Model predictivity assessment: incremental test-set selection and accuracy evaluation

Elias Fekhari, Bertrand Iooss, Joseph Muré, Luc Pronzato, Maria-João Rendas

Model predictivity assessment: incremental test-set selection and accuracy evaluation

Elias Fekhari, Bertrand Iooss, Joseph Muré, Luc Pronzato, and Maria-João Rendas

Abstract Unbiased assessment of the predictivity of models learnt by supervised machine learning (ML) methods requires knowledge of the learned function over a reserved test set (not used by the learning algorithm). The quality of the assessment depends, naturally, on the properties of the test set and on the error statistic used to estimate the prediction error. In this work we tackle both issues, proposing a new predictivity criterion that carefully weights the individual observed errors to obtain a global error estimate, and using incremental experimental design methods to "optimally" select the test points on which the criterion is computed. Several incremental constructions are studied, including greedy-packing (coffee-house design), support points and kernel herding techniques. Our results show that the incremental and weighted versions of the latter two, based on Maximum Mean Discrepancy concepts, yield superior performance. An industrial test case provided by the historical French electricity supplier (EDF) illustrates the practical relevance of the methodology, indicating that it is an efficient alternative to expensive cross-validation techniques.

Key words: Design of experiments, Discrepancy, Gaussian process, Machine learning, Metamodel, Validation

1 Introduction

The development of tools for automatic diagnosis relying on learned models imposes strict requirements on model validation. For example, in industrial non-destructive

Elias Fekhari · Bertrand Iooss* · Joseph Muré
EDF R&D, 6 Quai Watier, 78401 Chatou, France - e-mail: elias.fekhari@edf.fr, bertrand.iooss@edf.fr, joseph.mure@edf.fr - * Corresponding author, phone: +33130877969

Luc Pronzato · Maria-João Rendas
CNRS, Université Côte d'Azur, Laboratoire I3S, Bât. Euclide, Les Algorithmes, 2000 route des Lucioles, 06900 Sophia Antipolis cedex, France - e-mail: luc.pronzato@i3s.unice.fr, rendas@i3s.unice.fr

testing (e.g. for the aeronautic or the nuclear industry), generalized automated inspection, which increases efficiency and lowers costs, must provide high performance guarantees [14, 20]. Establishing these guarantees requires availability of a reserved test set, i.e. a data set that has not been used either to train or to select the machine learning (ML) model [3, 56, 21]. Using the prediction residuals on this test set, an independent evaluation of the proposed ML model can be done, enabling the estimation of relevant performance metrics, such as the mean-squared error for regression problems, or the misclassification rate for classification problems.

The same need for independent test sets arises in the area of computer experiments, where computationally expensive simulation codes are often advantageously replaced by ML models, called surrogate models (or metamodels) in this context [44, 15]. Such surrogate models can be used, for instance, to estimate the region of the input space that maps to specific values of the model outputs [7] with a significantly decreased computational load when compared to direct use of the original simulation code. Validation of these surrogate models consists in estimating their predictivity, and can either rely on a suitably selected validation sample, or be done by cross-validation [25, 22, 12]. One of the numerical studies presented in this paper will address an example of this situation of practical industrial interest in the domain of nuclear safety assessment, concerning the simulation of thermal-hydraulic phenomena inside nuclear pressurized water reactors, for which finely validated surrogate models have demonstrated their usefulness [28, 31].

In this paper, we present methods to choose a "good" test set, either within a given dataset or within the input space of the model, as recently motivated in [21, 23]. A first choice concerns the size of the test set. No optimal choice exists, and, when only a finite dataset is available, classical ML handbooks [19, 17] provide different heuristics on how to split it, e.g., 80%/20% between the training and test samples, or 50%/25%/25% between the training, validation (used for model selection) and test samples. We shall not formally address this point here (see [56] for a numerical study of this issue), but in the industrial case-study mentioned above we do study the impact of the ratio between the sizes of the training and test sets on the ability of assessing the quality of the surrogate model. A second issue concerns how the test sample is picked within the input space. The simplest – and most common – way to build a test sample is to extract an independent Monte Carlo sample [19]. For small test sets, these randomly chosen points may fall too close to the training points or leave large areas of the input space unsampled, and a more constructive method to select points inside the input domain is therefore preferable. Similar concerns motivate the use of space-filling designs when choosing a small set of runs for cpu-time expensive computer experiments on which a model will be identified [15, 38].

When the test set must be a subset of an initial dataset, the problem amounts to selecting a certain number of points within a finite collection of points. A review of classical methods for solving this issue is given in [3]. For example, the CADEX and DUPLEX algorithms [24, 50] can sequentially extract points from a database to include them in a test sample, using an inter-point distance criterion. In ML, identifying within the dataset "prototypes" (set of data instances representative of the whole data set) and "criticisms" (data instances poorly represented by the prototypes)

has recently been proposed to help model interpretation [32]; the extraction of prototypes and criticisms relies on a Maximum Mean Discrepancy (MMD) criterion [49] (see e.g. [40], and [37] for a performance analysis of greedy algorithms for MMD minimization).

Several algorithms have also been proposed for the case where points need to be added to an already existing training sample. When the goal is to assess the quality of a model learnt using a known training set, one may be tempted to locate the test points the furthest away from the training samples, such that, in some sense, the union of the training and test sets is space-filling. As this paper shows, test sets built in this manner do enable a good assessment of the quality of models learnt with the training set if the observed residuals are appropriately weighted. Moreover, the incremental augmentation of a design can be useful when the assessed model turns out to be of poor quality, or when an additional computational budget is available after a first study [47, 46]. Different empirical strategies have been proposed for incremental space-filling design [22, 8, 27], which basically entail the addition of new points in the zones poorly covered by the current design. Shang and Apley [46] have recently proposed an improvement of the CADEX algorithm, called the Fully-sequential space-filling (FSSF) design; see also [36] for an alternative version of coffee-house design enforcing boundary avoidance. Although they are developed for different purposes, nested space filling designs [41] and sliced space filling designs [42] can also be used to build sequential designs.

In this work, we provide new insights into these subjects in two main directions: (*i*) definition of new predictivity criteria through an optimal weighting of the test points residuals, and (*ii*) use of test sets built by incremental space-filling algorithms, namely FSSF, support points [29] and kernel herding [6], the latter two algorithms being typically used to provide a representative sample of a desired theoretical or empirical distribution. Besides, this paper presents a numerical benchmark analysis comparing the behaviour of the three algorithms on a selected set of test cases.

This paper is organized as follows. Section 2 defines the predictivity criterion considered and proposes different methods for its estimation. Section 3 presents the three algorithms used for test-point selection: FSSF, support points and kernel herding. Our numerical results are presented in Sections 4 and 5: in Section 4 a test set is freely chosen within the entire input space, while in Section 5 an existing data set can be split into a training sample and a test set. Finally, Section 6 concludes and outlines some perspectives.

2 Predictivity assessment criteria for an ML model

In this section, we propose a new criterion to assess the predictive performance of a model, derived from a standard model quality metric by suitably weighting the errors observed on the test set. We denote by $\mathcal{X} \subset \mathbb{R}^d$ the space of the input variables $\mathbf{x} = (x_1, \ldots, x_d)$ of the model. Then let $y(\mathbf{x}) \in \mathbb{R}$ (resp. $y(\mathbf{x}') \in \mathbb{R}$) be the observed output at point $\mathbf{x} \in \mathcal{X}$ (resp. $\mathbf{x}' \in \mathcal{X}$). We denote by $(\mathbf{X}_m, \mathbf{y}_m)$ the

training sample, with $\mathbf{y}_m = [y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(m)})]^\top$. The test sample is denoted by $(\mathbf{X}_n, \mathbf{y}_n) = (\mathbf{x}^{(m+i)}, y(\mathbf{x}^{(m+i)}))_{1 \leq i \leq n}$.

2.1 The predictivity coefficient

Let $\eta_m(\mathbf{x})$ denote the prediction at point $\mathbf{x}$ of a model learned using $(\mathbf{X}_m, \mathbf{y}_m)$ [19, 43]. A classical measure for assessing the predictive ability of η_m, in order to evaluate its validity, is the predictivity coefficient. Let μ denote the measure that weights how comparatively important it is to accurately predict y over the different regions of $\mathcal{X}$. For example the input could be a random vector with known distribution: in that case, this distribution would be a reasonable choice for μ. The true (ideal) value of the predictivity is defined as the following normalization of the Integrated Square Error (ISE):

$$Q^2_{\text{ideal}}(\mu) = 1 - \frac{\text{ISE}_\mu(\mathbf{X}_m, \mathbf{y}_m)}{V_\mu}\,, \tag{1}$$

where

$$\text{ISE}_\mu(\mathbf{X}_m, \mathbf{y}_m) = \int_{\mathcal{X}} [y(\mathbf{x}) - \eta_m(\mathbf{x})]^2 \, \mathrm{d}\mu(\mathbf{x})\,,$$
$$V_\mu = \int_{\mathcal{X}} \left[y(\mathbf{x}) - \int_{\mathcal{X}} y(\mathbf{x}') d\mu(\mathbf{x}') \right]^2 \, \mathrm{d}\mu(\mathbf{x})\,.$$

The ideal predictivity $Q^2_{\text{ideal}}(\mu)$ is usually estimated by its empirical version calculated over the test sample $(\mathbf{X}_n, \mathbf{y}_n)$, see [10, p. 32]:

$$\widehat{Q}^2_n = 1 - \frac{\sum_{\mathbf{x} \in \mathbf{X}_n} [y(\mathbf{x}) - \eta_m(\mathbf{x})]^2}{\sum_{\mathbf{x} \in \mathbf{X}_n} \left[y(\mathbf{x}) - \overline{y}_n \right]^2}\,, \tag{2}$$

where $\overline{y}_n = (1/n) \sum_{i=1}^n y(\mathbf{x}^{(m+i)})$ denotes the empirical mean of the observations in the test sample. Note that the calculation of $\widehat{Q}^2_n$ only requires access to the predictor $\eta_m(\cdot)$. To compute $\widehat{Q}^2_n$, one does not need to know the training set which was used to build $\eta_m(\cdot)$. $\widehat{Q}^2_n$ is the coefficient of determination (a standard notion in parametric regression) common in prediction studies [25, 22], often called "Nash-Sutcliffe criterion" [35]: it compares the prediction errors obtained with the model η_m with those obtained when prediction equals the empirical mean of the observations. Thus, the closer $\widehat{Q}^2_n$ is to one, the more accurate the surrogate model is (for the test set considered). On the contrary, $\widehat{Q}^2_n$ close to zero (negative values are possible too) indicates poor predictions abilities, as there is little improvement compared to prediction by the simple empirical mean of the observations. The next section shows how a suitable weighting of the residual on the training sample may be key to improving the estimation of $\widehat{Q}^2_n$.

2.2 Weighting the test sample

Let $\xi_n = n^{-1} \sum_{i=1}^{n} \delta_{\mathbf{x}^{(i)}}$ be the empirical distribution of the prediction error, with $\delta_{\mathbf{x}}$ the Dirac measure at $\mathbf{x}$. In $\widehat{Q}_n^2$, $\mathrm{ISE}_\mu(\mathbf{X}_m, \mathbf{y}_m)$ is estimated by the empirical average of the squared residuals

$$\mathrm{ISE}_{\xi_n}(\mathbf{X}_m, \mathbf{y}_m) = \frac{1}{n} \sum_{i=1}^{n} \left[y(\mathbf{x}^{(m+i)}) - \eta_m(\mathbf{x}^{(m+i)}) \right]^2 .$$

When the points $\mathbf{x}^{(m+i)}$ of the test set $\mathbf{X}_n$ are distant from the points of the training set $\mathbf{X}_m$, the squared prediction errors $|y(\mathbf{x}^{(m+i)}) - \eta_m(\mathbf{x}^{(m+i)})|^2$ tend to represent the worst possible error situations, and $\mathrm{ISE}_{\xi_n}(\mathbf{X}_m, \mathbf{y}_m)$ tends to overestimate $\mathrm{ISE}_\mu(\mathbf{X}_m, \mathbf{y}_m)$. In this section, we postulate a statistical model for the prediction errors in order to be able to quantify this potential bias when sampling the residual process, enabling its subsequent correction .

