NASA/TM–20230006753

Efficient Calibration of Expensive Computational Models

Patrick E. Leser
Durability, Damage Tolerance, and Reliability Branch
Langley Research Center, Hampton, VA

Joshua M. Fody
Structural Mechanics and Concepts Branch
Langley Research Center, Hampton, VA

May 2023

NASA STI Program. . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collection of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI Program, see the following:

- Access the NASA STI program home page at `http://www.sti.nasa.gov`

- E-mail your question to `help@sti.nasa.gov`

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM–20230006753

Efficient Calibration of Expensive Computational Models

Patrick E. Leser
Durability, Damage Tolerance, and Reliability Branch
Langley Research Center, Hampton, VA

Joshua M. Fody
Structural Mechanics and Concepts Branch
Langley Research Center, Hampton, VA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

May 2023

Acknowledgments

This work was supported by the NASA Aeronautics Research Mission Directorate, through the Transformational Tools and Technologies Project.

Available from:

NASA STI Program / Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
Fax: 757-864-6500

Abstract

Accounting for uncertainty when calibrating expensive computational models is a common challenge faced by scientists and engineers. Often Bayesian techniques are adopted to estimate a probability density function over the model parameters given noisy empirical data. The methods used to perform this type of probabilistic calibration are computationally prohibitive in that they require a large number of evaluations of the expensive model. In these cases, surrogate modeling – that is, using a fast-to-evaluate, lower fidelity stand-in for the original computational model – may be the only option to alleviate this computational burden. However, the upfront cost of generating training data to build a surrogate model can itself be expensive. As such, it is important to be judicious when selecting training points at which the full-fidelity model is evaluated. Here, an active learning approach is proposed that enables efficient selection of training points using approximate samples of the calibrated parameter probability density function. In this way, the training points can be concentrated in regions where the calibration algorithm requires high model accuracy.

1 Introduction

Computational models are becoming an essential part of science and engineering. Often, these models require calibration to empirical data to be useful. While deterministic methods for correlating model response to available data are commonly used for this purpose, these methods neglect uncertainty due to natural variability, measurement noise, limited data, missing physics, numerical errors, etc. In contrast, Bayesian techniques for model calibration allow for quantification of these uncertainties by assigning probability density functions (PDFs) to the model parameters and estimating measurement noise levels [1]. Bayesian methods are particularly well-suited for this task as the resulting PDFs can be easily propagated through the model to predict corresponding PDFs of any number of downstream quantities of interest using methods such as Monte Carlo simulation.

The gain in information from probabilistic calibration is not free; it comes with added computational expense. Sampling methods for solving the calibration problem such as Markov chain Monte Carlo (MCMC) [1, 2] require a large number of model evaluations, which is particularly problematic when the evaluations involve computationally intensive simulations such as finite element analysis. Utilizing machine learning algorithms to develop quick-to-evaluate surrogates for these expensive models is a common approach for alleviating the computational burden. The surrogate is trained with data from the full-fidelity model, incurring some up front costs to generate the data, and then the surrogate is used during the computationally expensive Bayesian calibration. However, developing enough training data to achieve acceptable surrogate accuracy can be challenging, and, when the model is expensive, space-filling techniques such as Latin hypercube sampling (LHS) [3] are often inefficient uses of computational resources. This inefficiency is due to the fact that, in a supervised learning scheme, the model must be evaluated once per training point to generate the associated label. Additional complications arise when the high-fidelity model behaves poorly (e.g., inconsistent outputs or solver divergence) outside of some unknown subset of the calibration parameter space or when the parameter space is large with respect to the support of the calibrated PDF. Judicious selection of training data is even more important under these circumstances.

Active learning [4] is a machine learning field akin to optimal experimental design that enables efficient selection of training data in a sequential manner to improve training efficiency. Each new training point is selected in a sequential manner to maximize information gain. This approach is in contrast to pre-specifying a large number of training simulations as is done with LHS, for example. The choice of active learning algorithm depends on the intended use of the resulting surrogate (e.g., active learning methods for targeting failure thresholds [5, 6] or Bayesian optimization of a function [7]). Rather than targeting a failure threshold or finding a single parameter value via optimization, this work focuses on obtaining the solution to a probabilistic calibration problem that characterizes the joint probability distribution of model parameters conditioned on a set of calibration data, referred to as the parameter posterior distribution. Therefore, the goal of active learning in this context is to select training points that improve the surrogate's ability to make accurate and rapid predictions specifically in the region of the parameter space where the posterior probability is significant (i.e., non-zero) in order to facilitate calibration.

Takhtaganov and Müller [8] presented an active learning algorithm selecting training data around the posterior distribution using a gradient-based maximization of expected improvement [7]. Kandasamy et al. presented a similar approach but used their surrogate to approximate the likelihood function[1] rather than the computational model itself [9]. The authors presented two methods for selecting new training points that performed equally well. The first was similar to that of [8] and required a computationally demanding optimization, while the second simply selected the point with the highest surrogate-predicted variance. Inspired by these works, an alternative active learning algorithm is proposed herein that aims to address specific challenges associated with generating surrogates for the calibration of expensive models. Similar to [9], the method selects the training point with the highest predictive variance but does so from a set of candidate points sampled from the surrogate-estimated posterior distribution at each active learning iteration.

2 Theory

2.1. Bayesian model calibration

In this work, the goal is to calibrate an expensive computational model to noisy measurements of one or more model response variables. The model, $\mathcal{M}(\theta)$, is a

[1]In Bayesian statistics, the likelihood function defines the likelihood of observing the data as a function of the model parameters and can be used to perform model calibration with MCMC. Traditionally, the likelihood function is dependent on the model response, but the approach in [9] skips the need to evaluate the model directly. Skipping the direct evaluation of the model can have disadvantages if the model response is of particular interest rather than the calibration parameters only; e.g., when performing surrogate model validation independent of the calibration data.

function of a d-dimensional vector of model parameters (i.e., $\theta \in \Omega \subset \mathbb{R}^d$) and produces predictions,

$$y = \mathcal{M}(\theta), \tag{1}$$

where y is an n-dimensional vector of measurable model responses (e.g., displacements of a beam in bending, melt pool dimensions in an additive manufacturing process, etc.). Ω is the calibration parameter space and $\mathbb{R}^d$ denotes the d-dimensional real space. Here, it is assumed that the model is related to a single measurement by

$$z = y + \varepsilon = \mathcal{M}(\theta) + \varepsilon, \tag{2}$$

where z is the n-dimensional vector of measured values and ε is a sample from a zero-mean multivariate normal (MVN) random variable with $n \times n$ covariance matrix Σ; i.e., $\varepsilon \sim MVN(0, \Sigma)$. This term is a probabilistic description of the measurement variability (e.g., noise).

