Uncertainty quantification in multi-physics model for wind turbine asset management

Elias FEKHARI

Électricité de France R&D

Chatou, France

&

Côte d'Azur University

Nice, France

A thesis submitted in partial fullfilment of the requirements for the degree of

Doctor of Philosophy
(in Computer Science)

publicly defended on March 12, 2024 in front of the following jury:

Pr. Mireille BOSSY,	INRIA, Sophia-Antipolis	Examiner
Dr. Vincent CHABRIDON	EDF R&D, Chatou	Co-advisor
Dr. Sébastien DA VEIGA	ENSAI, Rennes	Examiner
Dr. Bertrand IOOSS	EDF R&D, Chatou	Thesis director
Dr. Anaïs LOVERA	EDF R&D, Saclay	Invite
Dr. Joseph MURÉ	EDF R&D, Chatou	Co-advisor
Pr. Franck SCHOEFS	Nantes Université, Nantes	Reviewer
Pr. Daniel STRAUB	TUM, Munich	Reviewer
Pr. Bruno SUDRET	ETH, Zürich	Examiner

Contents

List of Figures

List of Tables

Introduction

Industrial context and motivation

The current challenge of energy transition involves, among other things, reducing the share of fossil fuels in the global electricity mix. In this context, offshore wind energy offers several advantages (Beauregard et al., 2022). Offshore energy benefits from more consistent winds than onshore, mainly due to the absence of terrain roughness, it also makes possible the installation of larger and more powerful wind turbines. Since the construction of the first offshore wind farm in Vindeby, Denmark, in 1991, the industry has experienced rapid growth, with a total capacity of 56 GW in operation worldwide in 2021. Over time, offshore wind technology has matured, resulting in significant achievements such as securing projects in Europe through "zero-subsidy bids" where the electricity produced is directly sold on the wholesale market (Beauregard et al., 2022).

However, despite the progress of this sector, scaling limitations and numerous scientific challenges emerge. To meet ambitious national and regional development targets, the wind energy industry must address various scaling issues, including port logistics, the demand for critical natural resources, and sustainable end-of-life processes. Furthermore, the field presents several scientific challenges that often involve coupling data with numerical simulations of physical systems and their surrounding environment. The wind energy community is focused on different objectives, including enhancing the design of floating offshore wind turbines, refining wind resource estimation techniques, and optimizing maintenance operations. In general, several decisions are made throughout the lifespan of a wind turbine by its designer, installer, and operator, all while having only partial knowledge of certain physical phenomena. Therefore, modeling and controlling the various sources of uncertainties associated with offshore wind energy proved to be a key success factor in this highly competitive industry.

Overall, the offshore wind industry needs methods for uncertainty management regarding safety margins and industrial asset management (at the component, wind turbine, and overall wind farm levels) (Van Kuik et al., 2016). For wind project developers, the primary focus is on improving the wind potential assessment of candidate sites by combining various sources of information and modeling the multivariate distribution of environmental conditions. In the case of floating wind projects, the goal is to incorporate a probabilistic aspect from the design phase

(e.g., of the floaters) to define safer, more robust, and more cost-effective solutions. For wind farm owners, end-of-life management is another significant concern. An owner of a wind farm at the end of its life has three options: extend the operational life of assets, replace current wind turbines with newer models, or decommission and sell the wind farm. The first two options require evaluating the structural reliability and residual lifespan, with quantitative assessments reviewed by certification bodies and insurers to issue operating permits. To provide rigorous risk assessments, the generic methodology of *uncertainty quantification methodology* is a widely accepted approach in industrial sectors facing these types of issues (de Rocquigny et al., 2008).

Generic methodology for uncertainty quantification

Computer experiment is a discipline that emerged with the advent of informatics. This practice produces numerical models that allow the simulation of complex system behavior based on initial conditions defined by the analyst. Numerical models quickly became essential for the analysis, design, and certification of complex systems in cases where experiments or physical measurements are too costly or even unfeasible. However, such numerical models are mostly deterministic: the reproducible result of a simulation is associated with a fixed input set of parameters. The issue of managing uncertainties associated with these inputs arises when performing analysis with numerical models.