In [39], the authors propose a weighting scheme for the test set when the ML model interpolates the train set observations. They suggest several variants corresponding to different constraints on the weights (e.g., non-negativity, summing to one). In the following, we consider the unconstrained version only, which in our experience works best. Let $\delta_m(\mathbf{x}) = y(\mathbf{x}) - \eta_m(\mathbf{x})$ denote the predictor error and assume it is a realization of a Gaussian Process (GP) with zero mean and covariance kernel $\sigma^2 K_{|m}$, which we shall note $\delta_m(\mathbf{x}) \sim \mathsf{GP}(0, \sigma^2 K_{|m})$, with

$$\sigma^2 K_{|m}(\mathbf{x}, \mathbf{x}') = \mathbb{E}\{\delta_m(\mathbf{x})\delta_m(\mathbf{x}')\} = \sigma^2 \left[K(\mathbf{x}, \mathbf{x}') - \mathbf{k}_m^\top(\mathbf{x}) \mathbf{K}_m^{-1} \mathbf{k}_m(\mathbf{x}') \right] .$$

Here, $\mathbf{k}_m(\mathbf{x})$ denotes the column vector $[K(\mathbf{x}, \mathbf{x}^{(1)}), \ldots, K(\mathbf{x}, \mathbf{x}^{(m)})]^\top$ and $\mathbf{K}_m$ is the $m \times m$ matrix whose element (i, j) is given by $\{\mathbf{K}_m\}_{i,j} = K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, with K a positive definite kernel. The rationale for using this model is simple: assume first a prior GP model $\mathsf{GP}(0, \sigma^2 K)$ for the error process $\delta(\mathbf{x})$; if η_m interpolates the observations $\mathbf{y}_m$, the errors observed at the design points $\mathbf{x}^{(i)}$ equal zero, $i = 1, \ldots, m$, leading finally to the posterior $\mathsf{GP}(0, \sigma^2 K_{|m})$ for $\delta_m(\mathbf{x})$.

However, the predictor η_m is not always an interpolator, see Section 5 for an example, so we extend the approach of [39] to the general situation where η_m does not necessarily interpolate the training data $\mathbf{y}_m$. The same prior $\mathsf{GP}(0, \sigma^2 K)$ for $\delta(\mathbf{x})$ yields $\delta_m(\mathbf{x}) \sim \mathsf{GP}(\widehat{\delta}_m(\mathbf{x}), \sigma^2 K_{|m})$, where

$$\widehat{\delta}_m(\mathbf{x}) = \mathbf{k}_m^\top(\mathbf{x}) \mathbf{K}_m^{-1} (\mathbf{y}_m - \eta_m) \tag{3}$$

is the Kriging interpolator for the errors, with $\eta_m = [\eta_m(\mathbf{x}^{(1)}), \ldots, \eta_m(\mathbf{x}^{(m)})]^\top$.

The model above allows us to study how well $\mathrm{ISE}_\mu(\mathbf{X}_m, \mathbf{y}_m)$ is estimated using a given test set. Denote by $\overline{\Delta}^2(\xi_n, \mu; \mathbf{X}_m, \mathbf{y}_m)$ the expected squared error when estimating $\mathrm{ISE}_\mu(\mathbf{X}_m, \mathbf{y}_m)$ by $\mathrm{ISE}_{\xi_n}(\mathbf{X}_m, \mathbf{y}_m)$,

$$\begin{aligned}\overline{\Delta}^2(\xi_n,\mu;\mathbf{X}_m,\mathbf{y}_m) &= \mathbb{E}\left\{\left[\mathrm{ISE}_{\xi_n}(\mathbf{X}_m,\mathbf{y}_m)-\mathrm{ISE}_{\mu}(\mathbf{X}_m,\mathbf{y}_m)\right]^2\right\}\\ &= \mathbb{E}\left\{\left[\int_{\mathcal{X}}\delta_m^2(\mathbf{x})\,\mathrm{d}(\xi_n-\mu)(\mathbf{x})\right]^2\right\}\\ &= \mathbb{E}\left\{\int_{\mathcal{X}^2}\delta_m^2(\mathbf{x})\delta_m^2(\mathbf{x}')\,\mathrm{d}(\xi_n-\mu)(\mathbf{x})\mathrm{d}(\xi_n-\mu)(\mathbf{x}')\right\}.\end{aligned}$$

Tonelli's theorem gives

$$\overline{\Delta}^2(\xi_n,\mu;\mathbf{X}_m,\mathbf{y}_m) = \int_{\mathcal{X}^2}\mathbb{E}\{\delta_m^2(\mathbf{x})\delta_m^2(\mathbf{x}')\}\,\mathrm{d}(\xi_n-\mu)(\mathbf{x})\mathrm{d}(\xi_n-\mu)(\mathbf{x}')\,.$$

Since $\mathbb{E}\left\{U^2V^2\right\} = 2\,(\mathbb{E}\{UV\})^2 + \mathbb{E}\left\{U^2\right\}\mathbb{E}\left\{V^2\right\}$ for any one-dimensional normal centered random variables U and V, when $\eta_m(\mathbf{x})$ interpolates $\mathbf{y}_m$, we obtain

$$\overline{\Delta}^2(\xi_n,\mu;\mathbf{X}_m,\mathbf{y}_m) = \sigma^2\, d^2_{\overline{K}_{|m}}(\xi_n,\mu)\,, \tag{4}$$

where

$$\begin{aligned}d^2_{\overline{K}_{|m}}(\xi_n,\mu) &= \int_{\mathcal{X}^2}\overline{K}_{|m}(\mathbf{x},\mathbf{x}')\,\mathrm{d}(\xi_n-\mu)(\mathbf{x})\mathrm{d}(\xi_n-\mu)(\mathbf{x}')\,,\\ \overline{K}_{|m}(\mathbf{x},\mathbf{x}') &= 2\,K^2_{|m}(\mathbf{x},\mathbf{x}') + K_{|m}(\mathbf{x},\mathbf{x})K_{|m}(\mathbf{x}',\mathbf{x}')\,,\end{aligned} \tag{5}$$

and we recognize $d^2_{\overline{K}_{|m}}(\xi_n,\mu)$ as the squared Maximum-Mean-Discrepancy (MMD) between ξ_n and μ for the kernel $\overline{K}_{|m}$; see (18) in Appendix A. Note that σ^2 only appears as a multiplying factor in (4), with the consequence that σ^2 does not impact the choice of a suitable ξ_n.

When η_m does not interpolate $\mathbf{y}_m$, similar developments still give $\overline{\Delta}^2(\xi_n,\mu;\mathbf{X}_m,\mathbf{y}_m) = \sigma^2\, d^2_{\overline{K}_{|m}}(\xi_n,\mu)$, with now

$$\begin{aligned}\overline{K}_{|m}(\mathbf{x},\mathbf{x}') = 2&\left[K_{|m}(\mathbf{x},\mathbf{x}') + 2\,\widehat{\delta}_m(\mathbf{x})\widehat{\delta}_m(\mathbf{x}')\right]K_{|m}(\mathbf{x},\mathbf{x}')\\ &+\left[\widehat{\delta}^2_m(\mathbf{x}) + K_{|m}(\mathbf{x},\mathbf{x})\right]\left[\widehat{\delta}^2_m(\mathbf{x}') + K_{|m}(\mathbf{x}',\mathbf{x}')\right],\end{aligned}$$

where $\widehat{\delta}_m(\mathbf{x})$ is given by (3).

The idea is to replace ξ_n, uniform on $\mathbf{X}_n$, by a nonuniform measure ζ_n supported on $\mathbf{X}_n$, $\zeta_n = \sum_{i=1}^n w_i\delta_{\mathbf{x}^{(m+i)}}$ with weights $\mathbf{w}_n = (w_1,\ldots,w_n)^\top$ chosen such that the estimation error $\overline{\Delta}^2(\xi_n,\mu;\mathbf{X}_m,\mathbf{y}_m)$, and thus $d^2_{\overline{K}_{|m}}(\zeta_n,\mu)$, is minimized. Direct calculation gives

$$d^2_{\overline{K}_{|m}}(\zeta_n,\mu) = \mathscr{E}_{\overline{K}_{|m}}(\mu) - 2\,\mathbf{w}_n^\top\mathbf{p}_{\overline{K}_{|m},\mu}(\mathbf{X}_n) + \mathbf{w}_n^\top\overline{\mathbf{K}}_{|m}(\mathbf{X}_n)\mathbf{w}_n\,,$$

where $\mathbf{p}_{\overline{K}_{|m},\mu}(\mathbf{X}_n) = \left[P_{\overline{K}_{|m},\mu}(\mathbf{x}^{(m+1)}),\ldots,P_{\overline{K}_{|m},\mu}(\mathbf{x}^{(m+n)})\right]^\top$, with $P_{\overline{K}_{|m},\mu}(\mathbf{x})$ defined by (17) in Appendix A, $\{\overline{\mathbf{K}}_{|m}(\mathbf{X}_n)\}_{i,j} = \overline{K}_{|m}(\mathbf{x}^{(m+i)},\mathbf{x}^{(m+j)})$ for any $i,j = 1,\ldots,n$, and $\mathscr{E}_{\overline{K}_{|m}}(\mu) = \int_{\mathcal{X}^2} \overline{K}_{|m}(\mathbf{x},\mathbf{x}')\,\mathrm{d}\mu(\mathbf{x})\mathrm{d}\mu(\mathbf{x}')$.

When $\mathbf{X}_m \cap \mathbf{X}_n = \emptyset$, the $n \times n$ matrix $\mathbf{K}_{|m}(\mathbf{X}_n)$, whose element i,j equals $K_{|m}(\mathbf{x}^{(m+i)},\mathbf{x}^{(m+j)})$, is positive definite. The elementwise (Hadamard) product $\mathbf{K}_{|m}(\mathbf{X}_n) \circ \mathbf{K}_{|m}(\mathbf{X}_n)$ is thus positive definite too, implying that $\overline{\mathbf{K}}_{|m}(\mathbf{X}_n)$ is positive definite. The optimal weights $\mathbf{w}_n^*$ minimizing $d^2_{\overline{K}_{|m}}(\zeta_n,\mu)$ are thus

$$\mathbf{w}_n^* = \overline{\mathbf{K}}_{|m}^{-1}(\mathbf{X}_n)\mathbf{p}_{\overline{K}_{|m},\mu}(\mathbf{X}_n)\,. \tag{6}$$

We shall denote by ζ_n^* the measure supported on $\mathbf{X}_n$ with the optimal weights (6) and

$$\begin{aligned} Q^2_{n*} &= 1 - \frac{\mathrm{ISE}_{\zeta_n^*}(\mathbf{X}_m,\mathbf{y}_m)}{\frac{1}{n}\sum_{i=1}^n \left[y(\mathbf{x}^{(m+i)}) - \overline{y}_n\right]^2} \\ &= 1 - \frac{\sum_{i=1}^n w_i^* \left[y(\mathbf{x}^{(m+i)}) - \eta_m(\mathbf{x}^{(m+i)})\right]^2}{\frac{1}{n}\sum_{i=1}^n \left[y(\mathbf{x}^{(m+i)}) - \overline{y}_n\right]^2}\,, \end{aligned} \tag{7}$$

with $\overline{y}_n = (1/n)\sum_{i=1}^n y(\mathbf{x}^{(m+i)})$. Notice that the weights w_i^* do not depend on the variance parameter σ^2 of the GP model.

Remark 1 When $\mathbf{X}_n$ is constructed by kernel herding, see Section 3.3, K can be chosen identical to the kernel used there. This will be the case in Sections 4 and 5, but it is not mandatory.

