LOCAL VORONOI TESSELLATIONS FOR ROBUST MULTI-WAVE CALIBRATION OF COMPUTER MODELS

Wenzhe Xu,[1,*] Daniel B. Williamson ,[2] & Peter Challenor[3]

[1]*Department of Mathematical Sciences University of Exeter, UK, EX4 4PY*

[2]*Department of Mathematical Sciences University of Exeter, UK, EX4 4PY and The Alan Turing Institute The British Library, London, UK, NW1 2DB*

[3]*Department of Mathematical Sciences University of Exeter, UK, EX4 4PY and The Alan Turing Institute The British Library, London, UK, NW1 2DB*

*Address all correspondence to: Wenzhe Xu, Department of Mathematical Sciences, University of Exeter, UK, EX4 4PY, E-mail: wx229@exeter.ac.uk

Original Manuscript Submitted: 17/05/2020; Final Draft Received: mm/dd/yyyy

History matching using Gaussian process emulators is a well-known methodology for the calibration of computer models. It attempts to identify the parts of input parameter space that are likely to result in mismatches between simulator outputs and physical observations by using emulators. These parts are then ruled out. The remaining "Not Ruled Out Yet (NROY)" input space is then searched for good matches by repeating the history matching process. An easily neglected limitation of this method is that the emulator must simulate the target NROY space well, else good parameter choices can be ruled out. We show that even when an emulator passes standard diagnostic checks on the whole parameter space, good parameter choices can easily be ruled out. We present novel methods for detecting these cases and a Local Voronoi Tessellation method for a robust approach to calibration that ensures that the true NROY space is retained and parameter inference is not biased.

KEY WORDS: *History matching, Uncertainty quantification, Calibration, Gaussian Processes, Emulator Diagnostics*

2152–5080//$35.00

1. INTRODUCTION

Computer models typically solve physical equations, such as coupled systems of PDEs in order to learn about features of the real-world. Calibration of computer models broadly involves using partial and imperfect observations of the real world in order to learn which settings of the model's input parameters lead to outputs that are consistent with real-world observations given relevant uncertainties such as measurement error and model discrepancy (we define these terms in Section 2.2).

When a computer model is inexpensive, it can be embedded in an MCMC or optimization algorithm for calibration directly (see e.g. [1]). However, many computer models are expensive and/or take a long time to run. For example, climate models may take days or weeks of run time on supercomputers [2]. When it is not possible to run the model often enough to calibrate directly, a small, carefully chosen, set of model runs, often termed a 'design' or 'ensemble', can be run and used to construct an 'emulator' or 'surrogate'; an inexpensive statistical model used to approximate the computer model [3,4].

Bayesian calibration [5,6] and history matching [7–10] are both extensively used methods for calibrating with an emulator. Bayesian calibration places a probability distribution over a 'best input' and updates this distribution using model runs and observations of the real process. Rather than relying on making distributional assumptions, history matching attempts to identify the parts of the input parameter space that are likely to result in mismatches between computer outputs and observations by iteratively removing those regions of parameter space in which we are virtually certain that there are no good matches. Previous research has applied history matching to many fields, including oil reservoir modelling [11,12], epidemiology [13,14] galaxy formation [15–17] climate systems, [1,18–23]. In this paper we focus on history matching. For discussions comparing the two approaches, see [24,25] and the discussion in [15].

Beginning with an emulator, history matching uses a distance function to rule out input choices that lead to outputs that are 'too far' from observations. The distance, called *implausibility*, is computed according to a norm that standardises according to all relevant uncertainties, including the uncertainty contributed by the emulator. We define this formally in Section 2. Large implausibility regions are ruled out, whilst small implausibility regions may either be good parameter choices or "too uncertain to tell", hence they are termed "Not Ruled Out Yet" (NROY). Following an initial history match (wave 1), a new design is then constructed within NROY space and more accurate emulators are constructed within this region for the purposes of cutting out further space.

A currently unexplored limitation of history matching can occur when the emulator is unable to simulate the unknown target NROY space effectively, even if it seems to pass all standard emulator diagnostic checks [26]. Poor simulation may result in true NROY space being ruled out without any indication for the analyst that this has occurred. For simulators that are constantly under development, such as climate models, this could be a costly mistake that causes parameterizations or even computational methods and hardware to be needlessly revisited, even though the model was already fit for purpose.

This paper will discuss factors that contribute towards ruling out good parameter choices and then will present a novel Local Voronoi Tessellation design that can be used for robust multi-wave calibration of computer models that ensures the true NROY space is retained without biasing the parameter inference. Section 2 will present a review of emulation and history matching methodologies. It will further illustrate these with a numerical example to demonstrate that, even when an emulator validates well on the whole parameter space, good parameter choices can still be ruled out. Section 3 presents the novel detecting method we have developed, as well as a local Voronoi Tessellation method for robust history matching. Section 4 will outline the application of the study methods to two illustrative examples, as well as to the output of the French climate model, IPSL-CM [27,28].

2. EMULATION AND HISTORY MATCHING

2.1 Emulation

Gaussian process emulators are used to approximate expensive computer simulators whilst quantifying uncertainty in the approximation [29,30]. A Gaussian Process (GP) is a stochastic process. Any finite number of random variables from the Gaussian process has a joint Gaussian distribution [31]. Assuming f represents the complex computer model with input parameters x, an emulator for $f(x)$ can be constructed by fitting a mean function $m(x)$ and a correlation function $c(x, x')$ so that

$$f(x)|\beta, \sigma^2, \upsilon, \delta \sim \mathcal{GP}\left(m(x), \sigma^2 c(x, x'; \delta, \upsilon)\right), \tag{1}$$

where $m(x) = h^T(x)\beta$, $\beta = (\beta_1, \beta_2, ..., \beta_k)$ is a vector of unknown regression coefficients, $h(x)$ is a q-length vector of regression functions, σ^2 is a hyperparameter that controls the scaling of the process, δ is a vector of correlation length parameters that used to define the correlation function. The nugget term, υ, is a small number added on the principal diagonal of the design correlation matrix and is often used

to represent observation noise, account for uncertainty in inactive inputs or to avoid numerical instability during computation. The nugget can be specified or it may need to be trained along with the other hyper-parameters (for example, when inputs are varied in the design, yet not included in the correlation function) [32]. With a zero nugget, the emulator interpolates the model runs at the training locations.

The choice of correlation function, $c(x, x')$, is one of the key elements of the Gaussian Process. Standard choices include the power exponential correlation function [21] and the Matérn correlation function [31], both depending on correlation length parameters (which we will refer to as δ throughout), and only on the distance $|x - x'|$, rather than the individual locations x and x'. Such correlation functions, also called kernels, are known as stationary and are used in the majority of cases. An overview of methods for fitting non-stationary Gaussian processes is given in [33].

Let the computer model run at n points, $X = (x_1, \ldots, x_n)^T \in \mathcal{X}$ at which we obtain training runs $\mathbf{F} = (f(x_1), \ldots, f(x_n))$, where $\mathcal{X}$ is a p -dimensional input space. Given the emulator hyperparameters and $\mathbf{F}$, the posterior for $f(x)$ is

$$f(x)|\mathbf{F}, X, \beta, \delta, \upsilon, \sigma^2 \sim \mathcal{GP}(m^*(x), c^*(x, x')), \tag{2}$$

with the posterior expectation $m^*(x)$

$$m^*(x) = h(x)^T\beta + c(x, X)\mathbf{A}^{-1}(\mathbf{F} - h(x)^T\beta),$$

and posterior variance $c^*(x, x')$

$$c^*(x, x') = c(x, x') - c(x, X)\mathbf{A}^{-1}c(X, x')),$$

and $\mathbf{A} = k(X, X)$. There are often different approaches to handling hyperparameters β, δ and σ^2. [29] adopted a maximum likelihood method to fit the hyperparameters. A drawback of this is that the hyper-parameters are usually highly confounded leading to a ridge on the likelihood surface for large δ and σ^2. One way out of this is to specify δ. [30] do this and then propose a 'non-informative' prior $P(\beta, \sigma^2) \propto \sigma^{-2}$. β and σ^2 can then be integrated out leading to a posterior predictive student-t process emulator. [34] use an informative Normal-inverse gamma prior for β and σ^2. Fitting GPs via Full Bayes Markov chain Monte Carlo (MCMC) methods with a benefit that prior distributions can be used to penalise the ridge on the likelihood surface [6,33,35].

2.2 History matching

Like Bayesian calibration, History matching requires a 'best input' assumption linking the model and reality via

$$y = f(x^*) + \eta, \tag{3}$$

where y represents reality, x^* is the 'best input' of the computer model and η is the model discrepancy, which is independent of $f(x)$ and x^* [7]. To learn about x^*, we have observations z with unknown measurement error e and

$$z = y + e, \tag{4}$$

where e has mean zero and is independent of η.