Given a set of measurements, $\mathbf{z} = \{z_1, \ldots, z_m\}$, the model is calibrated in a probabilistic fashion by solving the inverse problem posed by Bayes' Theorem,

$$\pi(\theta|\mathbf{z}) = \frac{\pi(\mathbf{z}|\theta)\pi(\theta)}{\int_\Omega \pi(\mathbf{z}|\theta)\pi(\theta)d\theta}, \tag{3}$$

where $\pi(\cdot)$ represents a PDF. The left-hand side of Eq. 3 is referred to as the posterior PDF and describes the relative likelihood that the model parameters take on a particular value conditioned on the available data. Thus, the posterior distribution is the solution to the inverse problem and provides a probabilistic calibration of the model based on available data. The numerator on the right-hand side comprises the product of the likelihood function, $\pi(\mathbf{z}|\theta)$ and the prior PDF, $\pi(\theta)$. The former is a measure of how likely it would be to observe $\mathbf{z}$ given a particular θ, and the latter quantifies any prior knowledge about θ. The form of $\pi(\theta)$ is typically derived from expert elicitation or previous experiments. The denominator on the right-hand side is a normalizing constant and involves integration over the entire parameter space.

Obtaining a direct solution to Eq. 3 is often intractable in practice due in part to the complexity of the expensive integral in the denominator. Computation of this integral can be bypassed through the use of sampling methods (e.g., MCMC) that exploit the proportionality $\pi(\theta|\mathbf{z}) \propto \pi(\mathbf{z}|\theta)\pi(\theta)$ to generate samples from the unknown posterior distribution. A large number (e.g., on the order of thousands to millions) of model evaluations are typically required to provide an accurate solution to the inverse problem, as the likelihood function must be evaluated one or more times[2] for each sample generated. When $\mathcal{M}(\cdot)$ is a high-fidelity simulator, as is the motivation for this work, this requirement can lead to exorbitant computation times. To alleviate this burden, surrogate models can be used to approximate the high-fidelity model input-output relationship; i.e.,

$$\widehat{y} = \widehat{\mathcal{M}}(\theta) \approx y. \tag{4}$$

[2]Algorithms such as MCMC are based on sample acceptance/rejection probabilities that mean a number of candidate samples may be discarded, so the total number of evaluations M for a final sample size N is often much higher; i.e., $M >> N$.

Assuming that the surrogate is faster to evaluate than the original model, sampling methods can then be used in a tractable manner to obtain the approximate calibration

$$\widehat{\pi}(\theta|\mathbf{z}) \approx \pi(\theta|\mathbf{z}), \tag{5}$$

where $z \approx \widehat{y} + \varepsilon$.

2.2. Active learning for efficient surrogate modeling

A common approach to developing a surrogate model is supervised machine learning on a set of N pairs of model inputs and pre-computed model outputs. This set is referred to as training data,

$$\mathcal{D}_N = (\Theta_N, \mathbf{Y}_N), \tag{6}$$

where

$$\begin{aligned} \Theta_N &= \{\theta_i\}_{i=1}^N, \\ \mathbf{Y}_N &= \{\mathcal{M}(\theta_i)\}_{i=1}^N = \{y_i\}_{i=1}^N. \end{aligned} \tag{7}$$

Θ_N has dimensions $N \times d$, and $\mathbf{Y}_N$ has dimensions $N \times n$. The inputs Θ_N can be over a uniform grid, randomly sampled, generated using a space-filling technique such as LHS, etc. The upfront computational cost of training the surrogate model increases with N. Depending on the complexity of the learned input-output relationship and the runtime of $\mathcal{M}(\theta)$, the construction of $\mathcal{D}_N$ itself can be computationally challenging.

Active learning is a current area of research in the field of machine learning and generally describes a method in which a model trained on $\mathcal{D}_N$ is used to select a new training point θ_{N+1} by solving the optimization problem

$$\theta_{N+1} = \arg\max_{\theta} g(\theta; \widehat{\mathcal{M}}_N(\cdot)), \tag{8}$$

where g is an acquisition function that quantifies the estimated improvement of the surrogate model achieved by adding an arbitrary θ to the training dataset and $\widehat{\mathcal{M}}_N(\cdot)$ is the surrogate model trained on $\mathcal{D}_N$. Estimated improvement can be quantified using acquisition functions based on predictive variance of the surrogate, or the amount of uncertainty associated with $\widehat{y}$.[3] Gaussian process regression (GPR) models [10] provide both a mean prediction and a predicted variance when predicting between points in Θ_N and, therefore, are particularly useful for active learning. While other methods could be used for this purpose (e.g., Bayesian neural networks [11]), the remainder of this work will focus on using GPRs to facilitate active learning.

A simple acquisition function can be used to target regions with the highest predictive uncertainty,

$$g(\widehat{y}) = \mathrm{Var}[\widehat{y}], \tag{9}$$

where $\mathrm{Var}[\widehat{y}]$ is the predicted output variance. In cases where Ω is unbounded, this acquisition function would result in divergence with $\Theta_{N\to\infty}$ as GPRs have high uncertainty when extrapolating. In the present study, the goal is to have an accurate

[3]The dependence on N is excluded from the $\widehat{y}$ notation for simplicity.

surrogate model in the region of non-zero posterior probability (i.e., $\pi(\theta|\mathbf{z}) > 0$). However, this region is unknown *a priori* making it difficult to apply any bounds. An alternative approach is the use of a candidate set, $\mathcal{C}_N \subset \Omega$ to transform Eq. 8 into an easier-to-solve discrete optimization problem,

$$\theta_{N+1} = \underset{\theta \in \mathcal{C}_N}{\arg\max}(\mathrm{Var}[\widehat{y}]). \tag{10}$$

A natural choice for this candidate set would be samples from the posterior distribution itself, ensuring that the model's predictive uncertainty is being selectively reduced only in the region of interest, rather than in areas with very low posterior probability. The true posterior distribution is the target of the proposed calibration approach and is thus unavailable to sample from. However, a surrogate-approximated posterior is available (Eq. 5), yielding candidate set

$$\mathcal{C}_N = \{c_j\}_{j=1}^{M}, \tag{11}$$

with candidate parameter vectors sampled from the posterior,

$$c_j \sim \widehat{\pi}(\theta|\mathbf{z}). \tag{12}$$

The optimization of Eq. 10 can then be accomplished by sampling from the posterior using the methods highlighted in Section 2.1 and selecting the point with the highest variance in a greedy fashion as outlined in Algorithm 1. Some of the notation and concepts in the algorithm are explained in subsequent subsections. Note that any sampling algorithm can be used to perform the final calibration in step 14; e.g., MCMC.