Conversely, one may think of choosing a design $\mathbf{X}_n$ that minimizes $d^2_{\overline{K}_{|m}}(\xi_n,\mu)$, or $d^2_{\overline{K}_{|m}}(\zeta_n^*,\mu)$, also with the objective to obtain a precise estimation of $\mathrm{ISE}_\mu(\mathbf{X}_m,\mathbf{y}_m)$. This can be achieved by kernel herding using the kernel $\overline{K}_{|m}$ and is addressed in [39]. However, the numerical results presented there show that the precise choice of the test set $\mathbf{X}_n$ has a marginal effect compared to the effect of non-uniform weighting with $\mathbf{w}_n^*$, provided that $\mathbf{X}_n$ fills the holes left in $\mathcal{X}$ by the training design $\mathbf{X}_m$. ◁

Remark 2 When the observations $y(\mathbf{x}^{(i)})$, $i = 1,\ldots,n$, are available at the validation stage, an alternative version of $\widehat{Q}_n^2$ would be

$$\widehat{Q}_n'^2 = 1 - \frac{\sum_{\mathbf{x}\in\mathbf{X}_n}\left[y(\mathbf{x}) - \eta_m(\mathbf{x})\right]^2}{\sum_{\mathbf{x}\in\mathbf{X}_n}\left[y(\mathbf{x}) - \overline{y}_m\right]^2}\,, \tag{8}$$

where $\overline{y}_m = (1/m)\sum_{i=1}^m y(\mathbf{x}^i)$, which compares the performance on the test set of two predictors η_m and $\overline{y}_m$ based on the same training set. It is then possible to also apply a weighting procedure to the *denominator* of $\widehat{Q}_n'^2$,

$$D_{\xi_n}(\mathbf{X}_m, \mathbf{y}_m) = \frac{1}{n} \sum_{i=1}^{n} \left[y(\mathbf{x}^{(m+i)}) - \overline{y}_m \right]^2 ,$$

in order to make it resemble its idealized version $V'_\mu(\mathbf{y}_m) = \int_{\mathcal{X}} \left[y(\mathbf{x}) - \overline{y}_m \right]^2 \mathrm{d}\mu(\mathbf{x})$. The GP model is now $\varepsilon_m(\mathbf{x}) = y(\mathbf{x}) - \overline{y}_m \sim \mathsf{GP}(\eta_m(\mathbf{x}) - \overline{y}_m, \sigma^2 K_{|m})$. Similar developments to those used above for the numerator of $\widehat{Q}_n^2$ yield

$$\begin{aligned}
\overline{\Delta'}^2(\xi_n, \mu; \mathbf{X}_m, \mathbf{y}_m) &= \mathbb{E}\left\{ \left[D_{\xi_n}(\mathbf{X}_m, \mathbf{y}_m) - V_\mu(\mathbf{y}_m) \right]^2 \right\} \\
&= \mathbb{E}\left\{ \int_{\mathcal{X}^2} \varepsilon_m^2(\mathbf{x}) \varepsilon_m^2(\mathbf{x}') \, \mathrm{d}(\xi_n - \mu)(\mathbf{x}) \mathrm{d}(\xi_n - \mu)(\mathbf{x}') \right\} \\
&= \sigma^2 \, d^2_{\overline{K'}_{|m}}(\xi_n, \mu) ,
\end{aligned}$$

where

$$\begin{aligned}
\overline{K'}_{|m}(\mathbf{x}, \mathbf{x}') = \overline{K}_{|m}(\mathbf{x}, \mathbf{x}') &+ [\eta_m(\mathbf{x}) - \overline{\mathbf{y}}_m]^2 [\eta_m(\mathbf{x}') - \overline{\mathbf{y}}_m]^2 \\
&+ [\eta_m(\mathbf{x}) - \overline{\mathbf{y}}_m]^2 K_{|m}(\mathbf{x}', \mathbf{x}') + [\eta_m(\mathbf{x}') - \overline{\mathbf{y}}_m]^2 K_{|m}(\mathbf{x}, \mathbf{x}) \\
&+ 4 \, [\eta_m(\mathbf{x}) - \overline{\mathbf{y}}_m] [\eta_m(\mathbf{x}') - \overline{\mathbf{y}}_m] K_{|m}(\mathbf{x}, \mathbf{x}') .
\end{aligned}$$

We can then substitute $D_{\zeta_n'^*}(\mathbf{X}_m, \mathbf{y}_m)$ for $D_{\xi_n}(\mathbf{X}_m, \mathbf{y}_m)$ in (8), where $\zeta_n'^*$ allocates the weights $\mathbf{w}_n'^* = \overline{\mathbf{K}'}_{|m}^{-1}(\mathbf{X}_n) \mathbf{p}_{\overline{K'}_{|m}, \mu}(\mathbf{X}_n)$ to the n points in $\mathbf{X}_n$, with $\{\overline{\mathbf{K}'}_{|m}(\mathbf{X}_n)\}_{i,j} = \overline{K'}_{|m}(\mathbf{x}^{(m+i)}, \mathbf{x}^{(m+j)})$, $i, j = 1, \ldots, n$. ◁

3 Test-set construction

In the previous section we assumed the test set as given, and proposed a method to estimate $\mathrm{ISE}_\mu(\mathbf{X}_m, \mathbf{y}_m)$ by a weighted sum of the residuals. In this section we address the choice of the test set.

Below we give an overview of the three methods used in this paper, all relying on the concept of space-filling design [15, 38]. While most methods for the construction of such designs choose all points simultaneously, the methods we consider are incremental, selecting one point at a time.

Our objective is to construct an ordered test set of size n, denoted by $\mathbf{X}_n = \left\{ \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)} \right\} \subset \mathcal{X}$. When there is no restriction on the choice of $\mathbf{X}_n$, the advantage of using an incremental construction is that it can be stopped once the estimation of the predictivity of an initial model, built with some given design $\mathbf{X}_m$, is considered sufficiently accurate. In case the conclusion is that model predictions are not reliable enough, the full design $\mathbf{X}_{m+n} = \mathbf{X}_m \cup \mathbf{X}_n$ and the associated observations $\mathbf{y}_{m+n}$ can be used to update the model. This updated model can then be tested at additional design points, elements of a new test set to be constructed. All methods presented in this section (except the Fully Sequential Space-Filling method) are implemented in

the Python package `otkerneldesign` [1] which is based on the OpenTURNS library for uncertainty quantification [1].

3.1 Fully-Sequential Space-Filling design

The Fully-Sequential Space-Filling forward-reflected (FSSF-fr) algorithm [46] relies on the CADEX algorithm [24] (also called the "coffee-house" method [34]). It constructs a sequence of nested designs in a bounded set $\mathcal{X}$ by sequentially selecting a new point $\mathbf{x}$ as far away as possible from the $\mathbf{x}^{(i)}$ previously selected. New inserted points are selected within a set of candidates $\mathcal{S}$ which may coincide with $\mathcal{X}$ or be a finite subset of $\mathcal{X}$ (which simplifies the implementation, only this case is considered here). The improvement of FSSF-fr when compared to CADEX is that new points are selected *at the same time* far from the previous design points as well as far from the boundary of $\mathcal{X}$.

The algorithm is as follows:

1. Choose $\mathcal{S}$, a finite set of candidate points in $\mathcal{X}$, with size $N \gg n$ in order to allow a fairly dense covering of $\mathcal{X}$. When $\mathcal{X} = [0,1]^d$, [46] recommends to take $\mathcal{S}$ equal to the first $N = 1\,000\,d + 2\,n$ points of a Sobol sequence in $\mathcal{X}$.
2. Choose the first point $\mathbf{x}^{(1)}$ randomly in $\mathcal{S}$ and define $\mathbf{X}_1 = \{\mathbf{x}^{(1)}\}$.
3. At iteration i, with $\mathbf{X}_i = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(i)}\}$, select

$$\mathbf{x}^{(i+1)} \in \operatorname*{Arg\,max}_{\mathbf{x} \in \mathcal{S} \setminus \mathbf{X}_i} \left[\min \left(\min_{j \in \{1,\ldots,i\}} \|\mathbf{x} - \mathbf{x}^{(j)}\|, \sqrt{2}\, d\, \operatorname{dist}(\mathbf{x}, R(\mathbf{x})) \right) \right], \qquad (9)$$

   where $R(\mathbf{x})$ is the symmetric of $\mathbf{x}$ with respect to its nearest boundary of $\mathcal{X}$, and set $\mathbf{X}_{i+1} = \mathbf{X}_i \bigcup \mathbf{x}^{(i+1)}$.
4. Stop the algorithm when $\mathbf{X}_n$ has the required size.

The standard coffee-house (greedy packing) algorithm simply uses $\mathbf{x}^{(i+1)} \in \operatorname{Arg\,max}_{\mathbf{x} \in \mathcal{S} \setminus \mathbf{X}_i} \min_{j \in \{1,\ldots,i\}} \|\mathbf{x} - \mathbf{x}^{(j)}\|$. The role of the reflected point $R(\mathbf{x})$ is to avoid selecting $\mathbf{x}^{(i+1)}$ too close to the boundary of $\mathcal{X}$, which is a major problem with standard coffee-house, especially when $\mathcal{X} = [0,1]^d$ with d large. The factor $\sqrt{2}d$ in (9) proposed in [46] sets a balance between distance to the design $\mathbf{X}_i$ and distance to the boundary of $\mathcal{X}$. Another scaling factor, depending on the target design size n is proposed in [36].

FSSF-fr is entirely based on geometric considerations and implicitly assumes that the selected set of points should cover $\mathcal{X}$ evenly. However, in the context of uncertainty quantification [48] it frequently happens that the distribution μ of the model inputs is not uniform. It is then desirable to select a test set representative of μ. This can be achieved through the inverse probability integral transform: FSSF-fr constructs $\mathbf{X}_n$ in the unit hypercube $[0,1]^d$, and an "isoprobabilistic" transform $T : [0,1]^d \to \mathcal{X}$ is then applied to the points in $\mathbf{X}_i$, T being such that, if $\boldsymbol{U}$ is a

[1] https://pypi.org/project/otkerneldesign/

random variable uniform on $[0,1]^d$, then $T(\boldsymbol{U})$ follows the target distribution μ. The transformation can be applied to each input separately when μ is the product of its marginals, a situation considered in our second test-case of Section 4, but is more complicated in other cases, see [26, Chap. 4]. Note that FSSF-fr operates in the bounded set $[0,1]^d$ even if the support of μ is unbounded. The other two algorithms presented in this section are able to directly choose points representative of a given distribution μ and do not need to resort to such a transformation.

3.2 Support points

Support points [29] are such that their associated empirical distribution ξ_n has minimum Maximum-Mean-Discrepancy (MMD) with respect to μ for the energy-distance kernel of Székely and Rizzo [52, 53],

$$K_E(\mathbf{x},\mathbf{x}') = \frac{1}{2}\left(\|\mathbf{x}\| + \|\mathbf{x}'\| - \|\mathbf{x}-\mathbf{x}'\|\right) . \tag{10}$$

The squared MMD between ξ_n and μ for the distance kernel equals

$$d^2_{K_E}(\xi_n,\mu) = \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}\|\mathbf{x}^{(i)}-\zeta\| - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\| - \mathbb{E}\|\zeta-\zeta'\| , \tag{11}$$

where ζ and ζ' are independently distributed with μ; see [45]. A key property of the energy-distance kernel is that it is characteristic [51]: for any two probability distributions μ and ξ on $\mathcal{X}$, $d^2_{K_E}(\mu,\xi)$ equals zero if and only if $\mu = \xi$, and so it defines a norm in the space of probability distributions. Compared to more heuristic methods for solving quantization problems, support points benefit from the theoretical guarantees of MMD minimization in terms of convergence of ξ_n to μ as $n \to \infty$.

As $\mathbb{E}\|\mathbf{x}^{(i)}-\zeta\|$ is not known explicitly, in practice μ is replaced by its empirical version μ_N for a given large-size sample $(\mathbf{x}'^{(k)})_{k=1\ldots N}$. The support points $\mathbf{X}_n^s$ are then given by

$$\mathbf{X}_n^s \in \underset{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}}{\operatorname{Arg\,min}} \left(\frac{2}{nN}\sum_{i=1}^{n}\sum_{k=1}^{N}\|\mathbf{x}^{(i)}-\mathbf{x}'^{(k)}\| - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\| \right). \tag{12}$$

The function to be minimized can be written as a difference of functions convex in $\mathbf{X}_n$, which yields a difference-of-convex program. In [29], a majorization-minimization procedure, efficiently combined with resampling, is applied to the construction of large designs (up to $n = 10^4$) in high dimensional spaces (up to $d = 500$). The examples treated clearly show that support points are distributed in a way that matches μ more closely than Monte-Carlo and quasi-Monte Carlo samples [15].

The method can be used to split a dataset into a training set and a test set [23]: the N points $\mathbf{X}_N$ in (12) are those from the dataset, $\mathbf{X}_n^s$ gives the test set and the other

$N - n$ points are used for training. There is a serious additional difficulty though, as choosing $\mathbf{X}_n^s$ among the dataset corresponds to a difficult combinatorial optimization problem. A possible solution is to perform the optimization in a continuous domain $\mathcal{X}$ and then choose $\mathbf{X}_n^s$ that corresponds to the closest points in $\mathbf{X}_N$ (for the Euclidean distance) to the continuous solution obtained [23].

The direct determination of support points through (12) does not allow the construction of a nested sequence of test sets. One possibility would be to solve (12) sequentially, one point at a time, in a continuous domain, and then select the closest point within $\mathbf{X}_N$ as the one to be included in the test set. We shall use a different approach here, based on the greedy minimization of the MMD (11) for the candidate set $\mathcal{S} = \mathbf{X}_N$: at iteration i, the algorithm chooses

$$\mathbf{x}_{i+1}^s \in \operatorname*{Arg\,min}_{\mathbf{x}\in\mathcal{S}} \left(\frac{1}{N} \sum_{k=1}^{N} \|\mathbf{x} - \mathbf{x}'^{(k)}\| - \frac{1}{i+1} \sum_{j=1}^{i} \|\mathbf{x} - \mathbf{x}^{(j)}\| \right). \tag{13}$$

The method requires the computation of the $N(N-1)/2$ distances $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|$, $i, j = 1, \ldots, N$, $i \neq j$, which hinders its applicability to large-scale problems (a test-case with $N = 1\,000$ is presented in Section 5). Note that we consider support points in the input space $\mathcal{X}$ only, with $\mathcal{X} \subseteq \mathbb{R}^d$, in contrast with [23] which considers couples $(\mathbf{x}^{(i)}, y(\mathbf{x}^{(i)}))$ in $\mathbb{R}^{d+1}$ to split a given dataset into a training set and a test set.