The implausibility measure is used to calculate the distance between the output of the model and the observations so that we can rule out parameter settings that are too far from the observations. For a single output at a given value x, implausibility is defined as

$$\mathcal{I}(x) = \frac{|z - \mathrm{E}\left[f(x)\right]|}{\sqrt{\mathrm{Var}\left[z - \mathrm{E}\left[f(x)\right]\right]}}, \tag{5}$$

where $\mathrm{E}\left[f(x)\right]$ is the emulator prediction. Under (3) and (4)

$$\mathrm{Var}\left[z - \mathrm{E}\left[f(x)\right]\right] = \mathrm{Var}\left[f(x)\right] + \mathrm{Var}\left[e\right] + \mathrm{Var}\left[\eta\right]. \tag{6}$$

For r outputs, the implausibility can be calculated as the maximum across all outputs [11], the second or third largest [25], or via a multivariate version of (5) [e.g.24].

Large values of $\mathcal{I}(x)$ indicate that we are confident that $f(x)$ is too far from the observations and so can be ruled out. The space that has not yet been ruled out, "Not Ruled Out Yet" (NROY) space, $\mathcal{X}^1$, is defined as

$$\mathcal{X}^1 = \{x \in \mathcal{X} | \mathcal{I}(x) \leq T\}, \tag{7}$$

where T is a selected threshold. A common choice is $T = 3$ based on the three sigma rule [36]. We define the NROY space found by the computer model directly (without an emulator) as "true" NROY space or target NROY in order to compare with the NROY space found using an emulator. Target NROY space $\mathcal{X}^*$

is defined as

$$\mathcal{X}^* = \left\{ x \in \mathcal{X} | \frac{|z - f(x)|}{\sqrt{\mathrm{Var}\left[e\right] + \mathrm{Var}\left[\eta\right]}} \leq T \right\}. \tag{8}$$

History matching does not seek to identify $\mathcal{X}^*$ using a single set of computer model evaluations, but through iteratively designed experiments known as 'waves' [15]. In the first wave, the emulator is constructed based on an ensemble at a set of points which cover the whole input space $\mathcal{X}$, then history matching attempts to rule out space from the initial space through (7). Using an ensemble at a new set of points $X^1 \in \mathcal{X}^1$ (defined in equation(7)), a new emulator can be constructed, and a second wave of history matching will further reduce the input space. In general, at wave k, a new set of points x^k is drawn from the wave $k-1$ NROY space, $x^k \in \mathcal{X}^{k-1}$ and are used to construct a new emulator. The wave k NROY space is defined as

$$\mathcal{X}^k = \{x \in \mathcal{X}^{k-1} | \mathcal{I}^k(x) \leq T_k\}, \tag{9}$$

where T_k is a selected threshold for wave k and $\mathcal{I}^k(x)$ is the implausibility function (e.g. (5)) defined on $\mathcal{X}^{k-1}$ and with the wave k emulator. Moreover, for different approaches developed for multi-wave designs $\mathbf{x}_k$, please see [25] and [37].

As NROY space is iteratively reduced, the majority of the runs of the simulator are made closer to the target space, $\mathcal{X}^*$. This ensures the density of points in important regions of the model input space are greater than using Bayesian calibration (assuming an equivalent budget of model runs), ensuring our emulators are more accurate where it counts. In the later waves, we are more likely to believe Normality assumptions implicit in using Gaussian processes, a smaller value of threshold T_k can be adopted for history matching.

2.3 The importance of diagnostics

Before cutting areas of parameter space, diagnostics must be used to assess the adequacy of an emulator. [26] present a variety of diagnostics that compare Gaussian process emulator predictions and simulation outputs at validation points. One example is to look at standardised prediction errors, $D_i(f(x_i))$, calculated via

$$D_i(f(x_i)) = \frac{f(x_i) - \mathrm{E}\left[f(x_i)\right]}{\sqrt{\mathrm{Var}\left[f(x_i)\right]}}, \tag{10}$$

for a set of runs $f(x_1), \ldots, f(x_m)$ left out of the training set. Note often these left out runs are actually those used to fit the emulator, with one at a time left out and the emulator refit (these are Leave One Out (LOO) errors). [26] state that standardised large errors (larger than 2) suggest there could be a conflict between the emulator and the simulator, though we should expect 5% of points to fail this test if we have not been underconfident, and we may have extrapolation issues on the input space boundaries. In practice, if less than 5% of the errors are large and there is no systematic problem (e.g. all large errors are in the same region of parameter space) an emulator is considered to have "validated". It would then be used in history matching, calibration or for any other purpose by practitioners. Many applications using GPs now fit hundreds or thousands of emulators simultaneously [38,39], making detailed examination, beyond a quick check to see if the number of large errors is "about 5%", impractical.

Whilst the test described above may be adequate to assess the global performance of an emulator, when history matching the primary concern should be the local performance of the emulator within target NROY, $\mathcal{X}^*$. When the emulator at a given wave is incorrect outside target NROY, the worst thing that could happen is that a poor parameter choice is retained. However, this could still be removed by a future wave with a more accurate emulator. However, an emulator that is inaccurate within true NROY could lead to good choices being irrecoverably ruled out. As far as we are aware, this concern has not been addressed within the literature.

To fix ideas, we consider history matching on the 1D function considered by [40]. The function has the equation

$$y(x) = \sin\left(30(x-0.9)^4\right)\cos\left(2(x-0.9)\right) + \frac{(x-0.9)}{2}.$$

We use a 10-run maximin Latin Hypercube (LHC) [41] to design the runs to train the emulators used in this example.

Figure 1 shows the emulator performance and the results of the first wave of history matching, compared to the true NROY space found by the 1D model directly. The true function and the corresponding emulator posterior mean with uncertainty is shown in the top right panel. We can see that the region $[0, 0.4]$ is hard to predict by comparing the true function and emulator prediction. From the leave one out diagnostic plot (top left) we can see that the emulator has failed at one point, but that this single failure wouldn't be deemed serious enough to invalidate the emulator. The results of history matching with this emulator are shown in the bottom left and right panels (we set the threshold as 3). Comparing the true model calibration results with the emulator calibration results, we find nearly one third of the true NROY

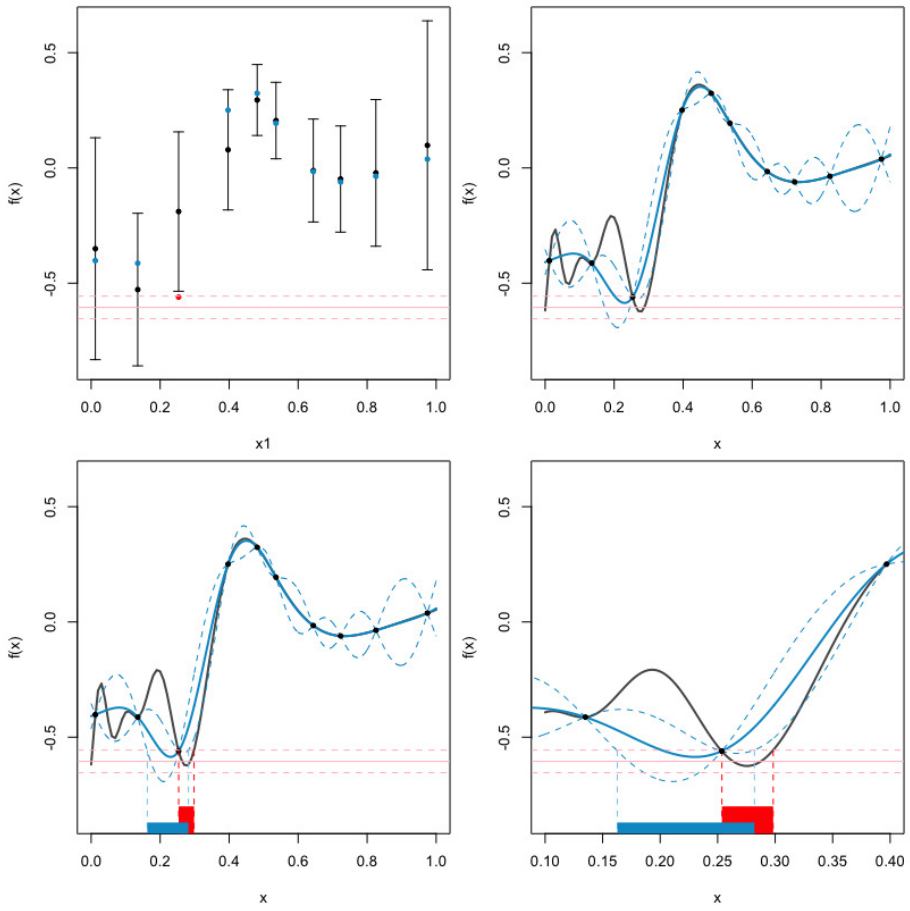

FIG. 1: *Top left:* Leave One Out diagnostic plot against x. The emulator prediction and two standard deviation intervals are given in black. The true function values are in blue if they lie within two standard deviation prediction intervals, or red otherwise. The pink line and the pair of red dotted lines present the observation with observation error and discrepancy in all 4 panels. *Top right:* Emulator performance for the 1D model. The true function is represented by the black curve and ten black points are inputs used to train the emulator. The blue line represents the emulator posterior mean, and the blue dotted lines give the two standard deviation prediction intervals. *Bottom left:* History matching results and the true NROY region. The blue interval defining the NROY space after first wave, the red Interval defining true NROY $\mathcal{X}^*$. *Bottom right:* As with bottom left but enlarged over the NROY regions.

space is ruled out.