Uncertainty quantification aims at modeling and controlling uncertainties around a numerical model. To do so, a generic methodology has been proposed to quantify and analyze uncertainties between input and output variables of a numerical model (de Rocquigny et al., 2008). An overview of the mathematical tools used in this field is provided by Sullivan (2015). This approach improves the understanding of a system, ultimately contributing to more robust decision-making. Figure 1 illustrates the main step of the generic uncertainty quantification method, which are briefly summarized hereafter:

- **Step A – Problem specification**: This step involves identifying the system under study and constructing a numerical model capable of precisely simulating its behavior. Specifying the problem also involves the definition of a set of parameters inherent to the numerical model. These parameters include both the input variables and the output variables generated by the simulation. In this document, the numerical model is considered a black box, in contrast to approaches that are integrated within the numerical solution schemes for the system's behavioral equations (referred to as intrusive approaches (Le Maître and Knio, 2010)). Generally, these numerical models are first calibrated against measured data and pass a process of validation and verification to reduce modeling errors (Oberkampf and Roy, 2010).

- **Step B – Uncertainty modeling**: The objective of the second step is to identify and model all the sources of uncertainty related to the input variables. Most of the time the uncertainty modeling is done in the probabilistic framework.

- **Step C – Uncertainty propagation**: This step consists in propagating the uncertain inputs through the computer model. Consequently, the output of the numerical model (commonly scalar) also becomes uncertain. The goal is to estimate a quantity of interest, which is a statistic related to the studied random output variable. The uncertainty propagation method may differ depending on the quantity of interest targeted (e.g., central tendency, a quantile, a rare event probability, etc.).

- **Step C' – Inverse analysis**: In this additional step, a sensitivity analysis can be performed to study the role allocated to each uncertain input leading to the uncertain output.

- **Metamodeling**: Considering the high computational cost associated with some simulations, statistical approaches emulate these expensive simulators with a limited number of simulations. Uncertainty quantification can then be carried out using a "surrogate model" (or metamodel) for a reduced computational cost. This optional step of statistical learning is not strictly a part of uncertainty quantification, but it often proves to be essential for enabling its practical implementation.

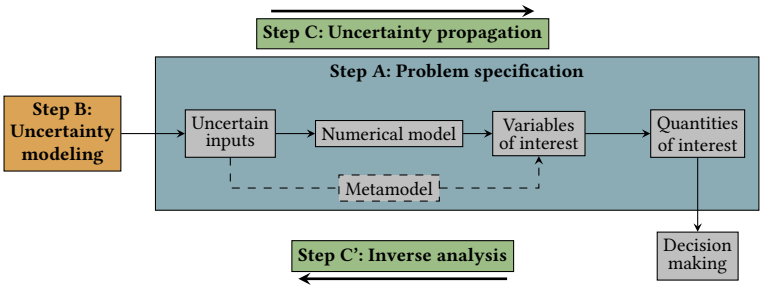

Figure 1 General uncertainty quantification framework (de Rocquigny et al. (2008), adapted by Ajenjo (2023))

Problem statement and outline of the thesis

Risk and uncertainty management in the field of wind energy is a significant concern for the electric utility Électricité de France (EDF). This thesis aims at adapting and applying the generic uncertainty quantification methodology to industrial offshore wind energy studies. As such, this use case raises scientific challenges related to its specific characteristics, described in the following:

- The numerical model exploited in the present work consists of a series of numerical models executed sequentially. This chain is divided into three parts: first, a temporal and stochastic generation of wind and wave velocity fields, followed by the simulation of the coupled hydro-aero-servo-elastic behavior of the wind turbine, and finally a post-processing phase to obtain scalar quantities of interest, aggregated over the temporal outputs.

- The complexity of this simulator, along with the high unit computational cost (about 40 minutes per simulation), requires the use of efficient sampling methods and high-performance computing systems. In addition to the complexity associated with the numerical model, modeling the input uncertainties also represents a challenge. Indeed, the joint distribution associated to environmental conditions present a complex dependence structure. The quality of the inference step is critical as it directly impacts the conclusions of uncertainty propagation.

In order to apply the generic methodology for uncertainty quantification to the offshore wind turbine case, this thesis aims at answering the following questions:

Q1. *How to accureately model the dependence structure associated with the joint environmental distribution?* (➪ Step B)

Q2. *How to perform uncertainty propagation through a computationally expensive numerical chain uniquely based on an empirical description (measured data) of input uncertainties?* (➪ Step C)

Q3. *How to estimate rare event probabilities related to the fatigue failure of offshore wind turbine structures?* (➪ Step C)

Q4. *How to assess and analyse the sensitivity of uncertain inputs regarding quantities of interest resulting from structural reliability (i.e., reliability-oriented sensitivity analysis)?* (➪ Step C')

To propose an answer to these questions, this manuscript is divided into three parts. The first part offers an introduction to uncertainty quantification methods and offshore wind turbine numerical modeling. The second part presents the contributions of this thesis to uncertainty quantification and propagation while the third part describes the contributions to rare event estimation. This manuscript is divided into seven chapters, which are summarized hereafter:

