44] Bai, Y, Song, M, Wang, H and Xiong, C 2021 Don't Just Blame Over-parametrization for Over-confidence: Theoretical Analysis of Calibration in Binary Classification *Proceedings of the 38th International Conference on Machine Learning, PMLR139:566-576.*
[45] Murphy, A H 1973 A New Vector Partition of the Probability Score *J. Appl. Meteorol.* **12** (4) 595–600
[46] Gneiting, T and Raftery, A E 2007 Strictly proper scoring rules, prediction, and estimation *JASA* **102** (477) 359–378
[47] Settles, B 2010 From Theories to Queries: Active Learning in Practice *Active Learning and Experimental Design workshop in conjunction with AISTATS 2010*