In an application with many emulators being used to cut a high dimensional parameter space using many metrics, such critical cases may often occur and be difficult to catch. A trainable nugget would enable simpler functions to fit the data [35], and this might alleviate the problem in some cases, particularly if we have achieved what looks like an acceptable fit by over-fitting. We would generally use a trainable nugget when building emulators, for these reasons. However, in most cases where we see this pathology, the emulator fits well across the parameter space, but near the true NROY there is an issue which would not normally raise a diagnostic flag. In these cases, it is likely that the overall fit is good, but that there is some

local non-stationarity near where the function behaves like the data. In such cases, if a trainable nugget had not already been used, one would be unlikely to solve the problem and may lead to reduction in global performance (i.e. we may still have a good emulator from a validation perspective, but might rule out less space given that there will be a larger posterior variance). With or without a trainable nugget, we should still expect 5% of predicted points to lie outside our prediction intervals. If most or all of these occur near true NROY, we may still rule out good parts of that space by mistake.

The history matching literature usually recommends only ruling space out if 3 or more outputs have large implausibility, and this might insure against ruling out true NROY in some cases (if we have more than 1 or 2 metrics). However, a poor emulator near the true NROY region may often be a feature of the design that can appear for emulators of all metrics, and flagging this issue in a key region might make us wary of trusting emulators for other metrics in that region. Larger cutoff thresholds are sometimes used in earlier waves to retain more space until we are more confident about ruling out. If this is done routinely, it may still be that true NROY is ruled out using the larger threshold. If this is done to ensure that all points where there may be an issue are not ruled out, the cutoff may have to be set so high as to ensure that no space would be ruled out at all. In the following section we offer a method for detecting the type of history matching failure we have highlighted and then use it to present a method of robust history matching that makes use of the emulator we have in the regions where it performs well.

3. METHODOLOGY

3.1 Detection

Suppose we have already fitted a GP emulator. We may use our GP to compute the standardised errors, $D_i(f(x_i))$, given in equation(10), for design $\mathbf{X}$. Those errors that would normally be flagged as too large are grouped into what we term the 'failed' set,

$$\mathbf{X}_F = \{x_i \in \mathbf{X} | D_i(f(x_i)) > T_F\} \subseteq \mathbf{X}. \tag{11}$$

Here T_F is a threshold which is usually set as 2 (or even 1.96 with the argument that if the emulator were a good fit, 5% of these points should raise a flag). We treat $\mathbf{X}_F$ as a set of candidate points near which the emulator could be failing in such a way as to cut out regions of target NROY, $\mathcal{X}^*$. This could only happen if a point were inside or close to $\mathcal{X}^*$ and if, at the same time, the flag indicated a failure that would mean it

(and neighbouring space) was ruled out. For each point $x_m \in \mathbf{X}_F$, $f(\mathbf{x}_m)$ is compared with the observation z to discover whether the emulator failure points are near $\mathcal{X}^*$ and we form a set of 'doubt points' that could be close enough to target NROY to cause an issue. The doubt points set, $\mathbf{X}_D$, is defined using (8) via

$$\mathbf{X}_D = \{x_m \in \mathbf{X}_F \,||\, z - f(x_m)| \leq T\sqrt{\mathrm{Var}\left[e\right] + \mathrm{Var}\left[\eta\right]}\Big\}, \tag{12}$$

where T is the selected threshold, e is the observation error and η is the discrepancy. We define the set of remaining points $\mathbf{X}_N$, $\mathbf{X}_N = \mathbf{X}\backslash\mathbf{X}_D$.

Standard history matching can be applied directly if $\mathbf{X}_D$ is empty. Otherwise, in principle, with existing methods we might have to seek to add further runs from the computer model and/or find a more complex or bespoke emulation. The latter may not always be easy or even possible. Emulation and history matching are increasingly methods being adopted by modellers in order to calibrate their own models. Developing a bespoke emulator using a tailored kernel or mean function may be possible for UQ experts in any given problem, but it raises barriers to wide application in general that may not be necessary. Further model runs near $\mathbf{X}_D$ will likely enable standard methods to work well and fix the issue in many cases. However, in applications like climate modelling where running the model requires specialist equipment (e.g. supercomputers) and scientist time, it often the case that runs need to be done in batches and time/budget constraints mean that only a small number of batches will be available. Wasting one of these just to improve part of an emulator may sacrifice a whole potential wave of history matching.

Our method is based on the notion that the emulator works well enough in most of the parameter space so that it can be used for history matching anyway. However, in regions of space near $\mathbf{X}_D$, it would be safer not to remove space at all, and to resample that space in the next wave. Essentially, we will add further runs of our simulator to correct the errors in this region, but we will first cut out all of the space that can safely be cut out with the existing emulator. The goal then, when $\mathbf{X}_D$ is not empty, is to separate the whole input space into two regions, one containing $\mathbf{X}_D$ and the other containing $\mathbf{X}_N$ in such a way as to ensure that history matching in the latter region only will not discard parts of $\mathcal{X}^*$.

3.2 Local Voronoi Tessellation

There are several different approaches that can be employed to partition the input space into two distinct regions. One conventional approach is to use a classification method [42]. Logistic regression may be seen

as an obvious choice for classification [43]. However, in these problems $\mathbf{X}_D$ may contain only 1 or 2 points, meaning that our training data is a highly imbalanced dataset which is hard to use to train a logistic regression, or any similar model-based classifier. The typical results of these attempts tend to put most or all points in the 'normal' region and fail to adequately capture the doubt region.

A machine learning method Synthetic Minority Over-sampling Technique (SMOTE) can be used for classification on imbalanced datasets [44]. SMOTE uses synthetic data generation to increase the number of samples in the minority class so that the data set becomes balanced. SMOTE first finds the n-nearest neighbours in the minority class for each of the samples in the class, then random samples are generated on the lines between the neighbours. Though promising, SMOTE requires at least two points in $\mathbf{X}_D$ which in many instances will not apply.

A Voronoi tessellation is a partitioning of a space into convex cells called Voronoi regions [45]. Supposing that $\mathbf{X} = (x_1, \ldots, x_n)^T \in \mathcal{X}$ is a set of centres of an n-cell Voronoi tessellation, a Voronoi region, $\mathcal{V}_i$, is defined as the set of points in $\mathcal{X}$, whose 'nearest' point is x_i, so that

$$\mathcal{V}_i = \left\{ x \in \mathcal{X} \,\middle|\, d(x, x_i) \leq d(x, x_j) \right\}, \forall j \in \{1, \ldots, n\} \backslash i. \tag{13}$$

where $d(x, x_i)$ is commonly defined as the Euclidean distance. When history matching, our correlation function, $c(x, x')$, provides an appropriate notion of distance between inputs. Our n inputs $x \in \mathcal{X}$ can be used as the centres of a Voronoi tessellation. We cannot use the correlation directly (as the distance between the two points increases, the correlation decreases), therefore we define a Voronoi Tessellation $\mathcal{V}_i$ with the emulator posterior correlation function as

$$\mathcal{V}_i = \left\{ x \in \mathcal{X} \,\middle|\, |c^*(x, x_i)| \geq |c^*(x, x_j)| \right\}, \forall j \in \{1, \ldots, n\} \backslash i. \tag{14}$$

Finding a Voronoi tessellation can be computationally challenging when the design is large or when the input dimensions become even moderately sized (e.g. > 4). However, we do not need to map the whole parameter space. Our goal is to find the local Voronoi tiles that cover $\mathbf{X}_D$, ensuring that all possible values we have not run but might doubt our emulator for near true NROY are included. Specifically, a local Voronoi tessellation will partition the input space into a doubt region, $\mathcal{X}_D \supseteq \mathbf{X}_D$, and normal region, $\mathcal{X}_N \supseteq \mathbf{X}_N$, by finding $\mathcal{X}_D$.