Algorithm 1 Active learning for Bayesian Model Calibration

Require: $\mathcal{M}(\cdot), \mathbf{z}, N, t$
1: Initialize the training inputs, $\Theta_N \sim$ LHS
2: Initialize the training outputs ,$\mathbf{Y}_N \leftarrow \mathcal{M}(\Theta_N)$
3: $\mathcal{D}_N \leftarrow (\Theta_N, \mathbf{Y}_N)$
4: **while** not converged **do**
5: Train a GPR model, $\widehat{\mathcal{M}}_N(\cdot) \leftarrow \mathcal{GP}(\mathcal{D}_N)$
6: Generate the candidate set using SMC, $\mathcal{C}_N^{(t)} \sim \mathrm{SMC}(t, \mathbf{z}, \widehat{\mathcal{M}}_N(\cdot))$
7: $\theta_{N+1} \leftarrow \arg\max_{\theta \in \mathcal{C}_N^{(t)}}(\mathrm{Var}[\widehat{y}])$
8: $y_{N+1} \leftarrow \mathcal{M}(\theta_{N+1})$
9: Check for convergence; e.g., using y_{N+1} and $\widehat{\mathcal{M}}_N(\theta_{N+1})$
10: $N \leftarrow N + 1$
11: Update t based on desired degree of exploration/exploitation
12: **end while**
13: Train final GPR model, $\widehat{\mathcal{M}}_N(\cdot) \leftarrow \mathcal{GP}(\mathcal{D}_N)$
14: Perform Bayesian model calibration, $\widehat{\pi}(\theta|\mathbf{z}) \leftarrow \mathrm{SMC}(t = T, \mathbf{z}, \widehat{\mathcal{M}}_N(\cdot))$

2.2.1. Initializing the surrogate

As is common in active learning methods, an initial set of training data is needed to start the process. The initial set can be small relative to the final number of points ultimately required for acceptable surrogate accuracy. Ideally, the initial samples are representative of the model response variability over the entire parameter space, making LHS a useful tool to generate the input samples, Θ_N. The full-fidelity model must then be run at each sample to generate the labeled outputs, $\mathbf{Y}_N$, before the initial surrogate model can be trained.

2.2.2. Constructing a candidate set

Drawing samples per Eq. 12 is non-trivial. As discussed in Section 2.1, Monte Carlo methods can be used to draw samples from the unknown posterior distribution with a computational expense. MCMC is typically used for this purpose but requires a burn-in period in which a number of samples are discarded before the method begins producing samples from $\widehat{\pi}(\theta|\mathbf{z})$, which amounts to wasted computation.

Sequential Monte Carlo (SMC) [12] is an ideal alternative to MCMC for active learning in that it does not require a burn-in period and allows for model evaluations to be run in parallel, yielding significant speedup. Rather than trying to sample from the posterior distribution directly, SMC uses a sequence of T target distributions that gradually transition from one that is easy to sample from (i.e., the prior distribution) to the posterior distribution of interest. A population of weighted parameters, called particles, is initially sampled from the first target distribution and then evolved sequentially via local moves and reweighting steps. At each step ($t = 0, \ldots, T$), the particles represent samples from the t^{th} target marginal distribution, with $t = T$ being the posterior distribution. The target distributions are typically defined using likelihood annealing such that $\pi_t(\theta|\mathbf{z}) \propto \pi(\mathbf{z}|\theta)^{\phi_t}\pi(\theta)$ where $\phi_t \in [0, 1]$ and is strictly monotonically increasing with t. See [12] for additional information.

Apart from providing significant speedup, SMC also enables smooth transition between *exploration* and *exploitation* when executing the proposed algorithm. These terms are commonly used in active learning literature to differentiate exploration of the parameter space through global sampling techniques (e.g., LHS) from local, targeted sampling (e.g., from the posterior distribution). Exploration improves global surrogate accuracy whereas exploitation focuses on making the surrogate accurate in the region of interest. With SMC, the degree of exploration versus exploitation can be controlled by forming the candidate set from the particle population at different target distributions in the SMC sequence,

$$c_j \sim \pi_t(\theta|\mathbf{z}), \tag{13}$$

where t is a tuning knob. Lower values of t produce samples from targets more similar to the prior distribution. When using a uniform prior over the parameter space, this means that $t = 0$ is equivalent to uniform random sampling from the parameter space. Increasing the value of t increases the amount of exploitation, reducing exploration and focusing more on the region of non-zero posterior probability. Full exploitation may be required if the computational budget is low, whereas some exploration may be preferred to ensure accurate resolution of the posertior distribution tails or to identify additional modes. A mix or cyclical sweep between exploration and exploitation during active learning is also easy to achieve with SMC. When using $t < 1$, SMC can be terminated early to improve efficiency of the algorithm.

2.2.3. Monitoring for convergence

Active learning can be continued for a pre-defined number of iterations or until a convergence criterion is met. Suggested approaches are to monitor either (i) some statistic representative of surrogate model error with respect to the measured data or (ii) the change in the posterior distribution itself. An example of the former is the Bollinger band (BB) [13], a simple analysis method for monitoring prices and price volatility over time. Adapting this method to the present active learning algorithm, the upper and lower BBs are

$$\beta^{\pm} = \mu_h \pm 2\sigma_h, \tag{14}$$

where μ_h and σ_h are the P-point moving average and standard deviation of an error function, $h(y_{N+1}, \widehat{\mathcal{M}}_N(\theta_{N+1}))$, that quantifies error between newly-evaluated training points and the corresponding surrogate-predicted points after N completed active learning iterations. Note that the convergence criterion is calculated *before* the surrogate is updated, avoiding a double use of data. As reflected in Algorithm 1, errors for which statistics are computed must be collected at each iteration – prior to updating the surrogate but after the proposed training point has been evaluated by the high-fidelity model.

A threshold value ϵ_{surr} on the upper BB could be used to determine convergence of the active learning scheme; e.g., the surrogate model has converged when $\beta^+ < \epsilon_{surr}$. Conversely, a threshold can be placed on a rate of change of the upper BB; e.g., the surrogate model has converged when $\frac{d\beta^{(+)}}{dN} < \epsilon_{surr}$. The use of BBs provides a simple means for determining (a) when the surrogate model mean relative error is low and (b) when volatility in the surrogate error is low without knowing how low either quantity can get (i.e., stability).