Chapter 1 – Introduction to uncertainty quantification. This chapter gives a brief overview of various topics in uncertainty quantification (Sullivan, 2015). After a reminder of some mathematical concepts, the model specification step is described, considering a black box and its input and output variables. The different types and sources of uncertainties are then presented, along with their modeling within a probabilistic framework. Uncertainty propagation depends on the estimated quantities of interest, therefore, one section addresses propagation methods for central tendency studies, and another focuses on rare event probability estimation (a statistic related to the tails of output distributions). The section dedicated to central tendency presents numerical integration, sampling, and design of experiments methods (Fang et al., 2018). The one about rare event probabilities introduces usual methods from the field of structural reliability (Lemaire et al., 2009; Morio and Balesdent, 2015).

This chapter also covers the main methods for global sensitivity analysis (Da Veiga et al., 2021). This field divides its methods into two major classes: screening methods and importance

measures. Screening techniques, typically applied in high-dimensional problems, aim to identify variables with low impact on the variability of the output of interest. Importance measures, on the other hand, quantitatively allocate, for each input variable, a share of the output variability, enabling the ranking of variables based on their influence.

Finally, this chapter presents an overview of the families of metamodels commonly used in uncertainty quantification (Forrester et al., 2008). Special attention is given to the Gaussian process regression, which involves conditioning a Gaussian process on a set of observations from the numerical model. Once conditioned, the Gaussian process provides richer information than other types of metamodels. This method simultaneously offers a surrogate model (mean of the Gaussian process, also called predictor) and an error function (variance of the process). Some iterative methods (called "active") use this additional information to progressively enrich the metamodel and improve its predictability. These techniques were quite successful in the 1990s for solving optimization problems with expensive functions (Jones et al., 1998). Since then, their use has expanded to solve problems in structural reliability Echard et al. (2011).

Chapter 2 – Introduction to wind turbine modeling and design. The simulation of an offshore wind turbine involves modeling multiple physical aspects interacting with random environmental conditions. This chapter first introduces spectral methods used to generate wind and wave velocity fields by applying inverse Fourier transforms (e.g., as implemented in the TurbSim tool (Jonkman, 2009)]). These simulated wind velocity fields then the become the inputs of a multi-physics wind turbines numerical model. Such simulation includes a simplified modeling of the interactions between fluids and structures (using the blade element momentum theory), dynamic modeling of the structure using flexible multibody methods, and modeling of wind turbine control systems [source]. The numerical code studied generates time series of several physical quantities describing the system's behavior.

This thesis particularly focuses on the probabilistic evaluation of fatigue damage in wind turbine structures. Fatigue damage is a phenomenon that deteriorates the mechanical properties of a material as a result of exposure to many of cyclic, low-amplitude stresses. Currently, standards recommend the use of deterministic safety factors to address this failure mode (IEC-61400-1, 2019; DNV-ST-0437, 2016). A probabilistic approach enhances the analysis and can sometimes reveal conservative of safety margins. Several recent studies have addressed this topic from different methodological perspectives (Huchet, 2019; Lataniotis, 2019; Cousin, 2021; Hirvoas, 2021; Petrovska, 2022).

In this context, this chapter enumerates the input parameters of the calculation chain that are considered uncertain. These random variables are grouped into two groups: the random vector related to the environment (e.g., average wind speed, wind speed standard deviation, wind direction, significant wave height, wave period, and wave direction), and the random vector related to the system (e.g., controller wind misalignment error, soil stiffness, fatigue calculation curve parameters).

Chapter 3 – Kernel-based uncertainty quantification. This chapter examines perturbations in environmental conditions within an offshore wind farm induced by wake effects Larsen et al. (2008). A theoretical offshore wind farm off the southern coast of Brittany is considered as a use case, and a simplified numerical model of wake in this wind farm is used. This model provides an analytical prediction of the wind speed deficit and turbulence created by the wake, taking into account the influence of the floaters' positions due to rigid body dynamics.

In a second phase, uncertainty propagation is carried out thought the wake model, considering the joint distribution of ambient environmental conditions as inputs. In the end, an environmental distribution perturbed by the wake is simulated for each wind turbine. A dissimilarity measure between distribution, based of kernels and named the *maximum mean discrepancy* (MMD), is used to compare the distributions perceived by each wind turbine. This measure allows the clustering of wind turbines exposed to similar environmental conditions, resulting in identical structural responses. Given the high computational cost of aero-servo-hydro-elastic simulations for offshore wind turbines, this preliminary study enables reliability analysis at the wind farm scale without repeating the analysis for each turbine. Ultimately, only four classes are selected to represent a wind farm of 25 turbines.