We define a local Voronoi tessellation $\mathcal{X}_D = \bigcup_{\{i:x_i \in \mathbf{X}_D\}} \mathcal{V}_i$, with

$$\mathcal{V}_i = \left\{ x \in \mathcal{X} \,\middle|\, |c^*(x, x_i)| \geq |c^*(x, x_j)| \right\}, \forall j \text{ s.t. } x_j \in \mathbf{X}_N.$$

3.3 Local augmentation

The local Voronoi tessellation, $\mathcal{X}_D$, represents the union of convex sets around the doubt points. Given that the emulator failed to predict the doubt points, but was able to predict the surrounding normal points, we can deduce that there is a region between each normal point and each doubt point where the emulator is reliable (it predicts the normal points well) and a region near the doubt points where it is not. Though $\mathcal{X}_D$ will contain much of this region, if not all, there is no guarantee that it should contain the whole badly performing region. We therefore include an augmentation step to ensure that as much of the region where the emulator cannot be trusted (near target NROY) is included in $\mathcal{X}_D$.

For any design point x_i, the design point x_j with the largest value of $c^*(x_i, x_j)$ is the point with the most influence on x_i. For $x_i \in \mathbf{X}_N$, we want to ensure that their most influential points are not doubt points where we do not trust our emulator as this would indicate a possibility that some part of the region bordering $\mathcal{X}_D$ and near to x_j is unreliable. Our augmentation step adds all points from $\mathbf{X}_N$ with this property to $\mathbf{X}_D$ before arriving at a final $\mathcal{X}_D$.

Specifically, for each $x_i \in \mathbf{X}_N$, let

$$x_{k(i)} = \arg \max_{k:x_k \in \mathbf{X}} c^*(x_i, x_k).$$

Let $\mathbf{X}_{D'} = \left\{ x_i : x_{k(i)} \in \mathbf{X}_D \right\}$, the set of points in $\mathbf{X}_N$ whose most influential point is a doubt point. We then augment the doubt set by $\mathbf{X}_{D'}$ so that $\mathbf{X}_D = \mathbf{X}_D \cup \mathbf{X}_{D'}$, and compute the local Voronoi tessellation on the augmented set as before.

3.4 Robust history matching

Having isolated a region of parameter space, $\mathcal{X}_N = \mathcal{X} \setminus \mathcal{X}_D$ in which we feel confident enough in our emulator to rule out parameter space, we can history match in just that region. Specifically, we define

$$\mathcal{X}'_N = \{x \in \mathcal{X}_N : \mathcal{I}(x) \leq T\} \tag{15}$$

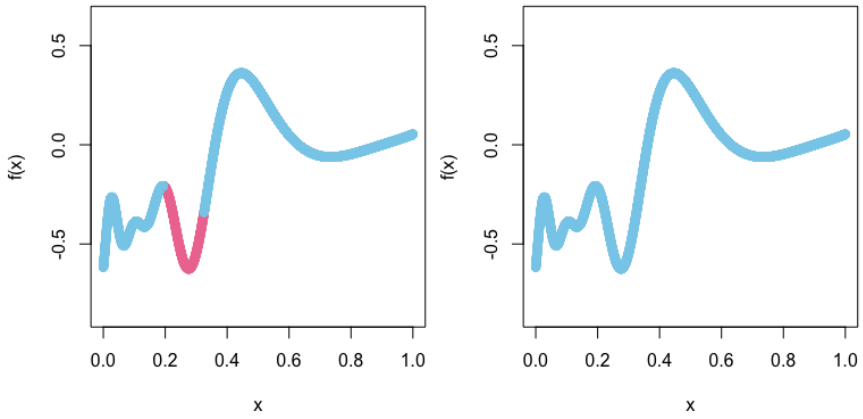

FIG. 2: Comparison between logistic regression classification (right), and Voronoi Tessellation with the GP emulator correlation prior function (left). The blue part is the normal region which can be employed in history matching. The red part is the retained doubt region.

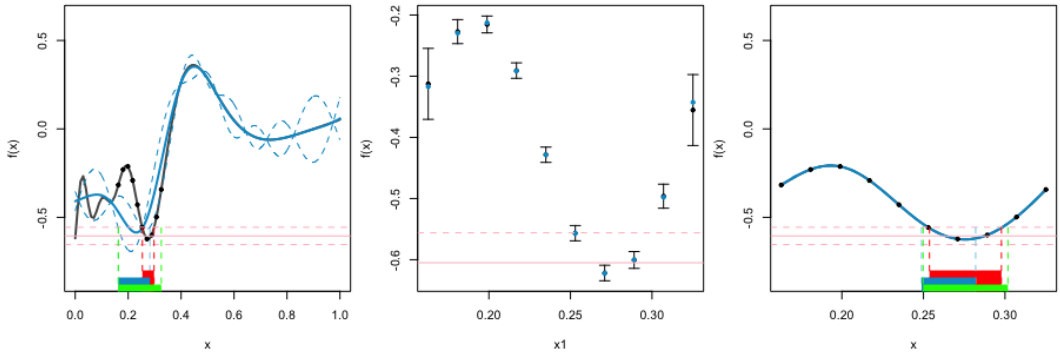

FIG. 3: The 1-dimensional model multi-wave calibration result. Left: history matching with our method first wave result, the red interval defining the true NROY space, the blue interval defining the NROY space by standrad history matching and the green interval defining the the NROY space by our method. Centre: leave One Out diagnostic plot against x for second wave emulator. Right: history matching second wave result.

with $\mathcal{I}(x)$ as in equation(5). The NROY space $\mathcal{X}^1$ after wave 1 is defined as

$$\mathcal{X}^1 = \mathcal{X}_D \cup \mathcal{X}'_N. \tag{16}$$

4. NUMERICAL EXAMPLES

4.1 The 1-dimensional function

We apply the methodology of the last section to the 1-dimensional function from Section 2.3 and a 5-dimensional function described below. We use the R package `DiceKriging` [46] to construct the emulators throughout.

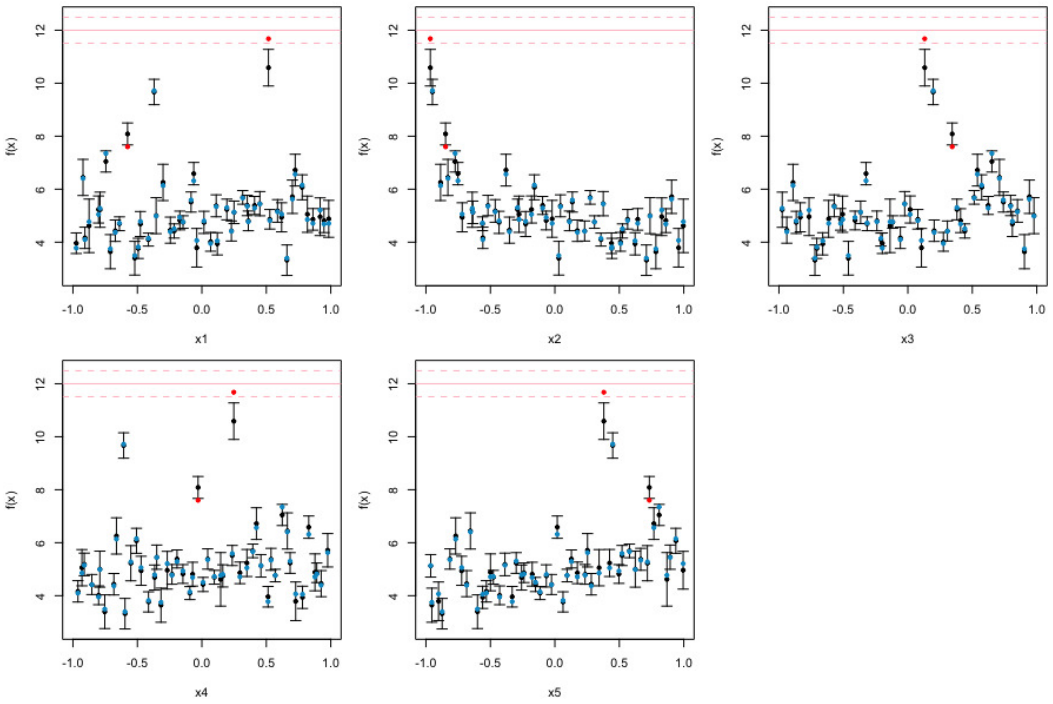

FIG. 4: Leave one out diagnostic plots. Each panel represents leave one out predictions from an emulator against one of the 5 inputs. Black points and error bars are from the emulator posterior mean and two standard deviation prediction intervals. The true function values are in blue if they lie within two standard deviation prediction intervals, or red otherwise.