Alternatively, changes in the estimated model posterior probability distribution can be monitored and the algorithm can be terminated in a similar fashion as above; e.g., when the change between subsequent iterations falls below a threshold. A number of methods exist for comparing probability distributions. For example, the Wasserstein distance [14] or Kullback–Leibler divergence [15] could be estimated using the posterior samples $\mathcal{C}_N^{(t=T)}$ and $\mathcal{C}_{N-1}^{(t=T)}$. One could also monitor the marginal likelihood, which is the previously-ignored normalizing constant in Eq. 3. The marginal likelihood represents the ability of the calibrated stochastic model to predict the observed data. For a fixed set of calibration data, as the surrogate model converges to the full-fidelity response surface during active learning, the mean value of the marginal likelihood will also converge. As mentioned previously, the marginal likelihood is at best challenging to compute directly. However, SMC provides an unbiased estimate of the marginal likelihood as a byproduct of the sampling process at $t = T$ [12], meaning it is available without any additional computational effort.[4]

[4]Practitioners should be aware that the SMC marginal likelihood estimate is itself a random variable, meaning that the mean marginal likelihood will converge in the limit of active learning iterations but there will be an associated variance that is a function of the specific SMC hyper-

3 Example

To demonstrate the proposed active learning approach, a toy example was derived using a single sample from a one-dimensional (i.e., $\theta \in \Omega \subset \mathbb{R}$ and $y \in \mathbb{R}$), zero-mean Gaussian process prior generated using the `scikit-learn` Python package [16]. This sample was assumed to be the ground truth, "high-fidelity" model, $\mathcal{M}(\theta)$, to be calibrated. For the purpose of the demonstration, this model was assumed to be too expensive to evaluate directly. The calibration data comprised five independent and identically distributed samples, $z_i \sim N(\mu, \sigma)$, with mean $\mu = 1.23$, standard deviation $\sigma = 0.1$, and $i = 1, \ldots, 5$. This data distribution was chosen such that it would cross the ground truth function at three distinct θ values, producing a trimodal posterior distribution. The function, data, and a histogram of the estimated posterior distribution are shown in Fig. 1.

Throughout this example, estimates of posterior distributions were obtained using `SMCPy`, a Python package for parallelized SMC sampling [17]. This implementation of SMC was the same as that used in the active learning procedure (see Algorithm 1). The `AdaptiveSampler` class was used with $1,000$ particles, 10 MCMC moves per adaptive iteration, and a target effective sample size of 800 particles. In practice, σ would typically be estimated along with model parameters, θ; however,

parameters chosen for the analysis. Thus, a running average of marginal likelihood would be an appropriate way to track convergence.

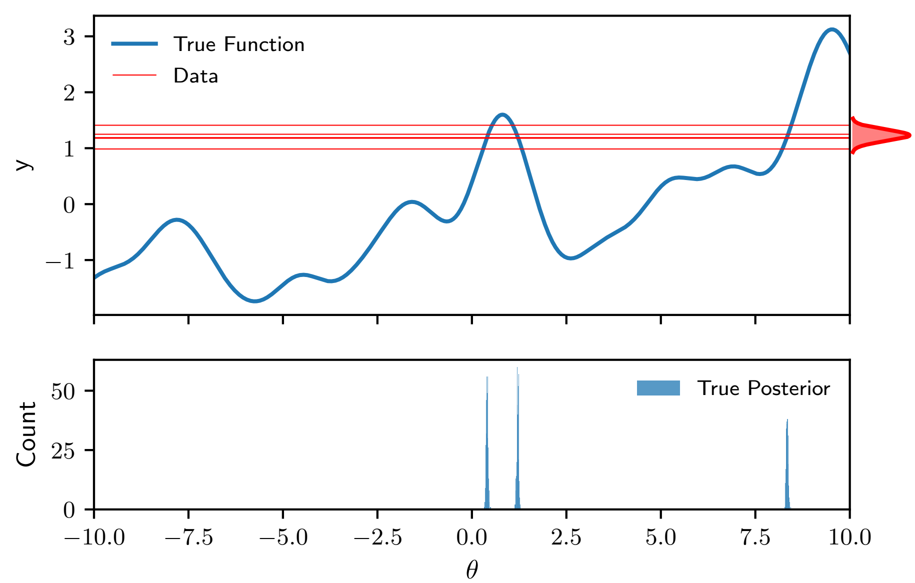


Figure 1: True function to be calibrated along with available calibration data and the resulting posterior distribution (i.e., the result of the calibration using the true function directly). The red PDF on right axis is the true data-generating distribution and is shown here for completeness.

for simplicity and to amplify separation between posterior modes, this value was assumed to be known. A uniform prior distribution was assumed over the interval $[-10, 10]$.

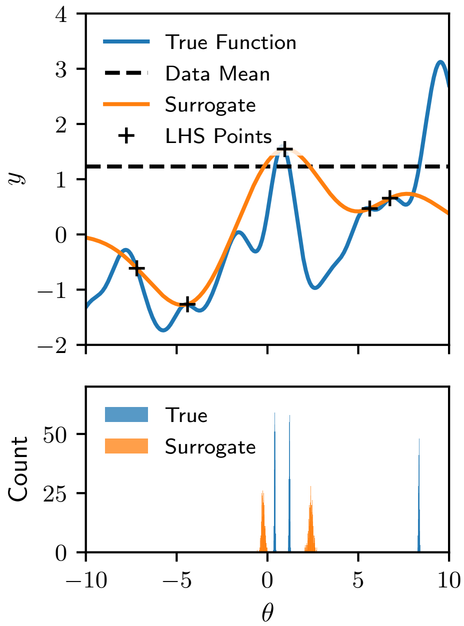
To motivate the use of active learning, surrogates trained with increasing amounts of data from LHS were first studied. The surrogate models trained on 5 points, 10 points, 15 points and 25 points are shown in Fig. 2 along with a comparison of the surrogate-estimated posterior distributions relative to that obtained using $\mathcal{M}(\theta)$ directly. LHS-based training exhibits global improvements in accuracy at the expense of accuracy in the region of the posterior distribution, resulting in missed modes for the 5 and 10 point cases and biased estimates for the 15 and 25 point cases. Zoomed versions of the 15 and 25 point cases are shown in Fig. 3. While all three modes are captured, even the 25 point case shows significant bias, particularly in the estimate of the third mode.

In contrast to the LHS results, applying the proposed active learning strategy with $t = T$ (i.e., full exploitation) results in earlier convergence and more efficient placement of training points as shown in Fig. 4. Here, only 15 total training points were used, 5 from the initial LHS and 10 from active learning. However, focusing on exploitation came at a cost, as the third mode was not identified due to the concentration of training points near the first two modes (see Fig. 5). Since the initial LHS sample included a single high-fidelity model evaluation above the mean data line, the initial surrogate prior to any active learning was able to identify at least one posterior mode in that region with relatively high certainty relative to the rest of the parameter space. As a result, there was negligible probability density at the third mode, and no candidate samples were generated there.