Greedy MMD minimization can be applied to other kernels than the distance kernel (10), see [54, 37]. In the next section we consider the closely related method of Kernel Herding (KH) [6], which corresponds to a conditional-gradient descent in the space of probability measures supported on a candidate set $\mathcal{S}$; see, e.g., [40] and the references therein.

3.3 Kernel herding

Let K be a positive definite kernel on $\mathcal{X} \times \mathcal{X}$. At iteration i of kernel herding, with $\xi_i = (1/i) \sum_{j=1}^{i} \delta_{\mathbf{x}^{(i)}}$ the empirical measure for $\mathbf{X}_i$, the next point $\mathbf{x}_{i+1}$ minimizes the directional derivative $F_K(\xi_i, \mu, \delta_{\mathbf{x}})$ of the squared MMD $d_K^2(\xi, \mu)$ at $\xi = \xi_i$ in the direction of the delta measure $\delta_{\mathbf{x}}$, see Appendix A. Direct calculation gives $F_K(\xi_i, \mu, \delta_{\mathbf{x}}) = 2\,[P_{K,\xi}(\mathbf{x}) - P_{K,\mu}(\mathbf{x})] - 2 \int_{\mathcal{X}} [P_{K,\xi}(\mathbf{x}) - P_{K,\mu}(\mathbf{x})]\, \mathrm{d}\xi(\mathbf{x})$, with $P_{K,\xi}(\mathbf{x})$ (resp. $P_{K,\mu}(\mathbf{x})$) the potential of ξ (resp. μ) at $\mathbf{x}$, see (17), and thus

$$\mathbf{x}_{i+1} \in \operatorname*{Arg\,min}_{\mathbf{x}\in\mathcal{S}} \left[P_{K,\xi_i}(\mathbf{x}) - P_{K,\mu}(\mathbf{x}) \right], \tag{14}$$

with $\mathcal{S} \subseteq \mathcal{X}$ a given candidate set. Here, $P_{K,\xi_i}(\mathbf{x}) = (1/i) \sum_{j=1}^{i} K(\mathbf{x}, \mathbf{x}^{(j)})$. When an empirical measure μ_N based on a sample $(\mathbf{x}'^{(k)})_{k=1\ldots N}$ is substituted for μ, we get $P_{K,\mu_N}(\mathbf{x}) = (1/N) \sum_{k=1}^{N} K(\mathbf{x}, \mathbf{x}'^{(k)})$, which gives

$$\mathbf{x}_{i+1} \in \operatorname*{Arg\,min}_{\mathbf{x}\in\mathcal{S}} \left[\frac{1}{i} \sum_{j=1}^{i} K(\mathbf{x}, \mathbf{x}^{(j)}) - \frac{1}{N} \sum_{k=1}^{N} K(\mathbf{x}, \mathbf{x}'^{(k)}) \right] .$$

When K is the energy-distance kernel (10) we thus obtain (13) with a factor $1/i$ instead of $1/(i+1)$ in the second sum.

The candidate set $\mathcal{S}$ in (14) is arbitrary and can be chosen as in Section 3.1. A neat advantage of kernel herding over support points in that the potential $P_{K,\mu}(\mathbf{x})$ is sometimes explicitly available. When $\mathcal{S} = \mathbf{X}_N$, this avoids the need to calculate the $N(N-1)/2$ distances $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|$ and thus allows application to very large sample sizes. This is the case in particular when $\mathcal{X}$ is the cross product of one-dimensional sets $\mathcal{X}_{[i]}$, $\mathcal{X} = \mathcal{X}_{[1]} \times \cdots \times \mathcal{X}_{[d]}$, μ is the product of its marginals $\mu_{[i]}$ on the $\mathcal{X}_{[i]}$, K is the product of one-dimensional kernels $K_{[i]}$, and the one-dimensional integral in $P_{K_{[i]},\mu_{[i]}}(x)$ is known explicitly for each $i \in \{1, \ldots, d\}$. Indeed, for $\mathbf{x} = (x_1, \ldots, x_d) \in \mathcal{X}$, we then have $P_{K,\mu}(\mathbf{x}) = \prod_{i=1}^{d} P_{K_{[i]},\mu_{[i]}}(x_i)$; see [40]. When K is the product of Matérn kernels with regularity parameter $5/2$ and correlation lengths θ_i, $K(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^{d} K_{5/2,\theta_i}(x_i - x'_i)$, with

$$K_{5/2,\theta}(x - x') = \left(1 + \frac{\sqrt{5}}{\theta}|x - x'| + \frac{5}{3\theta^2}(x - x')^2\right) \exp\left(-\frac{\sqrt{5}}{\theta}|x - x'|\right), \qquad (15)$$

the one-dimensional potentials are given in Appendix B for $\mu_{[i]}$ uniform on $[0, 1]$ or $\mu_{[i]}$ the standard normal $\mathcal{N}(0, 1)$. When no observation is available, which is the common situation at the design stage, the correlation lengths have to be set to heuristic values. We empirically found the values of the correlation lengths to have a large influence over the design. A reasonable choice for $\mathcal{X} = [0, 1]^d$ is $\theta_i = n^{-1/d}$ for all i, with n the target number of design points; see [40].

3.4 Numerical illustration

We apply FSSF-fr (denoted FSSF in the following), support points and kernel herding algorithms to the situation where a given initial design of size m has to be completed by a series of additional points $\mathbf{x}^{(m+1)}, \ldots, \mathbf{x}^{(m+n)}$. The objective is to obtain a full design $\mathbf{X}_{m+n}$ that is a good quantization of a given distribution μ.

Figures 1 and 2 correspond to μ uniform on $[0, 1]^2$ and μ the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_2)$, with $\mathbf{I}_2$ the 2-dimensional identity matrix, respectively. All methods are applied to the same candidate set $\mathcal{S}$.

The initial designs $\mathbf{X}_m$ are chosen in the class of space-filling designs, well suited to initialize sequential learning strategies [44]. When μ is uniform, the initial design is a maximin Latin hypercube design [33] with $m = 10$ and the candidate set is given by the $N = 2^{12}$ first points $\mathbf{S}_N$ of a Sobol sequence in $[0, 1]$. When μ is normal, the inverse probability transform method is first applied to $\mathbf{S}_N$ and $\mathbf{X}_m$ (this does not raise any difficulty here as μ is the product of its marginals). The candidate points

$\mathcal{S}$ are marked in gray on Figures 1 and 2 and the initial design is indicated by the red crosses. The index i of each added test point $\mathbf{x}^{(m+i)}$ is indicated (the font size decreases with i). In such a small dimension ($d = 2$), a visual appreciation gives the impression that the three methods have comparable performance. We can notice, however, that FSSF tends to choose points closer to the boundary of $\mathcal{S}$ than the other two, and that support points seem to sample more freely the holes of $\mathbf{X}_m$ than kernel herding, which seems to be closer to a space-filling continuation of the training set. We will come back to these designs when analysing the quality of the resulting predictivity metric estimators in the next section.

4 Numerical results I: construction of a training set and a test set

This section presents numerical results obtained on three different test-cases, in dimension 2 (test-cases 1 and 2) and 8 (test-case 3), for which $y(\mathbf{x}) = f(\mathbf{x})$ with $f(\mathbf{x})$ having an easy to evaluate analytical expression, see Section 4.1. This allows a good estimation of $Q^2_{\text{ideal}}(\mu)$ by $Q^2_{MC} = Q^2_{\text{ideal}}(\mu_M)$, see (1), where μ_M is the empirical measure for a large Monte-Carlo sample ($M = 10^6$), that will serve as reference when assessing the performance of each of the other estimators. We consider the validation designs built by FSSF, support points and kernel herding, presented in Sections 3.1, 3.2, and 3.3, respectively, and, for each one, we compare the performances obtained for both the uniform and the weighted estimator of Section 2.2.

4.1 Test-cases

The training design $\mathbf{X}_m$ and the set $\mathcal{S}$ of potential test set points are as in Section 3.4. For test-cases 1 and 3, μ is the uniform measure on $\mathcal{X} = [0,1]^d$, with $d = 2$ and $d = 8$, respectively; $\mathbf{X}_m$ is a maximin Latin hypercube design in $\mathcal{X}$, and $\mathcal{S}$ corresponds to the first N points $\mathbf{S}_N$ of Sobol' sequence in $\mathcal{X}$, complemented by the 2^d vertices. In the second test-case, $d = 2$, μ is the normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_2)$, and the sets $\mathbf{X}_m$ and $\mathbf{S}_N$ must be transformed as explained in section 3.1. There are $N = 2^{14}$ candidate points for test-cases 1 and 2 and $N = 2^{15}$ for test-case 3 (this value is rather moderate for a problem in dimension 8, but using a larger N yields numerical difficulties for support points; see Section 3.2).

For each test-case, a GP regression model is fitted to the m observations using ordinary Kriging [43] (a GP model with constant mean), with an anisotropic Matérn kernel with regularity parameter $5/2$: we substitute $[(\mathbf{x}-\mathbf{x}')^\top \mathbf{D}(\mathbf{x}-\mathbf{x}')]^{1/2}$ for $|x-x'|$ in (15), with $\mathbf{D}$ a diagonal matrix with diagonal elements $1/\theta_i^2$, and the correlation lengths θ_i are estimated by maximum likelihood via a truncated Newton algorithm. All calculations were done using the Python package OpenTURNS for uncertainty quantification [1]. The kernel used for kernel herding is different and corresponds to the tensor product of one-dimensional Matérn kernels (15), so that the potentials

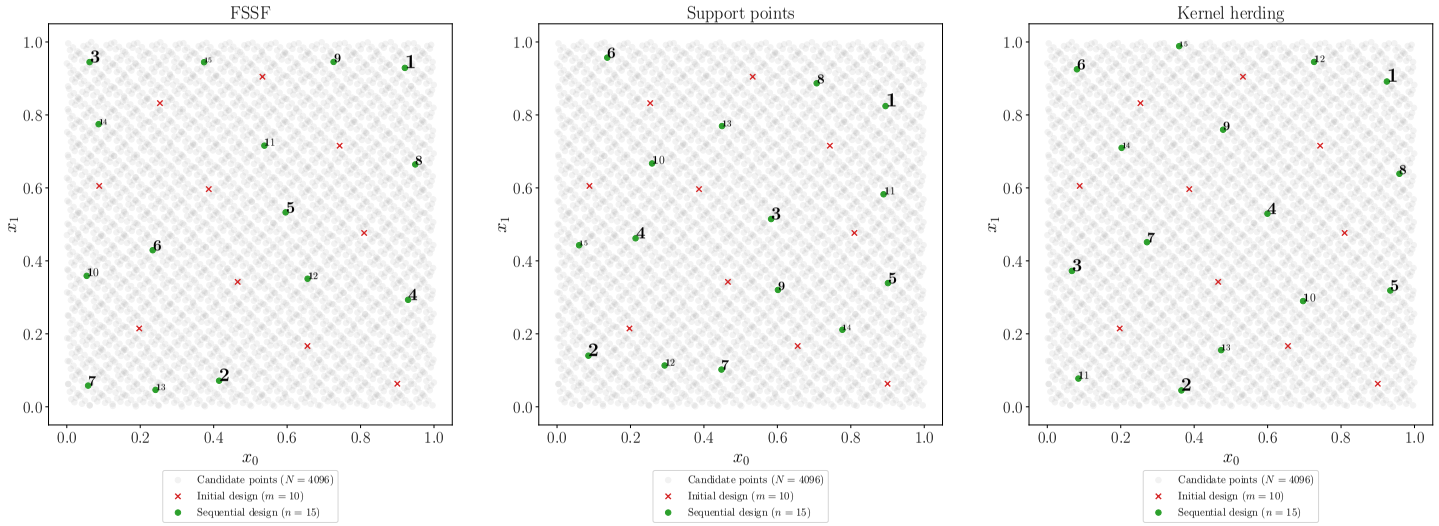

Fig. 1 Additional points (ordered, green) complementing an initial design (red crosses), μ is uniform on $[0, 1]$, the candidate points are in gray.