The 1D model of Section 2.3 only has 1 doubt point. The doubt region highlighted by our local Voronoi tessellation is highlighted in red on the left hand panel of Figure 2. The right hand panel shows that the logistic regression classifier fails to identify any doubt region due to the unbalanced design.

We present our robust history match of this function in Figure 3. The wave 1 result is shown in the left panel with the green interval defining wave 1 NROY space. A second wave is performed with 10 randomly selected runs within NROY space and the leave one outs are shown in the middle panel of Figure 3, highlighting that there are no doubt points. The right panel shows the second wave results. We see that all of the target NROY space is retained.

4.2 A 5-dimensional function

In order to examine the performance of our method in higher dimensions we look at the 5D function

$$f(\mathbf{x}) = \sqrt{x_1} + \frac{1}{\sqrt{x_2}} + x_3 + \sin(x_4) + \exp(x_5).$$

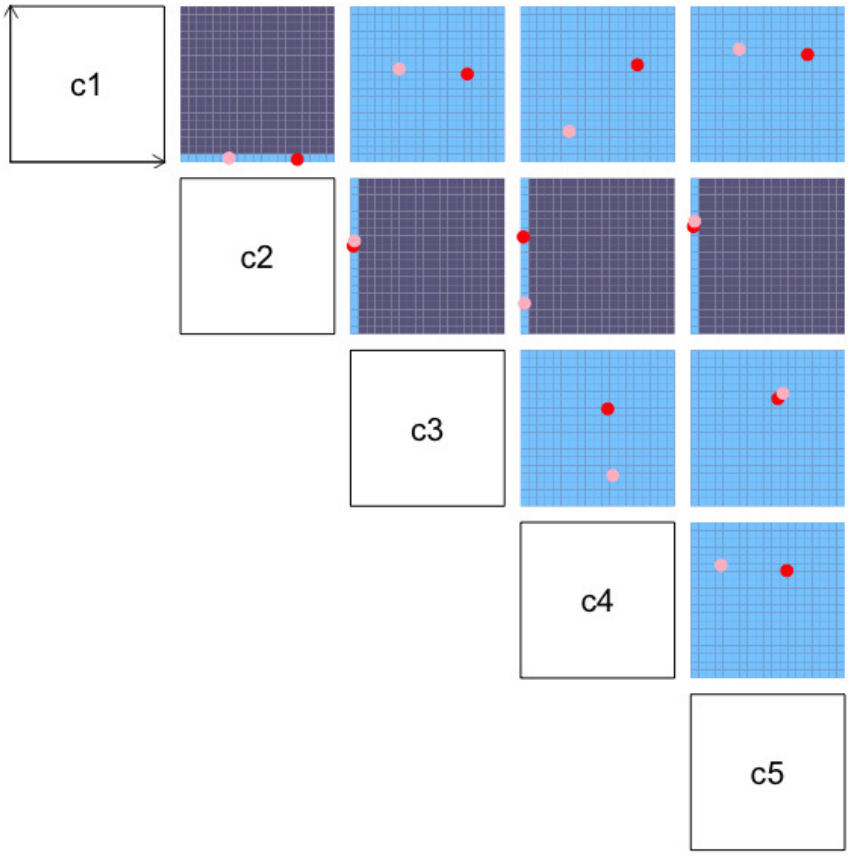

FIG. 5: Local Voronoi cell plots over each two parameters. The red point is the doubt point and the pink points are selected by augmentation step. The light blue region is the Local Voronoi cell of the doubt points which is the doubt region.

Note that this function tends to infinity as x_2 tends to zero which may happen in many physical models. We use a 50 member maximin Latin Hypercube (LHC) to select points for wave 1 and use the function evaluations to construct an emulator.

Leave one out diagnostics against each input are presented in Figure 4. By eye, we see that the emulator has individual large errors near the observations, which might indicate that the emulator does not simulate the target 'NROY' space effectively. Using equation(12) we identified 1 doubt point. The local Voronoi tessellation plot for all inputs is presented in Figure 5. The red point is the doubt point and the pink point is selected by the augmentation step. The light blue range is the Local Voronoi tessellation. We apply the robust history matching algorithm described in Section 3.4, retaining the local Voronoi tiles as part of NROY space and applying the usual constraint to the rest of the space.

TABLE 1: Standard vs robust history matching with top row as the percentage of the original space as NROY and the bottom the percentage of target NROY retained.

	Standard HM wave 1	Robust HM wave 1	Robust HM wave 2	Robust HM wave 3
Retained NROY volume	0.6373%	4.6660%	3.9132%	0.2251 %
Retained target NROY %	24.755%	99.643%	99.643%	99.449%

We compare our robust method with standard history matching in Figure 6. In these density plots, each pixel on any panel represents the proportion of points behind that pixel in the other 3 dimensions of the parameter space that is NROY. The scale corresponds to the colours in the upper triangles, whilst plots on the lower triangle mirror the upper triangle but with independent scales so as to reveal any structure hidden by the comparative colour scheme.

The top left panel in Figure 6 shows the target NROY space and the top right panel in Figure 6 shows the wave 1 NROY space following standard history matching. The first wave has incorrectly cut out a large corner of the target region (low x_1 and low x_2, x_3, x_4 and the lower half of x_5). Wave 1 of robust history matching, shown in Figure 6 (bottom left panel), does not have this issue and cuts out less parameter space overall (as expected). We continue to perform robust history matching for 2 further waves, though in waves 2 and 3, there were no doubt points, so these waves are the same as standard history matching (but from a different wave 1). The wave 3 NROY space is shown in Figure 6 (bottom right panel).

Table 1 shows the volume of NROY space as a percentage of the original space (top row) and the percentage of target NROY retained following each wave of history matching (bottom row). Target NROY is 0.09% of the original space. Though standard history matching cuts more space than our robust method in wave 1, it cuts out nearly 75% of target NROY, whilst we only cut 0.2%. After 2 further waves of history matching, we have still retained the target NROY, but have reduced our NROY to 0.17% of the original space.

This example shows a case where history matching can be non-robust in 5 dimensions and that our robust history matching effectively enables us to continue the analysis, without having to run a new wave 1. We now show a case from our work tuning climate models where this issue has presented itself and how our method performs.