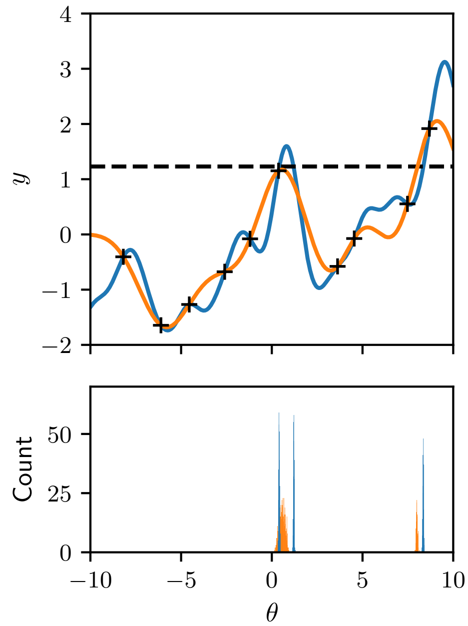
Implementing a more balanced exploration and exploitation strategy was shown to overcome the inability to identify the third mode. In this case, samples were generated from intermediate SMC target distributions according to Eq. 13 with $t \leq T$. The specific implementation of this approach (line 11 of Algorithm 1) was left to the user, although optimal methods should be the subject of future study. Here, it was found that varying t at each active learning iteration to produce the following sequence of repeated ϕ_t values

$$\Phi = [0.03, 0.03, 0.03, 1.0, 1.0, \ldots, 0.03, 0.03, 0.03, 1.0, 1.0] \tag{15}$$

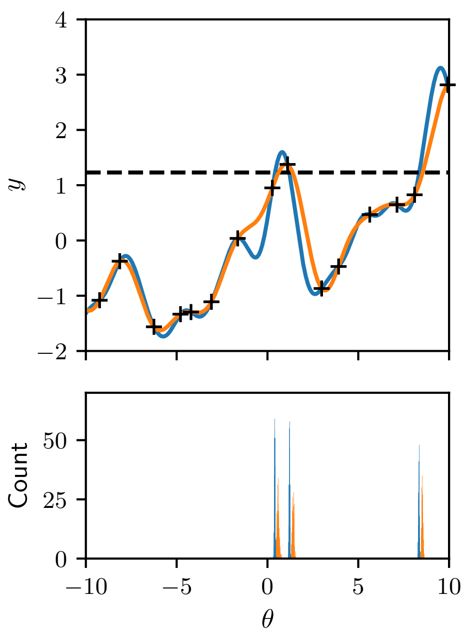
was effective. This choice meant that every three exploration steps were followed by two exploitation steps until the desired number of training points had been generated. Again, 15 total training points were used, and, as shown in Fig. 6 and Fig. 7, all three modes are captured with better accuracy than any of the previous surrogate models, including the 25-point LHS model (Fig. 2). By setting $\phi_t = 0.03$ for exploitation steps, more weight is given to the prior distribution than the likelihood function, meaning the target distribution is a more diffuse version of the posterior. An example of this is shown in Fig. 8 in which the target distributions for $\phi_t = 0.03$ and $\phi_t = 1.0$ are plotted prior to the first active learning iteration. The former results in candidate samples to be generated at the third mode while the latter does not. The choice of $\phi_t = 0.03$ was arbitrary, any $\phi_t \in [0, 1]$ could be chosen to smoothly control the degree of exploration and exploitation where



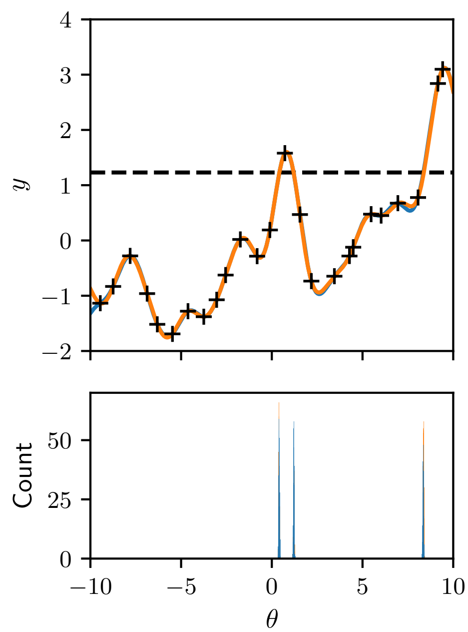
(a) 5 training points



(b) 10 training points



(c) 15 training points



(d) 25 training points

Figure 2: Surrogate models trained using LHS with (a) 5 training points (b) 10 training points (c) 15 training points and (d) 25 training points compared to the true function. Histograms of the corresponding posterior distribution estimates are shown in the bottom frame of each subfigure.

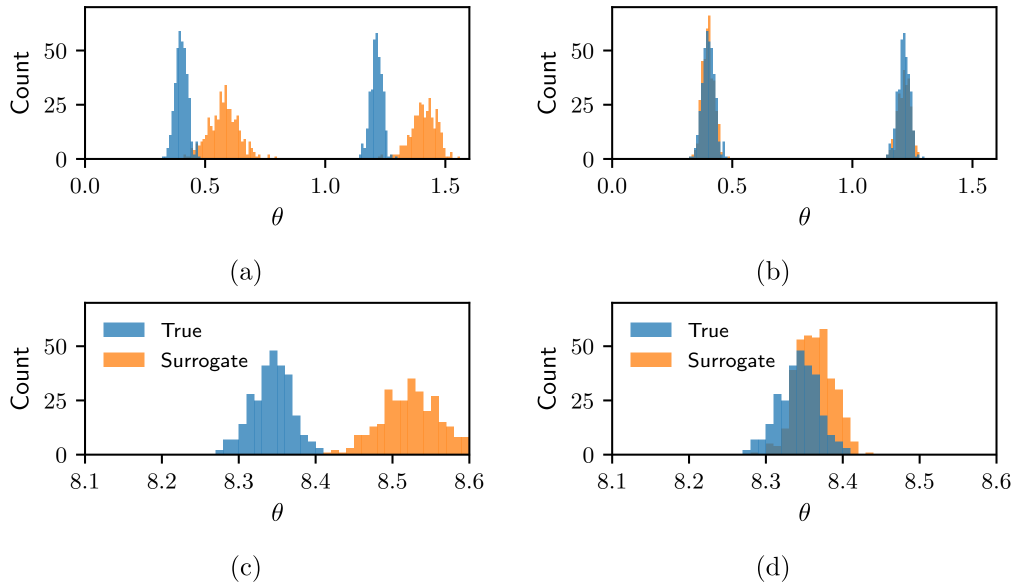


Figure 3: Zoomed views of estimates for the first two modes (from left to right) of the posterior distribution for the surrogate trained with (a) 15 points and (b) 25 points. The third mode is shown in (c) and (d) for 15 and 25 training points, respectively.