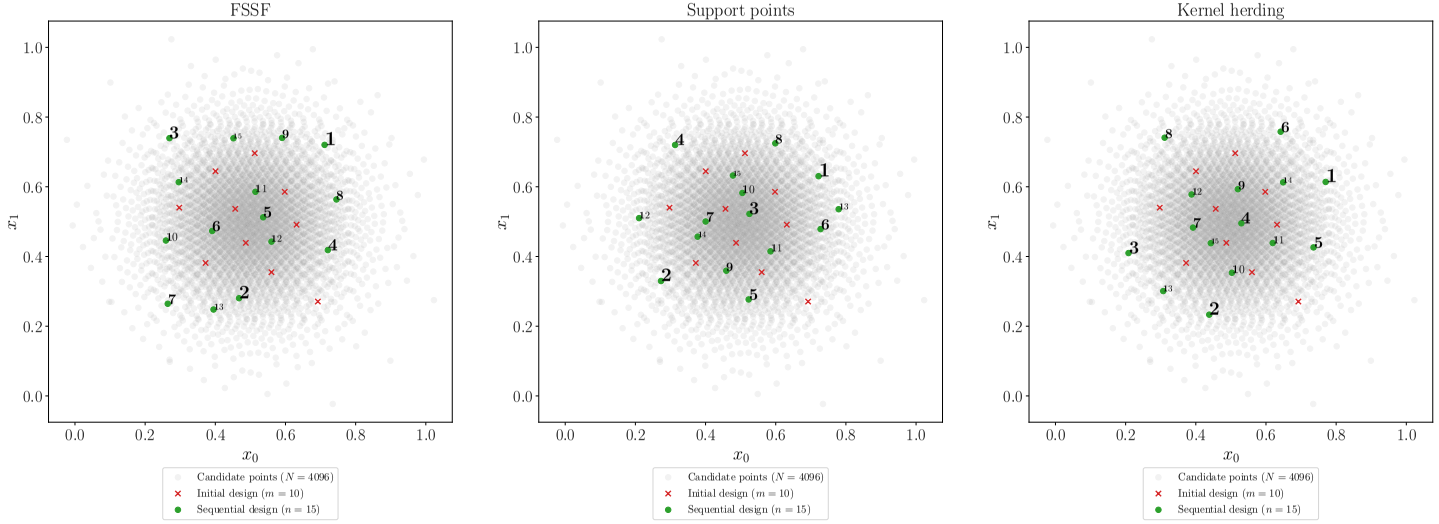

Fig. 2 Additional points (ordered, green) complementing an initial design (red crosses), μ normal, the candidate points are in gray.

$P_{K,\mu}(\cdot)$ are known explicitly (see Appendix B); the correlations lengths are set to $\theta = 0.2$ in test-cases 1 and 3 ($d = 2$) and to $\theta = 0.7$ in test-case 3 ($d = 8$).

Assuming that a model is classified, in terms of the estimated value of its predictivity index Q^2 as "poor fitting" if $Q^2 \in [0.6, 0.8]$, "reasonably good fitting", when $Q^2 \in (0.8, 0.9]$, and "very good fitting" if $Q^2 > 0.9$, we selected, for each test-case three different sizes m of the training set such that the corresponding models cover all three possible situations. For all test-cases, the impact of the size n of the test set is studied in the range $n \in \{4, \ldots, 50\}$.

Test-case 1.

This test function is $f_1(\mathbf{x}) = h(2x_1 - 1, 2x_2 - 1)$, $(x_1, x_2) \in \mathcal{X} = [0, 1]^2$, with

$$h(u_1, u_2) = \frac{\exp(u_1)}{5} - \frac{u_2}{5} + \frac{u_2^6}{3} + 4u_2^4 - 4u_2^2 + \frac{7u_1^2}{10} + u_1^4 + \frac{3}{4u_1^2 + 4u_2^2 + 1}.$$

Color coded 3d and contour plots of f_1 for $\mathbf{X} \in \mathcal{X}$ are shown on the left panel of Figure 3, showing that the function is rather smooth, even if its behaviour along the boundaries of $\mathcal{X}$, in particular close to the vertices, may present difficulties for some regression methods. The size of the training set for this function are: $m \in \{5, 15, 30\}$.

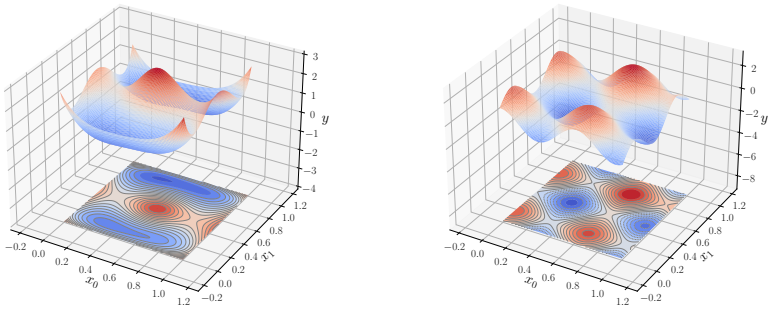

Fig. 3 Left: $f_1(\mathbf{x})$ (test-case 1); right: $f_2(\mathbf{x})$ (test-case 2); $\mathbf{x} \in \mathcal{X} = [0, 1]^2$.

Test-case 2.

The second test function, plotted in the right panel of Figure 3 for $\mathbf{x} \in [0, 1]^2$, is

$$f_2(\mathbf{x}) = \cos\left(5 + \frac{3}{2}x_1\right) + \sin\left(5 + \frac{3}{2}x_1\right) + \frac{1}{100}\left(5 + \frac{3}{2}x_1\right)\left(5 + \frac{3}{2}x_2\right).$$

Training set sizes for this test-case are $m \in \{8, 15, 30\}$.

Test-case 3.

The third function is the so-called "gSobol" function, defined over $\mathcal{X} = [0,1]^8$ by

$$f_3(\mathbf{x}) = \prod_{i=1}^{8} \frac{|4x_i - 2| + a_i}{1 + a_i}, \qquad a_i = i^2 \,.$$

This parametric function is very versatile as both the dimension of its input space and the coefficients a_i can be freely chosen. The sensitivity to input variables is determined by the a_i: the larger a_i is, the less f is sensitive to x_i. Larger training sets are considered for this test-case: $m \in \{15, 30, 100\}$.

4.2 Results and analysis

The numerical results obtained in this section are presented in Figures 4, 5, and 6. Each figure corresponds to one of the test-cases and gathers three sub-figures, corresponding to test sets with sizes m yielding poor (left), reasonably good (centre) or very good (right) fittings.

The baseline value of Q^2_{MC}, calculated with 10^6 Monte-Carlo points, is indicated by the black diamonds (the black horizontal lines). We assume that the error of Q^2_{MC} is much smaller than the errors of all other estimators, and compare the distinct methods through their ability to approximate Q^2_{MC}. For each sequence of nested test-sets ($n \in \{4, \dots, 50\}$), the observed values of $\widehat{Q}^2_n$ (equation (2)) and Q^2_{n*} (equation (7)), are plotted as the solid and dashed lines, respectively.

The figures also show the value Q^2_{LOO} obtained by Leave-One-Out (LOO) cross validation, which is indicated at the left of each figure by a red diamond (values smaller than 0.25 are not shown). Note that, contrarily to the other methods considered, for LOO the test set is not disjoint from the training set, and thus the method does not satisfy the conditions set in the Introduction. As we repeat the complete model-fitting procedure for each training sample of size $m-1$, including the maximum-likelihood estimation of the correlation lengths of the Matérn kernel, the closed-form expressions of [13] cannot be used, making the computations rather intensive. As the three figures show, and as we should expect, Q^2_{LOO} tends to underestimate Q^2_{ideal}: by construction of the training set, LOO cross validation relies on model predictions at points $\mathbf{x}^{(i)}$ far from the other $m-1$ design points used to build the model, and thus tends to systematically overestimate the prediction error at $\mathbf{x}^{(i)}$. The underestimation of Q^2_{ideal} can be particularly severe when m is small, the training set being then necessarily sparse; see Figure 4 where $Q^2_{LOO} < 0.3$ for $m = 5$ and 15.

Let us first concentrate on the non-weighted estimators (solid curves). We can see that the two MMD-based constructions, support points (in orange) and kernel herding (in blue), generally produce better validation designs than FSSF (green curves), leading to values of $\widehat{Q}^2_n$ that approach Q^2_{ideal} quicker as n increases. This is

particularly noticeable for "good" and "very good" models (central and rightmost panels of all three figures). This supports the idea that test sets should complement the training set $\mathbf{X}_m$ by populating the holes it leaves in $\mathcal{X}$ while at the same time be able to mimic the target distribution μ, this second objective being more difficult to achieve for FSSF than for the MMD-based constructions.

Comparison of the two MMD based estimators reveals that support points tend to under-estimate ISE, leading to an over-confident assessment of the model predictivity, while kernel herding displays the expected behaviour, with a negative bias that decreases with n. The reason for the positive bias of estimates based on support points designs is not fully understood, but may be linked to the fact that support points tend to place validation points at "mid-range" from the designs (and not at the furthest points like FSSF or kernel herding), see central and rightmost panels in Figure 1, and thus residuals at these points are themselves already better representatives of the local average errors.

We consider now the impact of the GP-based weighting of the residuals when estimating Q^2 (by Q^2_{n*}), which is related to the relative training-set/validation-set geometry (the manner in which the two designs are entangled in ambient space). The improvement resulting of applying residual weighting is apparent on all panels of the three figures, the dashed curves lying closer to Q^2_{ideal} than their solid counterparts; see in particular kernel herding (blue curve) in Figure 4 and FSSF (green curve) in Figure 5. Unexpectedly, the estimators based on support points seem to be rather insensitive to residual weighting, the dashed and solid orange curves being most of the time close to each other (and in any case, much closer that the green and blue ones). While the reason for this behavior deserves a deeper study, the fact that the support point designs – see Figure 1 – sample in a better manner the range of possible training-to-validation distances, being in some sense less space-filling than both FSSF and kernel herding, is again a plausible explanation for this weaker sensitivity to residual weighting.

Consider now comparison of the behaviour across test-cases. Setting aside the strikingly singular situation of test-case 2, for which kernel herding displays a pathological (bad) behaviour for the "very good" model, and all methods present an overall astonishing good behaviour, we can conclude that the details of the tested function do not seem to play an important role concerning the relative merits of the estimators and validation designs.

We finally observe how the methods behave for models of distinct quality (m leading to poor, good or very good models), comparing the three panels in each figure. On the left panels, m is too small for the model η_m to be accurate, and all methods and test-set sizes are able to detect this. For models of practical interest (good and very good), the test sets generated with support points and kernel herding allow a reasonably accurate estimation of Q^2 with a few points. Note, incidentally, that except for test-case 2 (where the interplay with a non-uniform measure μ complicates the analysis), it is in general easier to estimate the quality of the very good model (right-most panel) than that of the good model (central panel), indicating that the expected complexity (the entropy) of the residual process should be a key factor

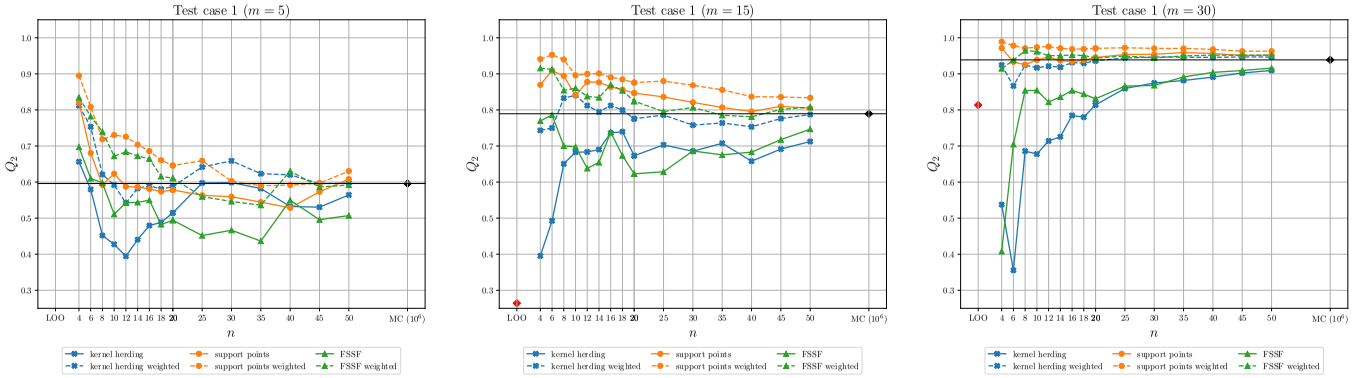

Fig. 4 Test-case 1: predictivity assessment of a poor (left), good (center) and very good (right) model with kernel herding, support points and FSSF test sets.