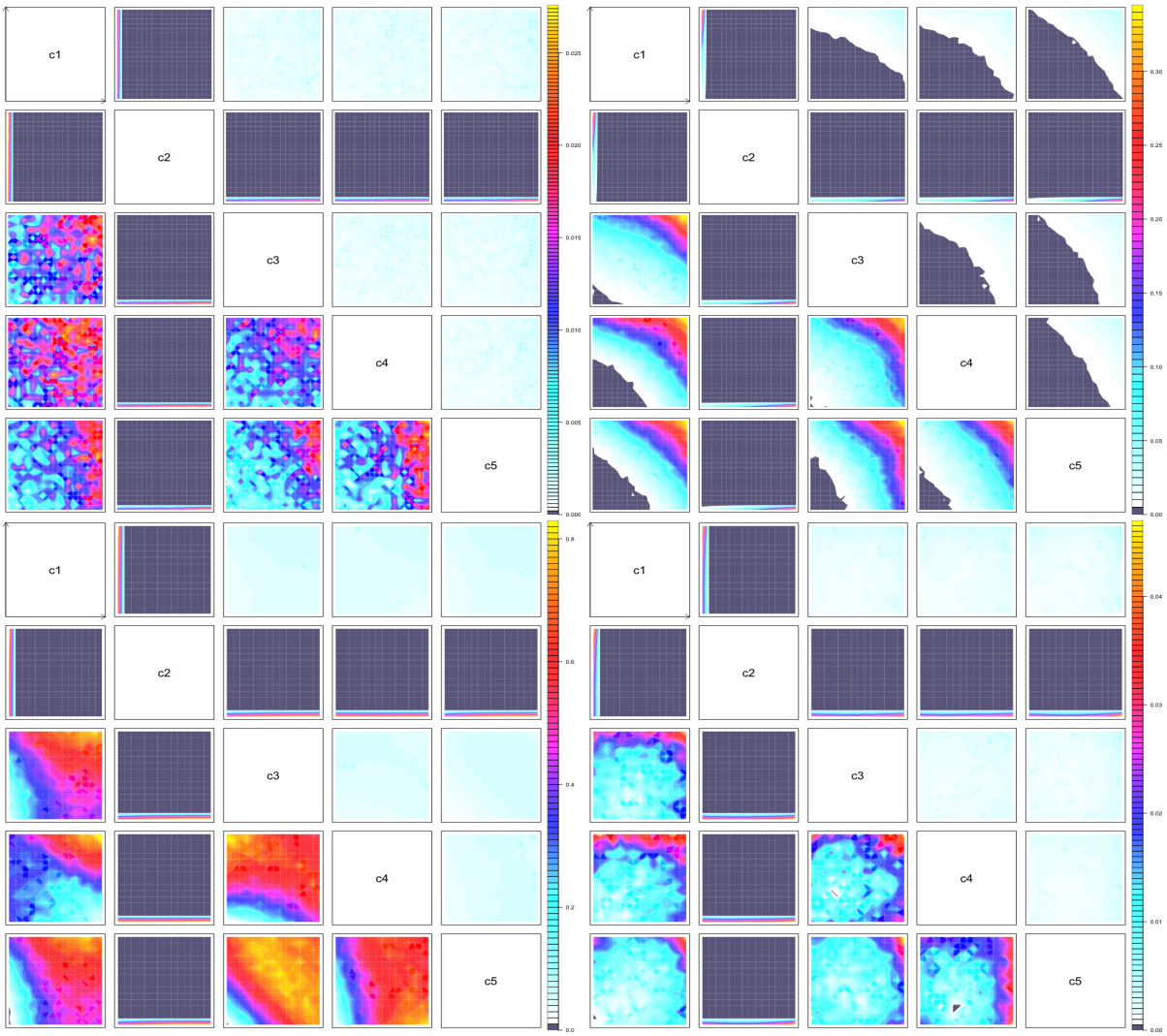

FIG. 6: NROY density plots for 2-D projections of NROY space. *Top left:* Target NROY space. *Top right:* Wave 1 NROY space following standard history matching. *Bottom left:* Wave 1 NROY space following robust history matching. *Bottom right:* Wave 3 NROY space after robust history matching. The scale corresponds to the colours in the upper triangles, whilst plots on the lower triangle mirror the upper triangle but with independent scales so as to reveal any structure hidden by the comparative colour scheme (the change from light blue, blue to red indicates that the density is rising).

5. APPLICATION: PROCESS-BASED TUNING OF CLIMATE MODELS

As part of the ANR (The French National Research Agency) funded project HIGH-TUNE, developers of the French climate models CNRM-CM and IPSL-CM are developing tools to automatically tune boundary layer cloud parameterisations within their models based on history matching to high resolution Large Eddy Simulations. Our collaboration involves providing methods to emulate and history match to a large number of process-based metrics quickly and automatically, so that the modellers can use the tools independently. With multiple unsupervised history matches, it is important that our methods are robust to possible ensemble issues, and so the method we describe in this paper should be part of our set of tools. We illustrate its importance through an example of a metric that fails our tests in IPSL-CM.

IPSL-CM is an atmosphere model that is used to predict planetary atmospheres, including the Earth and other celestial bodies (Mars, Titan, Venus), as well as regional climate, process studies [2,27,28]. We run a single column version of the model and perturb 5 cloud parameters chosen by the modellers. The model is run for a particular boundary layer case (in this case SANDU, capturing transitions from cumulus to stratocumulus clouds) with the idea of seeing which parameter choices lead to reasonable representations of clouds in these region types (compared to high-resolution simulations). Parameter ranges were determined by the project, and in our analysis we have mapped the parameters onto $[-1, 1]$ for fitting emulators and history matching.

We generate a 30-member design as the first 2 LHCs in a 150-member extended LHC composed of 10, 15-member LHCs following [22] (each additional LHC ensures that the composite design is orthogonal and fills the space in each extension phase). Leave one out diagnostic plots for our fitted emulator are presented in the top of Figure 7. To history match, we use an observation of 12.18, the observation error variance and discrepancy variance are both set as 0.0006.

There are 2 failed points near the observation, which might indicate that the emulator does not simulate the target NROY space well. Using equation(12) we identify 1 doubt point and another doubt point defined by our augmentation step. Since the target NROY is unknown in the climate model, in order to fairly compare our method with standard history matching, we use the remaining 120 data points (from our 150 member LHC) as validation data. The validation results are shown on Figure 7 in the middle and bottom rows. In this small data set, we have 11 points in target NROY space, the standard history matching misses one target point after wave 1, our method retains all the true NROY. In order to fairly compare, we do three waves with each methodology.

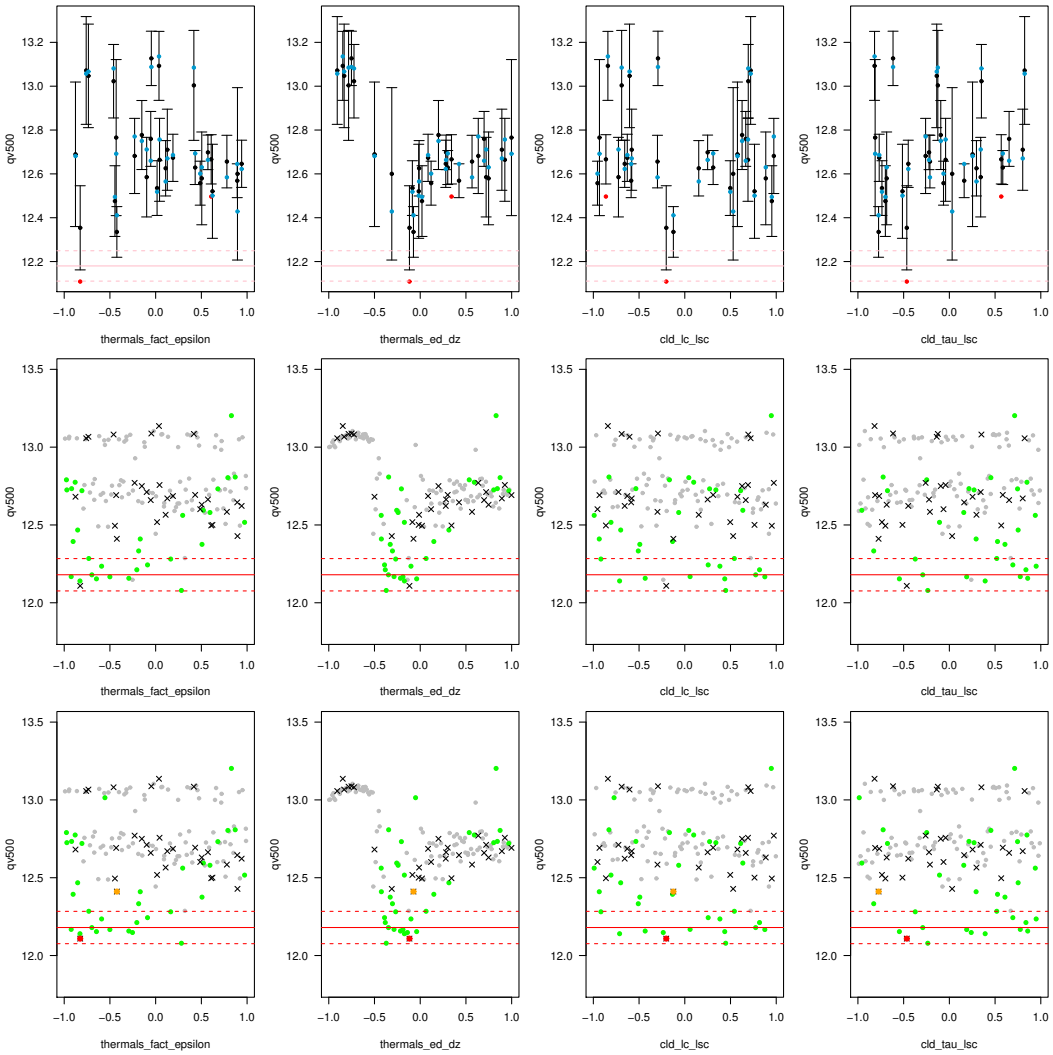

FIG. 7: *Top:* Leave one out diagnostic plots. Each panel represents one left-out latin hypercube predicted by the emulator, black points and error bar are from the emulator posterior mean and two standard deviation prediction intervals. The true function values are in blue if they lie within two standard deviation prediction intervals, or red otherwise. The observation with observation error are in solid and dotted red lines respectively. *Middle:* Validation results after wave 1 following standard history matching. All the points are model runs with the emulator training data presented in black. The validation data are green if they are retained in the NROY after wave 1 history matching, or grey otherwise. *Bottom:* Validation results after wave1 following robust history matching. The colours are as for the middle row with the red point being the original doubt point and the orange point, the doubt point selected by our augmentation step.