$\phi_t = 0$ corresponds to uniform sampling (full exploration) and $\phi_t = 1$ corresponds to sampling from the posterior (full exploitation).

To further illustrate the utility of the proposed active learning approach, and, in particular, the benefit of the controlling exploration and exploitation, the marginal log likelihood (MLL) was estimated for all three surrogate model types (i.e., LHS, exploitation only, and mixed exploration/exploitation). The results are plotted as a function of training points in Fig. 9. Both active learning methods result in relatively quick convergence (within approximately six iterations). However, the MLL for the model using $\phi_t = 1$ is clearly biased due to the missing mode when compared to the ground truth MLL (estimated as the average over 10 repeated SMC simulations). In contrast, the mixed approach results in convergence to the true MLL, with all three modes represented. The LHS approach does not exhibit any clear convergence up to 15 total training points. A final comparison of estimated posteriors is provided in Fig. 10. The active learning surrogate clearly outperforms the 15 point LHS surrogate. Furthermore, it appears to have less bias than the 25 point LHS even though it was trained with 40 % less data. These results illustrate the effectiveness of the approach and highlight the potential gains in both efficiency and accuracy over traditional space-filling approaches.

4 Summary

A general active learning algorithm for calibrating expensive computational models was presented that sequentially generates training points from gradually-improving

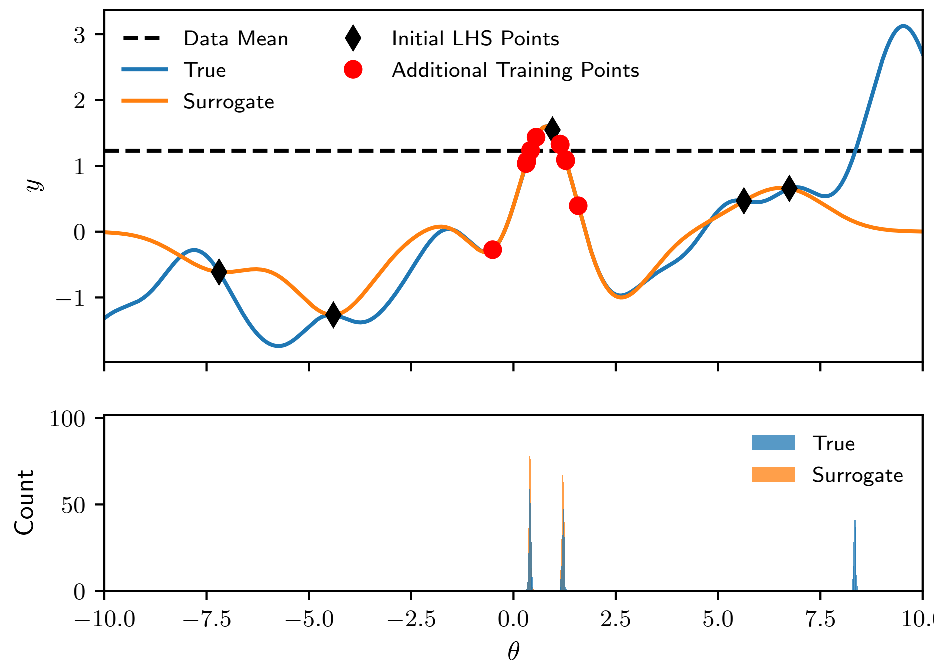
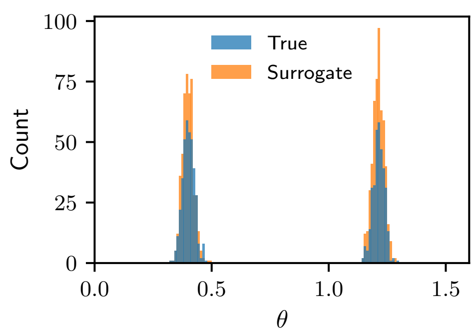
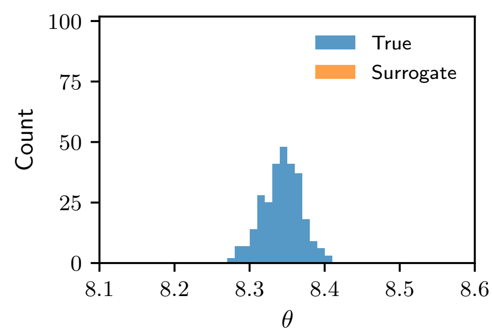


Figure 4: Surrogate model trained using 15 total training points (an initial 5 training points from LHS followed by 10 training points selected according to Algorithm 1 with $\phi_t = 1.0$). The resulting posterior distribution is shown in the bottom frame.



(a)



(b)

Figure 5: Zoomed views of (a) the first two modes and (b) the third mode of the posterior distribution. Posterior estimates for both the true function and the surrogate trained via Algorithm 1 with $\phi_t = 1.0$ are shown. Note that the surrogate-derived posterior does not produce training points in the third mode.