Chapter 4 – Kernel-based central tendency estimation. Chapter four presents the use of the kernel-based dissimilarity measure (MMD) in the context of probability distribution sampling, a method known as "kernel herding" introduced by Chen et al. (2010). This quadrature technique belongs to the family of "Bayesian quadratures" Briol et al. (2019), which can be viewed as a generalization of quasi-Monte Carlo methods Li et al. (2020).

The properties of this method are highlighted through an industrial application dedicated to estimating the mean fatigue damage of a wind turbine structure. Although this quantity is crucial in the design and certification of wind turbines, the methods used to estimated it are known to be suboptimal (i.e., regular grids). The study is conducted on a model of a fixed offshore wind turbine belonging to a farm in the North Sea. Uncertainties in input environmental conditions are inferred from in-situ measured data.

Finally, a numerical comparison with Monte Carlo and quasi-Monte Carlo sampling reveals the performance and practical advantages of kernel herding. This method allows for direct subsampling from a large environmental database without the need for inference (step B).

Chapter 5 – Kernel-based metamodel validation. This chapter proposes the use of kernel-based sampling methods in the context of model validation for machine learning (or surrogate models). Estimating the predictivity of supervised learning models requires an evaluation of the learned surrogate model on a set of test points that were not used during training. The quality of the validation naturally depends on the properties of the test set and the metric used to summarize the prediction error. This contribution first suggests using space-filling sampling methods to "optimally" select a test set, then, it introduces a new predictivity coefficient that weights the observed errors to improve the global error estimation. A numerical comparison between several sampling methods based on geometric approaches (Shang and Apley, 2020) or

kernel methods Chen et al. (2010); Mak and Joseph (2018) is carried out. Our results show that weighted versions of kernel methods offer superior performance. An application to simulated mechanical loads in an offshore wind turbine model is also presented. This experiment illustrates the practical relevance of this technique as an effective alternative to costly cross-validation techniques.

Chapter 6 – Nonparametric rare event estimation. Estimating rare events probabilities is a common issue in industrial risk management, especially in the field of structural reliability (Chabridon, 2018). To address this, several techniques have been proposed to overcome the known limitations of the Monte Carlo method. Among them, "subset sampling" (Au and Beck, 2001) is a technique based on the split of a rare probability into a product of less rare (and thus easier to estimate) conditional probabilities associated with nested failure events. However, this technique relies on conditional simulation using Markov chain Monte Carlo (MCMC) methods. These algorithms, while converging, often produce samples that are not independent and identically distributed (i.i.d.) due to the correlation between the Markov chains. In this chapter another conditional sampling method is proposed, with the advantage of preserving the i.i.d. property. Independent sampling is particularly relevant for reusing these samples in a posterior reliability-oriented sensitivity analysis. The algorithm introduced is based on the non-parametric inference of the conditional joint distribution using kernel density estimation of marginals combined with dependence inference using the empirical Bernstein copula (Sancetta and Satchell, 2004). The so-called "Bernstein adaptive nonparametric conditional sampling" (BANCS), is compared to the subset sampling method for several structural reliability problems. The initial results are promising, but further investigation is needed to control the estimator's bias.

Chapter 7 – Sequential reliability oriented sensitivity analysis. This chapter deals with sensitivity analysis for risk measures (e.g., rare event probabilities). Global sensitivity analysis (Da Veiga et al., 2021) assigns a portion of the global output variability to each variable (or group of variables), often using a functional decomposition of the output variance. However, when studying risk measure (often located in the distributions' tails), the global sensitivity might be very different to the sensitivity to the risk measure. "Reliability-oriented sensitivity analysis" (ROSA), studies the impact of the inputs in regard to a risk-measure such as a rare event probability (see e.g., Chabridon (2018)). Using the nested subsets obtained with the BANCS algorithm (presented in Chapter 6), the idea of this chapter is to study the ROSA evolution as the subsets get closer to the failure domain. For each subset, a ROSA is carried out with a kernel-based importance measure called the "Hilbert-Schmidt Independence Criterion" adapted to this context (Marrel and Chabridon, 2021).