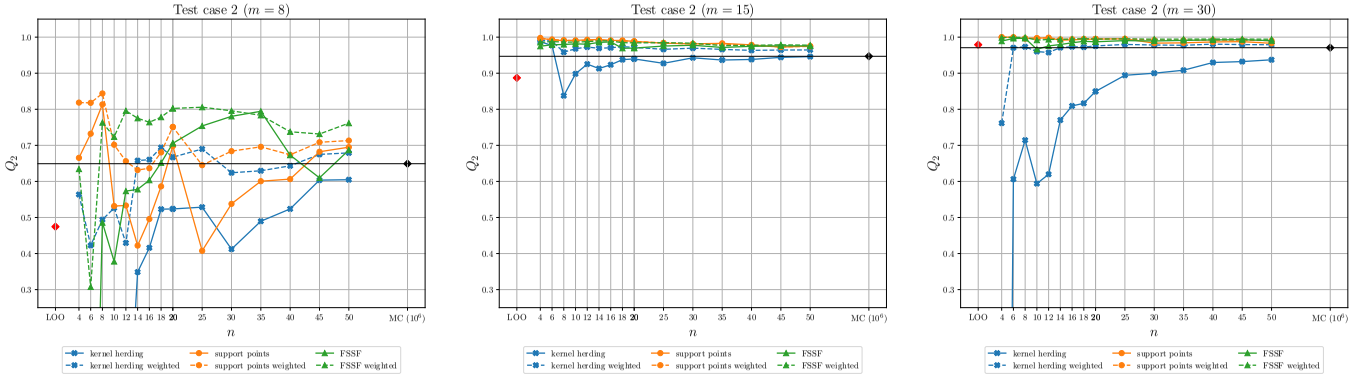

Fig. 5 Test-case 2: predictivity assessment of a poor (left), good (center) and very good (right) model with kernel herding, support points and FSSF test sets.

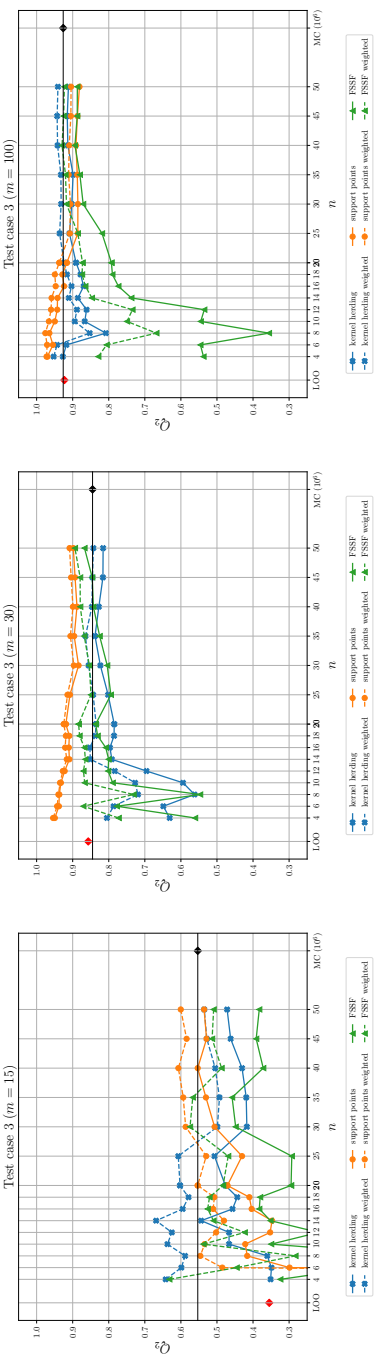

Fig. 6 Test-case 3: predictivity assessment of a poor (left), good (center) and very good (right) model with kernel herding, support points and FSSF test sets.

determining how large the validation set must be. In particular, it may be that larger values of m allow for smaller values of n.

5 Numerical results II: splitting a dataset into a training set and a test set

In this section, we illustrate the performance of the different designs and estimators considered in this paper when applied in the context of an industrial application, to split a given dataset of size N into training and test sets, with m and n points respectively, $m+n=N$. In contrast with [23], the observations $y(\mathbf{x}^{(i)})$, $i=1,\dots,N$, are not used in the splitting mechanism, meaning that it can be performed before the observations are collected and that there cannot be any selection bias related to observations (indeed, the use of observation values in a MMD-based splitting criterion may favour the allocation of the most different observations to different sets, training versus validation).

A ML model is fitted to the training data, and the data collected on the test-set are used to assess the predictivity of the model. The influence of the ratio $r_n = n/N = 1 - m/N$ on the quality assessment is investigated. We also consider Random Cross-Validation (RCV), where n points are chosen at random among the N points of the dataset: for each n, there are $\binom{N}{n}$ possible choices, and we randomly select $R = 1\,000$ designs among them. We fit a model to each of the m-point complementary designs ($m = N - n$), which yields an empirical distribution of Q^2 values for each ratio n/N considered.

5.1 Industrial test-case CATHARE

The test-case corresponds to the computer code CATHARE2 (for "Code Avancé de ThermoHydraulique pour les Accidents de Réacteurs à Eau"), which models the thermal-hydraulic behavior inside nuclear pressurized water reactors [16]. The studied scenario simulates a hypothetical large-break loss of primary coolant accident for which the output of interest is the peak cladding temperature [11, 22]. The complexity of this application lies in the large run-time of the computer model (of the order of twenty minutes) and in the high dimension of the input space: the model involves 53 input parameters z_i, corresponding mostly to constants of physical laws, but also coding initial conditions, material properties and geometrical modeling. The z_i were independently sampled according to normal or log-normal distributions (see axes histograms in Figure 7 corresponding to 10 inputs). These characteristics make this test-case challenging in terms of construction of a surrogate model and validation of its predictivity.

We have access to an existing Monte Carlo sample $\mathbf{Z}_N$ of $N = 1\,000$ points in $\mathbb{R}^{53}$, that corresponds to 53 independent random input configurations; see [22] for

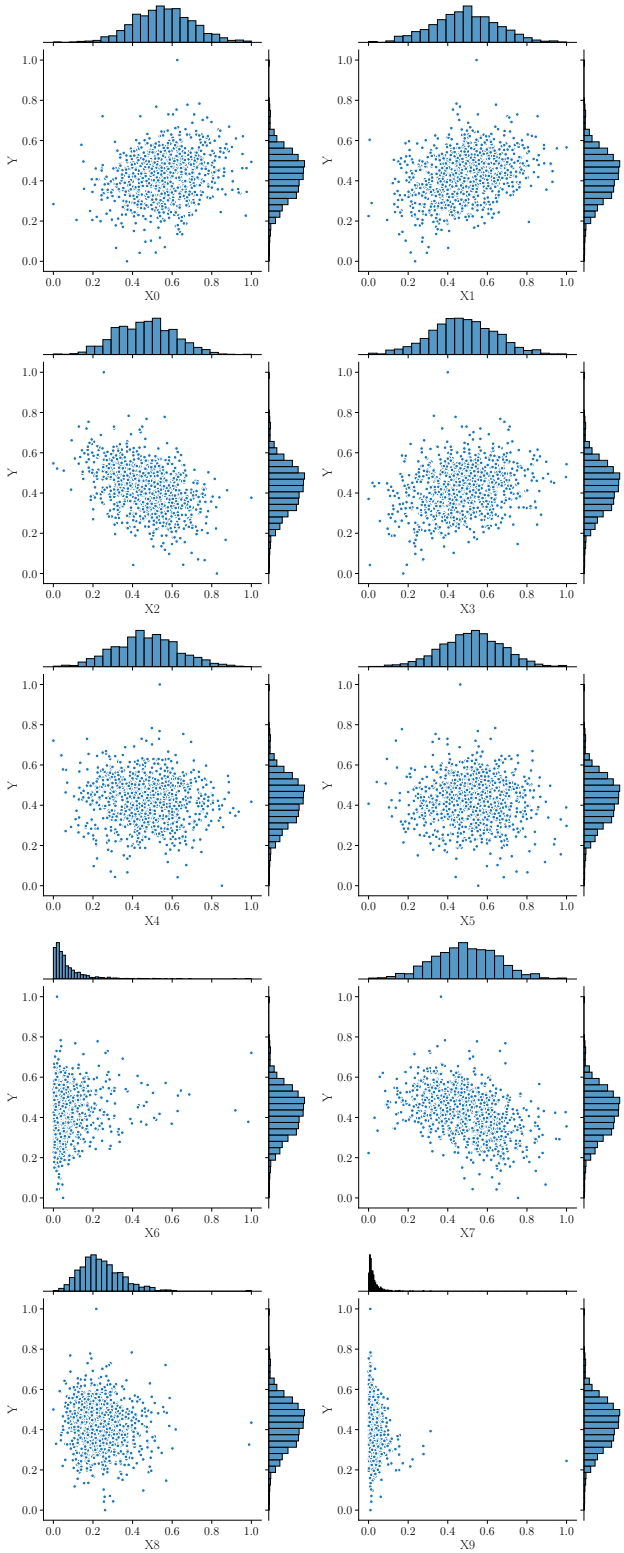

Fig. 7 Test-case CATHARE: inputs output scatter plots ($N = 10^3$)

details. The output of the CATHARE2 code at these N points is also available. To reduce the dimensionality of this dataset, we first performed a sensitivity analysis [10] to eliminate inputs that do not impact the output significantly. This dimension-reduction step relies on the Hilbert-Schmidt Independence Criterion (HSIC), which is known as a powerful tool to perform input screening from a single sample of inputs and output values without reference to any specific ML regression model [18, 9]. HSIC-based statistical tests and their associated p-values are used to identify (with a 5%-threshold) inputs on which the output is significantly dependent (and therefore, also those of little influence). They were successfully applied to similar datasets from thermal-hydraulic applications in [30, 31]. The screened dataset only includes 10 influential inputs, over which the candidate set $\mathbf{X}_N$ used for the construction of the test-set $\mathbf{X}_n$ (and therefore of the complementary training set $\mathbf{X}_{N-n}$) is defined. An input-output scatter plot is presented in Figure 7, showing that indeed the retained factors are correlated with the code output. The marginal distributions are shown as histograms along to the axes of the plots.

To include RCV in the methods to be compared, we need to be able to construct many (here, $R = 1\,000$) different models η_m for each considered design size m. Since Gaussian Process regression proved to be too expensive for this purpose, we settled for the comparatively cheaper Partial Least Squares (PLS) method [55], which retains acceptable accuracy. For each given training set, the model obtained is a sum of monomials in the 10 input variables. Note that models constructed with different training sets may involve different monomials and have different numbers of monomial terms.

5.2 Benchmark results and analysis

Figure 8 compares various ways of extracting an n-point test set from an N-point dataset to estimate model predictivity, for different splitting ratios $n/N \in \{0.1, 0.15, 0.2, \ldots, 0.9\}$.

Consider RCV first. For each value of $r_n = n/N$, the empirical distribution of Q^2_{RCV} obtained from $R = 10^3$ random splittings of $\mathbf{X}_N$ into $\mathbf{X}_m \cup \mathbf{X}_n$ is summarized by a boxplot. Depending on r_n, we can roughly distinguish three behaviors. For $0.1 \leq r_n \lesssim 0.3$ the distribution is bi-modal, with the lower mode corresponding to unlucky test-set selections leading to poor performance evaluations. When $0.3 \lesssim n/N \lesssim 0.7$, the distribution looks uni-modal, revealing a more stable performance evaluation. Note that this is (partly) in line with the recommendations discussed in section 1. For $r_n \gtrsim 0.7$, the variance of the distribution increases with r_n: many unlucky training sets lead to poor models. Note that the median of the empirical distribution slowly decreases as r_n increases, which is consistent with the intuition that the model predictivity should decrease when the size of the training set decreases.

For completeness, we also show by a red diamond on the left of Figure 8 the value of Q^2_{LOO} computed by LOO cross-validation. In principle, being computed using the entire dataset, this value should establish an upper bound on the quality

of models computed with smaller training sets. This is indeed the case for small training sets (rightmost values in the figure), for which the predictivity estimated by LOO is above the majority of the predictivity indexes calculated. But at the same time, we know that LOO cross-validation tends to overestimate the errors, which explains the higher predictivity estimated by some other methods when $m = N - n$ is large enough.

Compare now the behavior of the two MMD-based algorithms of Section 3, $\widehat{Q}^2_n$ (un-weighted) and Q^2_{n*} (weighted) are plotted using solid and dashed lines, respectively, for both kernel herding (in blue) and support points (in orange). FSSF test-sets are not considered, as the application of an iso-probabilistic transformation imposes knowledge of the input distribution, which is not known for this example. Compare first the unweighted versions of the two MMD-based estimators. For small values of the ratio r_n, $0.1 \lesssim r_n \lesssim 0.45$, the relative behavior of support points and kernel herding coincides with what we observed in the previous section, support points (solid orange line) estimating a better performance than kernel herding (solid blue line), which, moreover, is close to the median of the empirical distribution of Q^2_{RCV}. However, for $r_n \geq 0.5$, the dominance is reversed, support points estimating a worse performance than kernel herding.