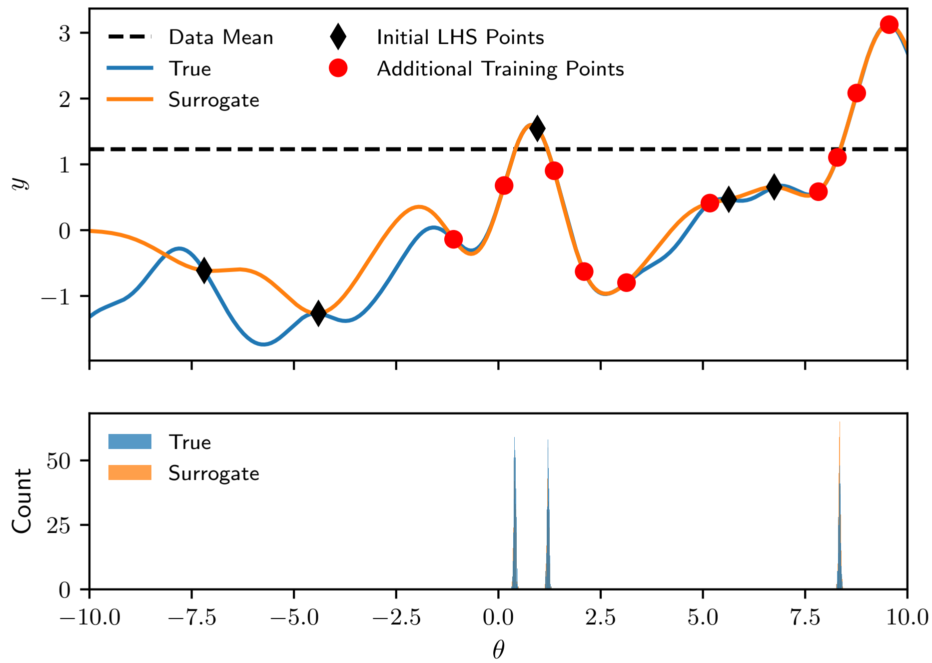
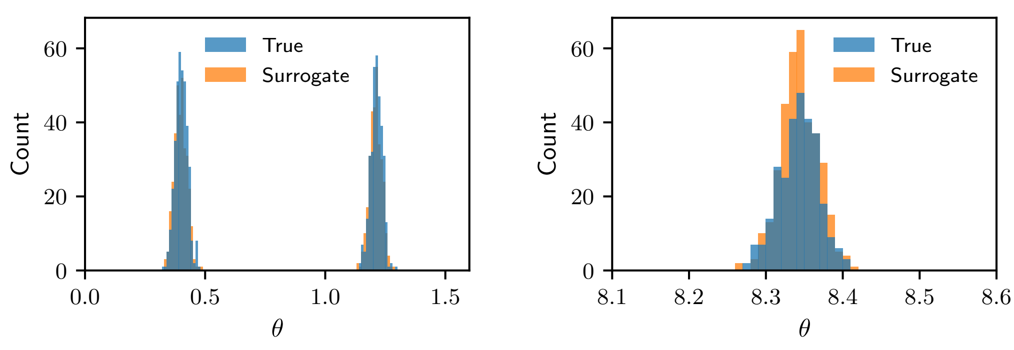


Figure 6: Surrogate model trained using 15 total training points (an initial 5 training points from LHS followed by 10 training points selected according to Algorithm 1 with $\phi_t \in \Phi$). The resulting posterior distribution is shown in the bottom frame.



(a) (b)

Figure 7: Zoomed views of (a) the first two modes and (b) the third mode of the posterior distribution estimates using the true function and the surrogate trained using active learning with balanced exploration and exploitation.

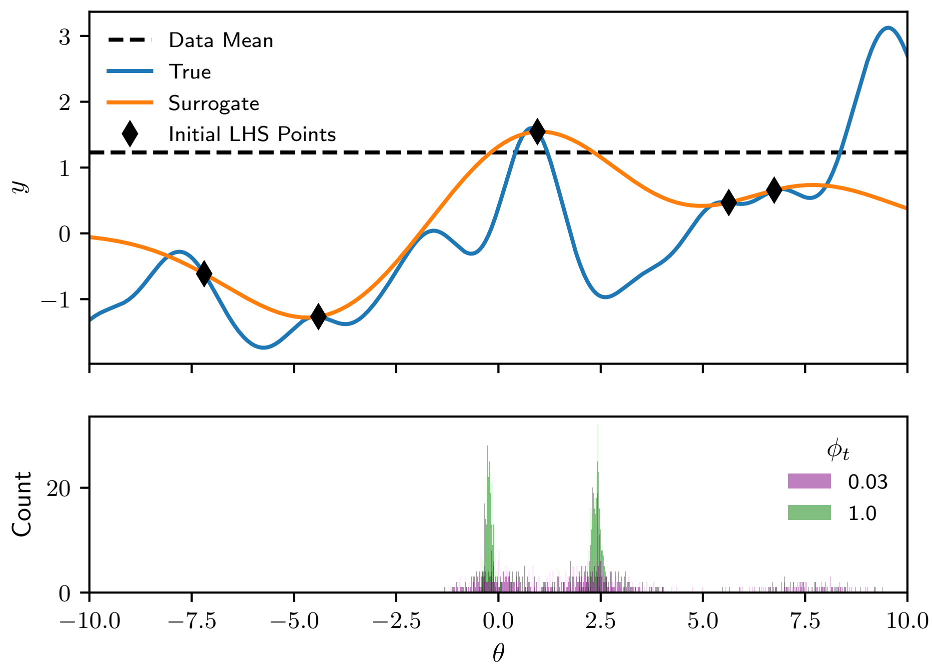


Figure 8: Comparison of the candidate sample distributions corresponding to $\phi_t = 0.03$ and $\phi_t = 1.0$ using a surrogate trained with 5 points from the initial LHS only. The use of smaller ϕ_t values allows for a controllable degree of exploration and, in this example, allows for the third mode to be captured while still achieving better local accuracy.

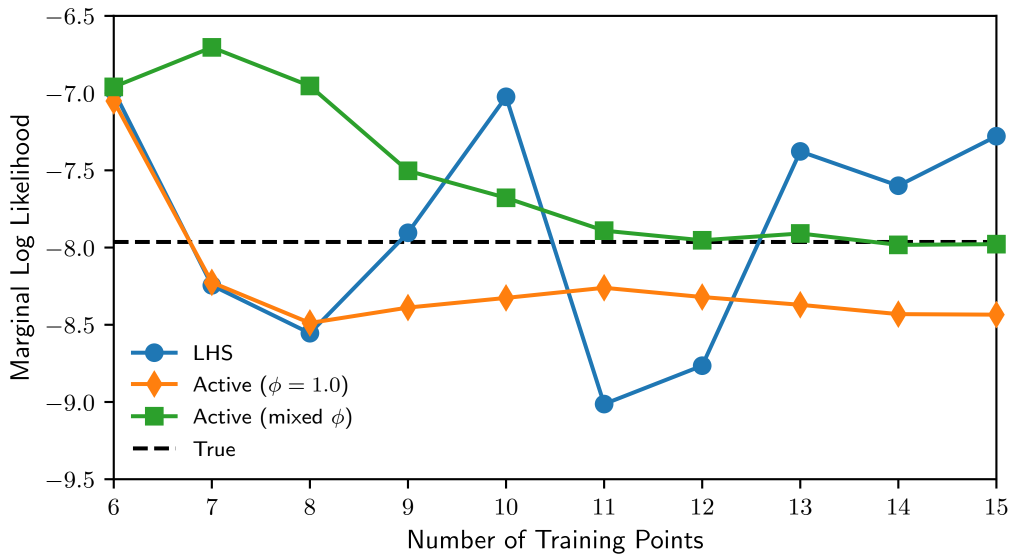


Figure 9: Marginal log likelihood estimates versus number of training points. Higher values are associated with models that better match the data.