The NROY density plots (upper triangle) and minimum implausibility plots (lower triangle) are presented in Figures 8, 9, 10 and 11 (upper right), for each pair of parameters. For the NROY density plots, each pixel on any panel represents the proportion of points behind that pixel in the other 2 dimensions of the parameter space that is NROY. The right scale corresponds to the colours in the NROY density plots.

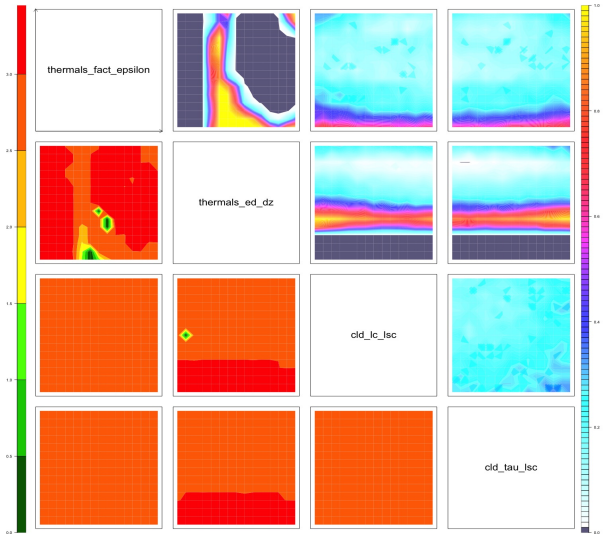

FIG. 8: Wave 1 NROY space for LMDZ-SANDU after robust history matching.

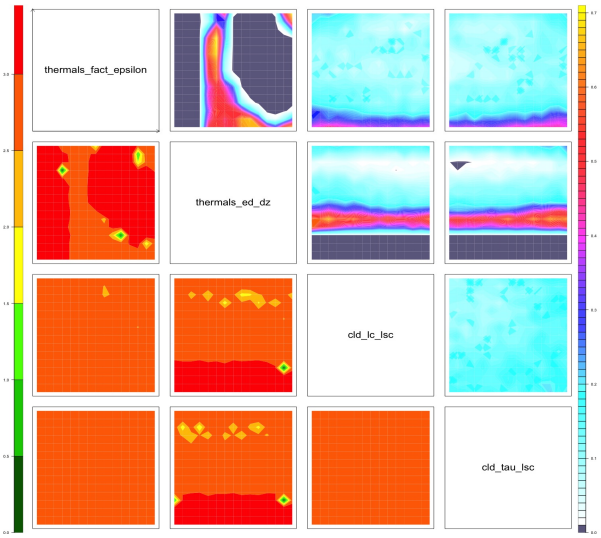

FIG. 9: Wave 3 NROY space for LMDZ-SANDU after robust history matching.

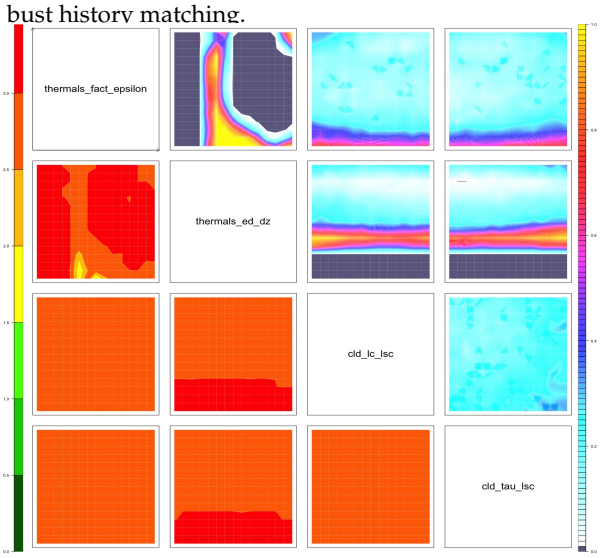

FIG. 10: Wave 1 NROY space for LMDZ-SANDU following standard history matching.

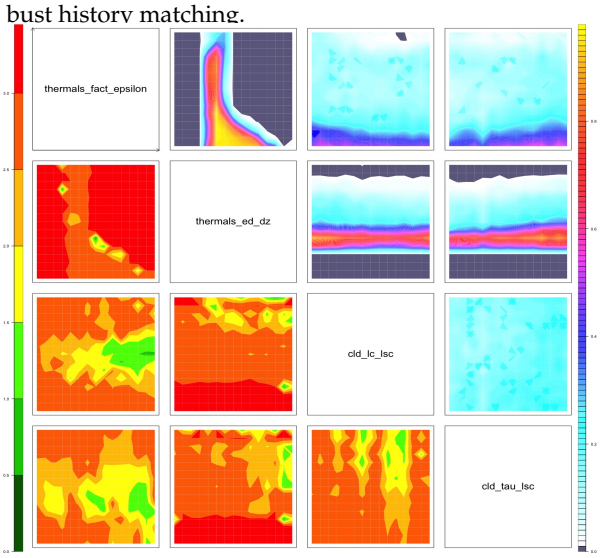

FIG. 11: Wave 3 NROY space for LMDZ-SANDU following standard history matching.

For the minimum implausibility plots, each pixel on any panel on the lower triangle represents the minimum implausibility found in the remaining dimensions of parameters space behind that pixel. The colours are given indicated by the scale on the left.

Only the first wave had doubt points, so wave 2 and wave 3 use standard history matching. Our method retains more space in the first wave (around 1%). We can see from Figure 10 that this retained space is in the centre of the space spanned by `thermals_fact_epsilon` and `thermals_ed_dz`. The wave 3 NROY density plot of robust history matching shows the doubt area is still in the NROY space, showing that standard history matching incorrectly ruled out part of $\mathcal{X}^*$.

This application showed that incorrectly ruling out parameter space can occur in practice: in this case when history matching climate model parameterisations. For climate models in particular, this mistake could prove very costly as history matching is used to assess the quality of a given parametersiation or an alternative. If good models are accidentally discarded, the parameterisation or even the resolution of the model or its implementation might be needlessly changed, wasting time and resource for the modelling centre.

6. DISCUSSION

In this paper we demonstrated a potential issue with history matching that occurs when emulators that seem to validate well by most standard analyses, do not simulate the (unknown) target subspace well enough. We showed that this can lead to good parts of parameter space being ruled out unintentionally, and that existing methods, such as using nuggets, variable thresholds or only ruling out if multiple metrics fail, were not designed specifically for such pathological cases and do not necessarily address the problem. We developed a method for detecting these cases based on standard diagnostics. We then presented a robust history matching method based on using a tailored local Voronoi tessellation designed to capture the region where the emulator is not as good as it needs to be, and isolate it so that the rest of the input space can be pruned as normal, without having to re-run the simulator.

We demonstrated the efficacy of our method in comparison to standard history matching for 2 numerical examples designed to demonstrate the issue, and then applied the method to a process metric from a single column version of the French climate model LMDZ. We showed that, unlike standard history matching, our method manages to effectively cut parameter space whilst ensuring that the target space is preserved.

Whilst it may be possible to observe the diagnostic issue we have highlighted and to offer a bespoke history match for a particular quantity in any given application, this is not feasible in applications where tens, hundreds or even thousands of emulators are built and are to be compared with observations [see, e.g.15,38,39] as part of the calibration. We also want methods that do not require frequent intervention by an experienced statistician. Hence our robust method provides a way to safely and automatically isolate any regions of parameter space where it would be dangerous to history match with the current emulator, but still allows the same emulator to be used appropriately without requiring a bespoke analysis.

ACKNOWLEDGMENTS

Daniel Williamson was funded by an Alan Turing Fellowship. The authors gratefully acknowledge the support from Agence Nationale de la Recherche (ANR) (grant HIGH-TUNE ANR-16-CE01-0010), which funded the runs of IPSL-CM. The authors thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support and hospitality during the Uncertainty Quantification programme where work on this article was undertaken (EPSRC grant no EP/K032208/1).

REFERENCES

1. Gladstone, R.M., Lee, V., Rougier, J., Payne, A.J., Hellmer, H., Le Brocq, A., Shepherd, A., Edwards, T.L., Gregory, J., and Cornford, S.L., Calibrated prediction of pine island glacier retreat during the 21st and 22nd centuries with a coupled flowline model, *Earth and Planetary Science Letters*, 333:191–199, 2012.
2. Hourdin, F., Mauritsen, T., Gettelman, A., Golaz, J.C., Balaji, V., Duan, Q., Folini, D., Ji, D., Klocke, D., Qian, Y., , The art and science of climate model tuning, *Bulletin of the American Meteorological Society*, 98(3):589–602, 2017.
3. Sacks, J., Welch, W.J., Mitchell, T.J., and Wynn, H.P., Design and analysis of computer experiments, *Statistical science*, pp. 409–423, 1989.
4. Santner, T.J., Williams, B.J., and Notz, W.I., *The design and analysis of computer experiments*, Springer Science & Business Media, 2003.
5. Kennedy, M.C. and O'Hagan, A., Bayesian calibration of computer models, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.
6. Higdon, D., Gattiker, J., Williams, B., and Rightley, M., Computer model calibration using high-dimensional output, *Journal of the American Statistical Association*, 103(482):570–583, 2008.