As r_n increases up to $r_n \lesssim 0.7$ the solid orange and blue curves crossover, and it is now $\widehat{Q}^2_n$ for kernel herding that approximates the RCV empirical median, while the value obtained with support points underestimates the predictivity index. Also, note that for (irrealistic) very large values of r_n both support points and kernel herding estimate lower Q^2 values, which are smaller than the median of the RCV estimates.

Let us now focus on the effect of residual weighting, i.e., in estimators Q^2_{n*} which use the weights computed by the method of Section 2.2, shown in dashed lines in Figure 8. First, note that while for kernel herding weighting leads, as in the previous section, to higher estimates of the predictivity (compare solid and dashed blue lines), this is not the case for support points (solid and dashed orange curves), which, for small split ratios, produces smaller estimates when weighting is introduced. In the large r_n region, the behavior is consistent with what we saw previously, weighting inducing an increase of the estimated predictivity. It is remarkable – and rather surprising – that Q^2_{n*} for support points (the dashed orange line) does not present the discontinuity of the uncorrected curve.

The sum $\sum_{i=1}^{n} w_i^*$ of the optimal weights of support points and kernel herding (6) is shown in Figure 9 (orange and blue curves, respectively). The slow increase with n/N of the sum of kernel-herding weights (blue line) is consistent with the increase of the volume of the input region around each validation point when the size of the training set decreases. The behavior of the sum of weights is more difficult to interpret for support points (orange line) but is consistent with the behavior of Q^2_{n*} on Figure 8. Note that the energy-distance kernel (10) used for support points cannot be used for the weighting method of Section 2.2 as K_E is not positive definite but only conditionally positive definite. A full understanding of the observed curves would require a deeper analysis of the geometric characteristics of the designs generated by the two MMD methods, in particular of their interleaving with the training designs, which is not compatible with the space constraints of this manuscript.

While a number of unanswered points remain, in particular how deeply the behaviours observed may be affected by the poor predictivity resulting from the chosen PLS modeling methodology, the example presented in this section shows that the construction of test sets via MMD minimization and estimation of the predictivity index using the weighted estimator Q^2_{n*} is promising as an efficient alternative to RCV: at a much lower computational cost, it builds performance estimates based on independent data the model developers may not have access to. Moreover, kernel herding proved, in the examples studied in this manuscript, to be a more reliable option for designing the test set, exhibiting a behavior that is consistent with what is expected, and very good estimation quality when the residuals over the design points are appropriately weighted.

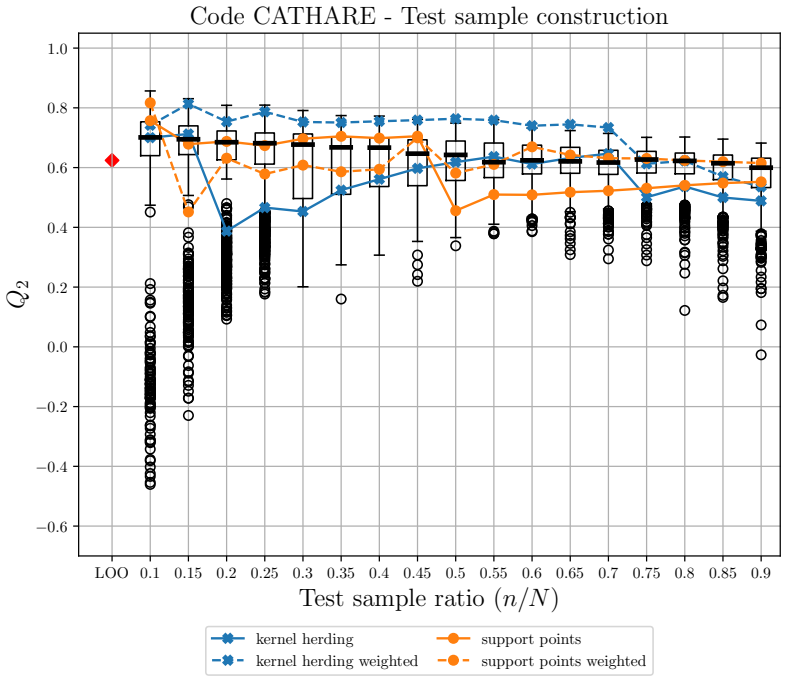

Fig. 8 Test-case CATHARE: estimated Q^2. The box-plots are for random cross-validation, the red diamond (left) is for Q^2_{LOO}.

6 Conclusion

Our study shows that ideas and tools from the design of experiments framework can be transposed to the problem of test-set selection. This paper explored approaches based on support points, kernel herding and FSSF, considering the incremental construction of a test set (i) either as a particular space-filling design problem, where

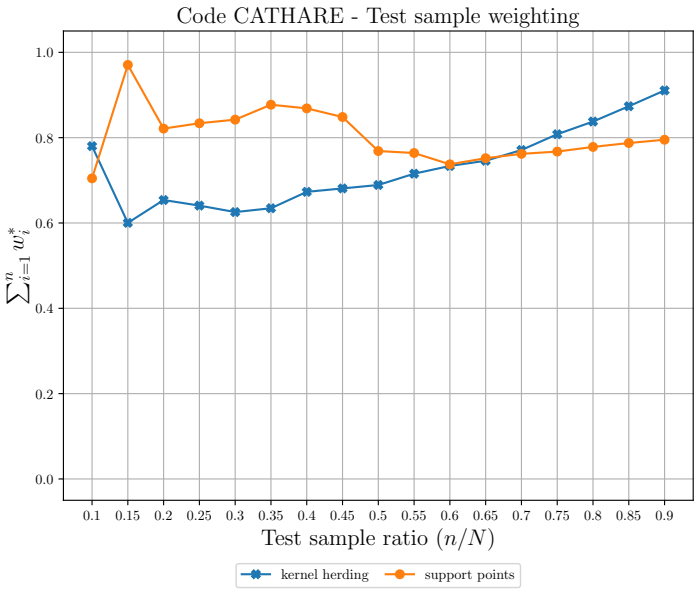

Fig. 9 Test-case CATHARE: sum of the weights (6).

design points should populate the holes left in the design space by the training set, or (*i*) from the point of view of partitioning a given dataset into a training set and a test set.

A numerical benchmark has been performed for a panel of test-cases of different dimensions and complexity. Additionally to the usual predictivity coefficient, a new weighted metric (see [39]) has been proposed and shown to improve assessment of the predictivity of a given model for a given test set.

This weighting procedure appears very efficient for interpolators, like Gaussian process regression models, as it corrects the bias when the points in the test set used to predict the errors are far from the training points. For the first three test-cases (Section 4), pairing one iterative design method with the weight-corrected estimator of the predictivity coefficient Q^2 shows promising results as the estimated Q^2 characteristic is close to the true one even for test-sets of moderate size.

Weighting can also be applied to models that do not interpolate the training data. For the industrial test-case of Section 5, the true Q^2 value is unknown, but the weight-corrected estimation Q^2_{n*} of Q^2 is close to the value estimated by Leave-One-Out cross validation and to the median of the empirical distribution of Q^2 values obtained by random k-fold cross-validation. At the same time, estimation by Q^2_{n*} involves a much smaller computational cost than cross-validation methods, and uses a dataset fully independent from the one used to construct the model.

To each of the design methods considered to select a test set a downside can be attached. FSSF requires knowledge of the input distribution to be able to apply an iso-probabilistic transformation if necessary; it tends to select many points along the

boundary of the candidate set considered. Support points require the computation of the $N(N-1)/2$ distances between all pairs of candidate points, which implies important memory requirements for large N; the energy-distance kernel on which the method relies cannot be used for the weighting procedure. Finally, the efficient implementation of kernel herding relies on analytical expressions for the potentials $P_{K,\mu}$, see Appendices A and B, which are available for particular distributions (like the uniform and the normal) and kernels (like Matérn) only. The great freedom in the choice of the kernel K gives a lot of flexibility, but at the same time implies that some non-trivial decisions have to be made; also, the internal parameters of K, such as its correlation lengths, must to be specified. Future work should go beyond empirical rules of thumb and study the influence of these choices.

We have only computed numerical tests with independent inputs. Kernel herding and support points are both well suited for probability measures not being equal to the product of their marginals, which is a frequent case in real datasets. We have also only considered incremental constructions, as they allow to stop the validation procedure as soon as the estimation of the model predictivity is deemed sufficiently accurate, but it is also possible to select several points at once, using support points [29], or MMD minimization in general [54].

Further developments around this work could be as follows. Firstly, the incremental construction of a test set could be coupled with the definition of an appropriate stopping rule, in order to decide when it is necessary to continue improving the model (possibly by supplementing the initial design with the test set, which seems well suited to this). The MMD $d_{\overline{K}_{|m}}(\zeta_n^*, \mu)$ of Section 2.2 could play an important role in the derivation of such a rule. Secondly, the approach presented gives equal importance to all the d inputs. However, it seems that inputs with a negligible influence on the output should receive less attention when selecting a test set. A preliminary screening step that identifies the important inputs would allow the test-set selection algorithm to be applied on these variables only. For example, when a $\mathbf{X}_N \subset \mathbb{R}^d$ dataset is to be partitioned into $\mathbf{X}_m \cup \mathbf{X}_n$, one could use only $d' < d$ components to define the partition, but still use all d components to build the model and estimate its (weighted) Q^2. Note, however, that this would imply a slight violation of the conditions mentioned in introduction, as it renders the test set dependent on the function observations.

Finally, in some cases the probability measure μ is known up to a normalizing constant. The use of a Stein kernel then makes the potential $P_{K,\mu}$ identically zero [5, 4], which would facilitate the application of kernel herding. Also, more complex problems involve functional inputs, like temporal signals or images, or categorical variables; the application of the methods presented to kernels specifically designed for such situations raises challenging issues.

Acknowledgements This work was supported by project INDEX (INcremental Design of EXperiments) ANR-18-CE91-0007 of the French National Research Agency (ANR). The authors are grateful to Guillaume Levillain and Thomas Bittar for their code development during their work at EDF. Thanks also to Sébastien Da Veiga for fruitful discussions.

Appendix

Appendix A: Maximum Mean Discrepancy

Let K be a positive definite kernel on $\mathcal{X} \times \mathcal{X}$, defining a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_K$ of functions on $\mathcal{X}$, with scalar product $\langle f, g \rangle_{\mathcal{H}_K}$ and norm $\|f\|_{\mathcal{H}_K}$; see, e.g., [2]. For any $f \in \mathcal{H}_K$ and any probability measures μ and ξ on $\mathcal{X}$, we have

$$\left| \int_{\mathcal{X}} f(\mathbf{x}) \, \mathrm{d}\xi(\mathbf{x}) - \int_{\mathcal{X}} f(\mathbf{x}) \, \mathrm{d}\mu(\mathbf{x}) \right| = \left| \int_{\mathcal{X}} \langle f, K_{\mathbf{x}} \rangle_{\mathcal{H}_K} \, \mathrm{d}(\xi - \mu)(\mathbf{x}) \right| = \left| \langle f, (P_{K,\xi} - P_{K,\mu}) \rangle_{\mathcal{H}_K} \right| , \tag{16}$$

where we have denoted $K_{\mathbf{x}}(\cdot) = K(\mathbf{x}, \cdot)$ and used the reproducing property $f(\mathbf{x}) = \langle f, K_{\mathbf{x}} \rangle_{\mathcal{H}_K}$ for all $\mathbf{x} \in \mathcal{X}$, and where, for any probability measure ν on $\mathcal{X}$ and $\mathbf{x} \in \mathcal{X}$,

$$P_{K,\nu}(\mathbf{x}) = \int_{\mathcal{X}} K(\mathbf{x}, \mathbf{x}') \, \mathrm{d}\nu(\mathbf{x}') , \tag{17}$$

is the potential of ν at $\mathbf{x}$. $P_{K,\nu} \in \mathcal{H}_K$ and is called kernel embedding of ν in ML. In some cases, the potential can be expressed analytically (see. Appendix 6), otherwise it can be estimated by numerical quadrature (Quasi Monte Carlo). Cauchy-Schwartz inequality applied to (16) gives

$$\left| \int_{\mathcal{X}} f(\mathbf{x}) \, \mathrm{d}\xi(\mathbf{x}) - \int_{\mathcal{X}} f(\mathbf{x}) \, \mathrm{d}\mu(\mathbf{x}) \right| \leq \|f\|_{\mathcal{H}_K} \, \|P_{K,\xi} - P_{K,\mu}\|_{\mathcal{H}_K}$$

and therefore

$$\|P_{K,\xi} - P_{K,\mu}\|_{\mathcal{H}_K} = \sup_{f \in \mathcal{H}_K : \, \|f\|_{\mathcal{H}_K} = 1} \left| \int_{\mathcal{X}} f(\mathbf{x}) \, \mathrm{d}\xi(\mathbf{x}) - \int_{\mathcal{X}} f(\mathbf{x}) \, \mathrm{d}\mu(\mathbf{x}) \right| .$$

The Maximum Mean Discrepancy (MMD) between ξ and μ (for the kernel K and set $\mathcal{X}$) is $d_K(\xi, \mu) = \|P_{K,\xi} - P_{K,\mu}\|_{\mathcal{H}_K}$. Direct calculation gives

$$d_K^2(\xi, \mu) = \|P_{K,\xi} - P_{K,\mu}\|_{\mathcal{H}_K}^2 = \int_{\mathcal{X}^2} K(\mathbf{x}, \mathbf{x}') \, \mathrm{d}(\xi - \mu)(\mathbf{x}) \mathrm{d}(\xi - \mu)(\mathbf{x}') \tag{18}$$

$$= \mathbb{E}_{\zeta, \zeta' \sim \xi} K(\zeta, \zeta') + \mathbb{E}_{\zeta, \zeta' \sim \mu} K(\zeta, \zeta') - 2\mathbb{E}_{\zeta \sim \xi, \zeta' \sim \mu} K(\zeta, \zeta') , \tag{19}$$

where the random variables ζ and ζ' in (19) are independent, see [49]. When K is the energy distance kernel (10), one recovers the expression (11) for the corresponding MMD. One may refer to [51] for an illuminating exposition on MMD, kernel embedding, and conditions on K (the notion of characteristic kernel) that make d_K a metric on the space of probability measures on $\mathcal{X}$. The distance and Matérn kernels considered in this paper are characteristic.