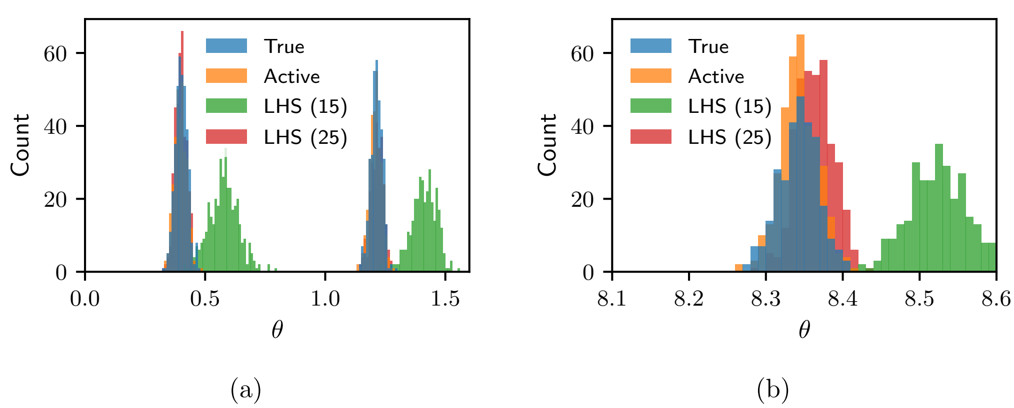


Figure 10: Zoomed views of (a) the first two modes and (b) the third mode of the posterior distribution estimates using the true function and surrogates trained using mixed ϕ active learning, 15 LHS training points, and 25 LHS training points.

approximations of the parameter posterior distribution (i.e., the solution to the calibration problem). The method uses Gaussian process regression to build the surrogate model, which provides estimates of predictive uncertainty and enables selection of the next training point from a sampled set of candidate points at each iteration of the algorithm. A sequential Monte Carlo sampler was suggested to conduct the calibration and construct the candidate set due to its computational efficiency and natural ability to tune smoothly between exploration and exploitation of the parameter space. Proposed methods for monitoring convergence and concluding the active learning process were also presented along with a basic example problem demonstrating the utility of the approach. Further exploration of these proposals and comparison to existing algorithms would be a useful contribution. Planned future work involves implementation of the algorithm for calibrating expensive material process models in the field of additive manufacturing.

References

1. R. C. Smith, Uncertainty quantification: Theory, implementation, and applications, Computational Science and Engineering, SIAM, 2013.

2. W. R. Gilks, S. Richardson, D. Spiegelhalter, Markov Chain Monte Carlo in practice, CRC press, 1995.

3. W.-L. Loh, On Latin hypercube sampling, The Annals of Statistics 24 (5) (1996) 2058–2080.

4. B. Settles, From theories to queries: Active learning in practice, in: I. Guyon, G. Cawley, G. Dror, V. Lemaire, A. Statnikov (Eds.), Active Learning and Experimental Design workshop In conjunction with AISTATS 2010, Vol. 16 of Proceedings of Machine Learning Research, PMLR, Sardinia, Italy, 2011, pp. 1–18.

5. D. A. Cole, R. B. Gramacy, J. E. Warner, G. F. Bomarito, P. E. Leser, W. P. Leser, Entropy-based adaptive design for contour finding and estimating reliability, Journal of Quality Technology 55 (1) (2023) 43–60.

6. V. Picheny, D. Ginsbourger, O. Roustant, R. T. Haftka, N. H. Kim, Adaptive designs of experiments for accurate approximation of a target region, Journal of Mechanical Design, Transactions of the ASME 132 (7) (2010) 0710081–0710089. doi:10.1115/1.4001873.

7. D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global Optimization 13 (4) (1998) 455–492. doi:10.1023/A:1008306431147.

8. T. Takhtaganov, J. Müller, Adaptive Gaussian process surrogates for Bayesian inference (September 2018). arXiv:1809.10784.
URL `http://arxiv.org/abs/1809.10784`

9. K. Kandasamy, J. Schneider, B. Póczos, Bayesian active learning for posterior estimation, in: Proceedings of the 24th International Conference on Artificial Intelligence, 2015, pp. 3605–3611.

10. C. E. Rasmussen, C. K. Williams, Gaussian processes for machine learning, 1st Edition, The MIT Press, 2006.

11. L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, M. Bennamoun, Hands-on Bayesian neural networks—a tutorial for deep learning users, IEEE Computational Intelligence Magazine 17 (2) (2022) 29–48.

12. P. Del Moral, A. Doucet, A. Jasra, Sequential Monte Carlo samplers, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68 (3) (2006) 411–436.

13. J. Bollinger, Using Bollinger bands, Stocks & Comodities 10 (2) (1992) 47–51.

14. L. Rüschendorf, The Wasserstein distance and approximation theorems, Probability Theory and Related Fields 70 (1) (1985) 117–129.

15. J. M. Joyce, Kullback-Leibler divergence, in: International Encyclopedia of Statistical Science, Springer, 2011, pp. 720–722.

16. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

17. P. Leser, SMCPy - Sequential Monte Carlo with Python, `https://github.com/nasa/smcpy` (2022).

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704–0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
01-05-2023	Technical Memorandum	

4. TITLE AND SUBTITLE
Efficient Calibration of Expensive Computational Models

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Patrick E. Leser[5], Joshua M. Fody[6]

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, Virginia 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NASA/TM–20230006753

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category
Availability: NASA STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES
An electronic version can be found at http://ntrs.nasa.gov.

14. ABSTRACT
Accounting for uncertainty when calibrating expensive computational models is a common challenge faced by scientists and engineers. Often Bayesian techniques are adopted to estimate a probability density function over the model parameters given noisy empirical data. The methods used to perform this type of probabilistic calibration are computationally prohibitive in that they require a large number of evaluations of the expensive model. In these cases, surrogate modeling – that is, using a fast-to-evaluate, lower fidelity stand-in for the original computational model – may be the only option to alleviate this computational burden. However, the upfront cost of generating training data to build a surrogate model can itself be expensive. As such, it is important to be judicious when selecting training points at which the full-fidelity model is evaluated. Here, an active learning approach is proposed that enables efficient selection of training points using approximate samples of the calibrated parameter probability density function. In this way, the training points can be concentrated in regions where the calibration algorithm requires high model accuracy.

15. SUBJECT TERMS
uncertainty quantification, model calibration, active learning

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			STI Information Desk (help@sti.nasa.gov)
U	U	U	UU		**19b. TELEPHONE NUMBER** ***(Include area code)*** (757) 864-9658