7. Craig, P., Goldstein, M., Seheult, A., and Smith, J., Bayes linear strategies for matching hydrocarbon reservoir history, *Bayesian statistics*, 5:69–95, 1996.

8. Vernon, I., Liu, J., Goldstein, M., Rowe, J., Topping, J., and Lindsey, K., Bayesian uncertainty analysis for complex systems biology models: emulation, global parameter searches and evaluation of gene functions, *BMC systems biology*, 12(1):1, 2018.

9. Gong, Z., DiazDelaO, F., and Beer, M., Sampling schemes for history matching using subset simulation, In *Proceedings for the 1st International Conference on Uncertainty Quantification in Computational Sciences and Engineering*, 2017.

10. Jackson, S.E., Vernon, I., Liu, J., and Lindsey, K., Bayesian uncertainty analysis establishes the link between the parameter space of a complex model of hormonal crosstalk in arabidopsis root development and experimental measurements, *Stat. Appl. Genet. Mol. Biol*, 2018.

11. Craig, P.S., Goldstein, M., Seheult, A.H., and Smith, J.A. Pressure matching for hydrocarbon reservoirs: a case study in the use of bayes linear strategies for large computer experiments. In *Case studies in Bayesian statistics*, pp. 37–93. Springer, 1997.

12. Cumming, J.A. and Goldstein, M., Bayes linear uncertainty analysis for oil reservoirs based on multiscale computer experiments, *O'Hagan, West, AM (eds.) The Oxford Handbook of Applied Bayesian Analysis*, pp. 241–270, 2010.

13. Andrianakis, I., Vernon, I.R., McCreesh, N., McKinley, T.J., Oakley, J.E., Nsubuga, R.N., Goldstein, M., and White, R.G., Bayesian history matching of complex infectious disease models using emulation: a tutorial and a case study on hiv in uganda, *PLoS computational biology*, 11(1):e1003968, 2015.

14. Andrianakis, I., McCreesh, N., Vernon, I., McKinley, T.J., Oakley, J.E., Nsubuga, R.N., Goldstein, M., and White, R.G., Efficient history matching of a high dimensional individual-based hiv transmission model, *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):694–719, 2017.

15. Vernon, I., Goldstein, M., Bower, R.G., , Galaxy formation: a bayesian uncertainty analysis, *Bayesian Analysis*, 5(4):619–669, 2010.

16. Bower, R.G., Vernon, I., Goldstein, M., Benson, A., Lacey, C.G., Baugh, C.M., Cole, S., and Frenk, C., The parameter space of galaxy formation, *Monthly Notices of the Royal Astronomical Society*, 407(4):2017–2045, 2010.

17. Rodrigues, L.F.S., Vernon, I., and Bower, R.G., Constraints on galaxy formation models from the galaxy stellar mass function and its evolution, *Monthly Notices of the Royal Astronomical Society*, 466(2):2418–2435, 2017.

18. Edwards, N.R., Cameron, D., and Rougier, J., Precalibrating an intermediate complexity climate model, *Climate dynamics*, 37(7-8):1469–1482, 2011.

19. McNeall, D., Challenor, P.G., Gattiker, J., and Stone, E., The potential of an observational data set for calibration of

a computationally expensive computer model, 2013.

20. Williamson, D., Goldstein, M., Allison, L., Blaker, A., Challenor, P., Jackson, L., and Yamazaki, K., History matching for exploring and reducing climate model parameter space using observations and a large perturbed physics ensemble, *Climate dynamics*, 41(7-8):1703–1729, 2013.
21. Williamson, D. and Blaker, A.T., Evolving bayesian emulators for structured chaotic time series, with application to large climate models, *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):1–28, 2014.
22. Williamson, D., Blaker, A.T., Hampton, C., and Salter, J., Identifying and removing structural biases in climate models with history matching, *Climate dynamics*, 45(5-6):1299–1324, 2015.
23. Salter, J.M. and Williamson, D., A comparison of statistical emulation methodologies for multi-wave calibration of environmental models, *Environmetrics*, 27(8):507–523, 2016.
24. Salter, J.M., Williamson, D.B., Scinocca, J., and Kharin, V., Uncertainty quantification for spatio-temporal computer models with calibration-optimal bases, *arXiv preprint arXiv:1801.08184*, 2018.
25. Williamson, D.B., Blaker, A.T., and Sinha, B., Tuning without over-tuning: parametric uncertainty quantification for the nemo ocean model, *Geoscientific Model Development*, 10(4):1789, 2017.
26. Bastos, L.S. and O'Hagan, A., Diagnostics for gaussian process emulators, *Technometrics*, 51(4):425–438, 2009.
27. Voldoire, A., Sanchez-Gomez, E., y Mélia, D.S., Decharme, B., Cassou, C., Sénési, S., Valcke, S., Beau, I., Alias, A., Chevallier, M., , The cnrm-cm5. 1 global climate model: description and basic evaluation, *Climate Dynamics*, 40(9-10):2091–2121, 2013.
28. Bony, S. and Dufresne, J.L., Marine boundary layer clouds at the heart of tropical cloud feedback uncertainties in climate models, *Geophysical Research Letters*, 32(20), 2005.
29. Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D., Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments, *Journal of the American Statistical Association*, 86(416):953–963, 1991.
30. Haylock, R. and O'Hagan, A., On inference for outputs of computationally expensive algorithms with uncertainty on the inputs, *Bayesian statistics*, 5:629–637, 1996.
31. Rasmussen, C.E. and Williams, C.K., *Gaussian processes for machine learning*, Vol. 1, MIT press Cambridge, 2006.
32. Andrianakis, I. and Challenor, P.G., The effect of the nugget on gaussian process emulators of computer models, *Computational Statistics & Data Analysis*, 56(12):4215–4228, 2012.
33. Volodina, V. and Williamson, D., Diagnostics-driven nonstationary emulators using kernel mixtures, *SIAM/ASA Journal on Uncertainty Quantification*, 8(1):1–26, 2020.

1 34. Oakley, J.E. and O'Hagan, A., Probabilistic sensitivity analysis of complex models: a bayesian approach, *Journal of*
2 *the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3):751–769, 2004.

3 35. Gramacy, R.B. and Lee, H.K., Cases for the nugget in modeling computer experiments, *Statistics and Computing*,
4 22(3):713–722, 2012.

5 36. Pukelsheim, F., The three sigma rule, *The American Statistician*, 48(2):88–91, 1994.

6 37. Coveney, S. and Clayton, R.H., Fitting two human atrial cell models to experimental data using bayesian history
7 matching, *Progress in biophysics and molecular biology*, 139:43–58, 2018.

8 38. Lee, L., Carslaw, K., Pringle, K., and Mann, G., Mapping the uncertainty in global ccn using emulation, *Atmospheric*
9 *Chemistry and Physics*, 12(20):9739–9751, 2012.

10 39. Gu, M., Berger, J.O., , Parallel partial gaussian process emulation for computer models with massive output, *The*
11 *Annals of Applied Statistics*, 10(3):1317–1347, 2016.

12 40. Xiong, Y., Chen, W., Apley, D., and Ding, X., A non-stationary covariance-based kriging method for metamodelling
13 in engineering design, *International Journal for Numerical Methods in Engineering*, 71(6):733–756, 2007.

14 41. Morris, M.D. and Mitchell, T.J., Exploratory designs for computational experiments, *Journal of statistical planning*
15 *and inference*, 43(3):381–402, 1995.

16 42. Bailey, K.D., *Typologies and taxonomies: an introduction to classification techniques*, Vol. 102, Sage, 1994.

17 43. Menard, S., *Applied logistic regression analysis*, Vol. 106, Sage, 2002.

18 44. Chawla, N.V., Bowyer, K.W., Hall, L.O., and Kegelmeyer, W.P., Smote: synthetic minority over-sampling technique,
19 *Journal of artificial intelligence research*, 16:321–357, 2002.

20 45. Gallier, J., Notes on convex sets, polytopes, polyhedra, combinatorial topology, voronoi diagrams and delaunay
21 triangulations, *arXiv preprint arXiv:0805.0292*, 2008.

22 46. Roustant, O., Ginsbourger, D., and Deville, Y., Dicekriging, diceoptim: Two r packages for the analysis of computer
23 experiments by kriging-based metamodeling and optimization, 2012.