Appendix B: Analytical computation of potentials for Matérn kernels

As for tensor-product kernels, the potential is the product of the one-dimensional potentials, we only consider one-dimensional input spaces.

For μ the uniform distribution on $[0,1]$ and K the Matérn kernel $K_{5/2,\theta}$ with smoothness $\nu = 5/2$ and correlation length θ, see (15), we get

$$P_{K_{5/2,\theta},\mu}(x) = \frac{16\theta}{3\sqrt{5}} - \frac{1}{15\theta}(S_\theta(x) + S_\theta(1-x)),$$

where

$$S_\theta(x) = \exp\left(-\frac{\sqrt{5}}{\theta}x\right)\left(5\sqrt{5}x^2 + 25\theta x + 8\sqrt{5}\theta^2\right).$$

The expressions $P_{K_{\nu,\theta},\mu}(x)$ for $\nu = 1/2$ and $\nu = 3/2$ can be found in [40].

When μ is the standard normal distribution $\mathcal{N}(0,1)$, the potential $P_{K_{5/2,\theta},\mathcal{N}(0,1)}$ is $P_{K_{5/2,\theta},\mathcal{N}(0,1)}(x) = T_\theta(x) + T_\theta(-x)$, where

$$T_\theta(x) = \frac{1}{6}\left(\frac{5}{\theta^2}x^2 + \left(3 - \frac{10}{\theta^2}\right)\frac{\sqrt{5}}{\theta}x + \frac{5}{\theta^2}\left(\frac{5}{\theta^2} - 2\right) + 3\right)$$
$$\times \operatorname{erfc}\left(\frac{\frac{\sqrt{5}}{\theta} - x}{\sqrt{2}}\right)\exp\left(\frac{5}{2\theta^2} - \frac{\sqrt{5}}{\theta}x\right) + \frac{1}{3\sqrt{2\pi}}\frac{\sqrt{5}}{\theta}\left(3 - \frac{5}{\theta^2}\right)\exp\left(-\frac{x^2}{2}\right).$$

References

1. M. Baudin, A. Dutfoy, B. Iooss, and A-L. Popelin. Open TURNS: An industrial software for uncertainty quantification in simulation. In R. Ghanem, D. Higdon, and H. Owhadi, editors, *Springer Handbook on Uncertainty Quantification*, pages 2001–2038. Springer, 2017.
2. A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer, 2004.
3. T. Borovicka, M. Jr. Jirina, P. Kordik, and M. Jirina. Selecting representative data sets. In A. Karahoca, editor, *Advances in data mining, knowledge discovery and applications*, pages 43–70. INTECH, 2012.
4. W.Y. Chen, A. Barp, F.-X. Briol, J. Gorham, M. Girolami, L. Mackey, and C. Oates. Stein point Markov Chain Monte Carlo. *arXiv preprint arXiv:1905.03673*, 2019.
5. W.Y. Chen, L. Mackey, J. Gorham, F.-X. Briol, and C.J. Oates. Stein points. *arXiv preprint arXiv:1803.10161v4, Proc. ICML*, 2018.
6. Y. Chen, M. Welling, and A. Smola. Super-samples from kernel herding. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pages 109–116. AUAI Press, 2010.
7. C. Chevalier, J. Bect, D. Ginsbourger, V. Picheny, Y. Richet, and E. Vazquez. Fast kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56:455–465, 2014.

8. K. Crombecq, E. Laermans, and T. Dhaene. Efficient space-filling and non-collapsing sequential design strategies for simulation-based modelling. *European Journal of Operational Research*, 214:683–696, 2011.
9. S. Da Veiga. Global sensitivity analysis with dependence measures. *Journal of Statistical Computation and Simulation*, 85:1283–1305, 2015.
10. S. Da Veiga, F. Gamboa, B. Iooss, and C. Prieur. *Basics and Trends in Sensitivity Analysis. Theory and Practice in R*. SIAM, 2021.
11. A. de Crécy, P. Bazin, H. Glaeser, T. Skorek, J. Joufcla, P. Probst, K. Fujioka, B.D. Chung, D.Y. Oh, M. Kyncl, R. Pernica, J. Macek, R. Meca, R. Macian, F. D'Auria, A. Petruzzi, L. Batet, M. Perez, and F. Reventos. Uncertainty and sensitivity analysis of the LOFT L2-5 test: Results of the BEMUSE programme. *Nuclear Engineering and Design*, 12:3561–3578, 2008.
12. C. Demay, B. Iooss, L. Le Gratiet, and A. Marrel. Model selection for Gaussian Process regression: an application with highlights on the model variance validation. *Quality and Reliability Engineering International Journal*, DOI:10.1002/qre.2973, 2021.
13. O. Dubrule. Cross validation of kriging in a unique neighborhood. *Journal of the International Association for Mathematical Geology*, 15(6):687–699, 1983.
14. ENIQ. *Qualification of an AI / ML NDT system - Technical basis*. NUGENIA, ENIQ Technical Report, 2019.
15. K-T. Fang, R. Li, and A. Sudjianto. *Design and Modeling for Computer Experiments*. Chapman & Hall/CRC, 2006.
16. G. Geffraye, O. Antoni, M. Farvacque, D. Kadri, G. Lavialle, B. Rameau, and A. Ruby. CATHARE2 V2.5_2: A single version for various applications. *Nuclear Engineering and Design*, 241:4456–4463, 2011.
17. I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. The MIT Press, 2016.
18. A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Proceedings Algorithmic Learning Theory*, pages 63–77. Springer-Verlag, 2005.
19. T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, second edition, 2009.
20. R. Hawkins, C. Paterson, C. Picardi, Y. Jia, R. Calinescu, and I. Habli. *Guidance on the assurance of machine learning in autonomous systems (AMLAS)*. Assuring Autonomy International Programme (AAIP), University of York, 2021.
21. B. Iooss. Sample selection from a given dataset to validate machine learning models. In *Proceedings of 50th Meeting of the Italian Statistical Society (SIS2021)*, pages 88–93, Pisa, Italy, june 2021.
22. B. Iooss, L. Boussouf, V. Feuillard, and A. Marrel. Numerical studies of the metamodel fitting and validation processes. *International Journal of Advances in Systems and Measurements*, 3:11–21, 2010.
23. V.R. Joseph and A. Vakayil. SPlit: An optimal method for data splitting. *Technometrics*, 64(2):166–176, 2022.
24. R.W. Kennard and L.A. Stone. Computer aided design of experiments. *Technometrics*, 11:137–148, 1969.
25. J.P.C. Kleijnen and R.G. Sargent. A methodology for fitting and validating metamodels in simulation. *European Journal of Operational Research*, 120:14–29, 2000.
26. M. Lemaire, A. Chateauneuf, and J-C. Mitteau. *Structural reliability*. Wiley, 2009.
27. W. Li, L. Lu, X. Xie, and M. Yang. A novel extension algorithm for optimized Latin hypercube sampling. *Journal of Statistical Computation and Simulation*, 87:2549–2559, 2017.
28. G. Lorenzo, P. Zanocco, M. Giménez, M.Marquès, B. Iooss, R. Bolado-Lavin, F. Pierro, G. Galassi, F. D'Auria, and L. Burgazzi. Assessment of an isolation condenser of an integral reactor in view of uncertainties in engineering parameters. *Science and Technology of Nuclear Installations*, 2011(827354, DOI: 10.1155/2011/827354), 2011.
29. S. Mak and V.R. Joseph. Support points. *The Annals of Statistics*, 46:2562–2592, 2018.
30. A. Marrel and V. Chabridon. Statistical developments for target and conditional sensitivity analysis: Application on safety studies for nuclear reactor. *Reliability Engineering & System Safety*, 214:107711, 2021.

31. A. Marrel, B. Iooss, and V. Chabridon. The ICSCREAM methodology: Identification of penalizing configurations in computer experiments using screening and metamodel – Applications in thermal-hydraulics. *Nuclear Science and Engineering*, DOI:10.1080/00295639.2021.1980362, 2021.
32. C. Molnar. *Interpretable machine learning*. github, 2019.
33. M.D. Morris and T.J. Mitchell. Exploratory designs for computationnal experiments. *Journal of Statistical Planning and Inference*, 43:381–402, 1995.
34. W. G. Müller. *Collecting Spatial Data*. Springer, 3rd edition, 2007.
35. J. Nash and J. Sutcliffe. River flow forecasting through conceptual models part I–A discussion of principles. *Journal of Hydrology*, 10(3):282–290, 1970.
36. A. Nogales Gómez, L. Pronzato, and M.-J. Rendas. Incremental space-filling design based on coverings and spacings: improving upon low discrepancy sequences. *Journal of Statistical Theory and Practice*, 15(4):77, 2021.
37. L. Pronzato. Performance analysis of greedy algorithms for minimising a maximum mean discrepancy. *Preprint*, 2021, hal-03114891, arXiv:2101.07564.
38. L. Pronzato and W. Müller. Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22:681–701, 2012.
39. L. Pronzato and M.-J. Rendas. Validation design I: construction of validation designs via kernel herding. *Preprint*, 2021, hal-03474805, arXiv:2112.05583.
40. L. Pronzato and A.A. Zhigljavsky. Bayesian quadrature and energy minimization for space-filling design. *SIAM/ASA Journal on Uncertainty Quantification*, 8:959–1011, 2020.
41. P.Z.G. Qian, M. Ai, and C.F.J. Wu. Construction of nested space-filling designs. *Annals of Statistics*, 37:3616–3643, 2009.
42. P.Z.G. Qian and C.F.J. Wu. Sliced space filling designs. *Biometrika*, 96:945–956, 2009.
43. C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
44. T. Santner, B. Williams, and W. Notz. *The Design and Analysis of Computer Experiments*. Springer, 2003.
45. D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263–2291, 2013.
46. B. Shang and D.W. Apley. Fully-sequential space-filling design algorithms for computer experiments. *Journal of Quality Technology*, 53(2):173–196, 2021.
47. R. Sheikholeslami and S. Razavi. Progressive Latin hypercube sampling: An efficient approach for robust sampling-based analysis of environmental models. *Environmental Modelling & Software*, 93:109–126, 2017.
48. R.C. Smith. *Uncertainty quantification*. SIAM, 2014.
49. A. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory*, pages 13–31. Springer, 2007.
50. R.D. Snee. Validation of regression models: Methods and examples. *Technometrics*, 19:415–428, 1977.
51. B.K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G.R. Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11(Apr):1517–1561, 2010.
52. G. J. Székely and M. L. Rizzo. Testing for equal distributions in high dimension. *InterStat*, 5:1–6, 2004.
53. G. J. Székely and M. L. Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143:1249–1272, 2013.
54. O. Teymur, J. Gorham, M. Riabiz, and C.J. Oates. Optimal quantisation of probability measures using maximum mean discrepancy. In *International Conference on Artificial Intelligence and Statistics*, pages 1027–1035, 2021. arXiv preprint arXiv:2010.07064v1.
55. S. Wold, M. Sjöström, and L. Eriksson. PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130, 2001.
56. Y. Xu and R. Goodacre. On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of Analysis and Testing*, 2:249–262, 2018.