Numerical developments

Several implementations developed in this thesis are available on different platforms, allowing the reader to reproduce some numerical results in an open-data approach:

`otkerneldesign`[1]	• This Python package generates designs of experiments based on kernel methods such as Kernel Herding and Support Points. A tensorized implementation of the algorithms was proposed, significantly increasing their performances. Additionally, optimal weights for Bayesian quadrature are provided. • This Python package, developed in collaboration with J.Muré, is available on the platform Pypi and fully documented.
`bancs`[2]	• This Python package proposes an implementation of the "Bernstein Adaptive Non-parametric Conditional Sampling" method for rare event estimation. • This Python package is available on the PyPI platform and is illustrated with examples and analytical benchmarks.
`ctbenchmark`[3]	• This Python package presents a standardized process to benchmark different sampling methods for central tendency estimation. • This Python package is available on a GitHub repository with analytical benchmarks.
`copulogram`[4]	• This Python package proposes an implementation of a synthetic visualization tool for multivariate distributions. • This Python package, developed in collaboration with V.Chabridon, is available on the Pypi platform.

[1] Documentation: https://efekhari27.github.io/otkerneldesign/master/
[2] Repository: https://github.com/efekhari27/bancs
[3] Repository: https://github.com/efekhari27/ctbenchmark
[4] Repository: https://github.com/efekhari27/copulogram

Publications and communications

The research contributions in this manuscript are based on the following publications:

Book Chap.	E. Fekhari, B. Iooss, J. Muré, L. Pronzato and M.J. Rendas (2023). "Model predictivity assessment: incremental test-set selection and accuracy evaluation". In: *Studies in Theoretical and Applied Statistics*, pages 315–347. Springer.
Jour. Pap.	E. Fekhari, V. Chabridon, J. Muré and B. Iooss (2023). "Given-data probabilistic fatigue assessment for offshore wind turbines using Bayesian quadrature". In: *Data-Centric Engineering*.
Int. Conf.	E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2022). "Numerical Studies of Bayesian Quadrature Applied to Offshore Wind Turbine Load Estimation". In: *SIAM Conference on Uncertainty Quantification (SIAM UQ22)*, Atlanta, USA. (Talk) E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2022). "Model predictivity assessment: incremental test-set selection and accuracy evaluation". In: *22nd Annual Conference of the European Network for Business and Industrial Statistics (ENBIS 2022)*, Trondheim, Norway. (Talk) E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2022). "Efficient techniques for fast uncertainty propagation in an offshore wind turbine multi-physics simulation tool". In: *Proceedings of the 5th International Conference on Renewable Energies Offshore (RENEW 2022)*, Lisbon, Portugal. (Paper & Talk) E. Fekhari, V. Chabridon, J. Muré and B. Iooss (2023). "Bernstein adaptive nonparametric conditional sampling: a new method for rare event probability estimation"[5]. In: *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP 14)*, Dublin, Ireland. (Paper & Talk) E. Vanem, E. Fekhari, N. Dimitrov, M. Kelly, A. Cousin and M. Guiton (2023). "A joint probability distribution model for multivariate wind and wave conditions". In: *Proceedings of the ASME 2023 42th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2023)*, Melbourne, Australia. (Paper) A. Lovera, E. Fekhari, B. Jézéquel, M. Dupoiron, M. Guiton and E. Ardillon (2023). "Quantifying and clustering the wake-induced perturbations within a wind farm for load analysis". In: *Journal of Physics: Conference Series (WAKE 2023)*, Visby, Sweden (Paper)
Nat. Conf.	E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2022). "Kernel-based quadrature applied to offshore wind turbine damage estimation". In: *Proceedings of the Mascot-Num 2022 Annual Conference (MASCOT NUM 2022)*, Clermont-Ferrand, France (Poster) E. Fekhari, B. Iooss, V. Chabridon, J. Muré (2023). "Rare event estimation using nonparametric Bernstein adaptive sampling". In: *Proceedings of the Mascot-Num 2023 Annual Conference (MASCOT-NUM 2023)*, Le Croisic, France (Talk)
Invited Lec.	Le Printemps de la Recherche 2022, Nantes, France. "Traitement des incertitudes pour la gestion d'actifs éoliens". (Talk) Journées Scientifiques de l'Eolien 2024, Saint-Malo, France. "Evaluation probabiliste de la fiabilité en fatigue des structures éoliennes en mer". (Talk)

[5]This contribution was rewarded by the "CERRA Student Recognition Award"

Part I:

Introduction to uncertainty quantification and wind energy

Toute pensée émet un coup de dé.

S. Mallarmé

Part II:

Contributions to uncertainty quantification and propagation

Le doute est un état mental désagréable,
mais la certitude est ridicule.

Voltaire

Chapter 1

Kernel-based central tendency estimation

This chapter is adapted from the following reference:

E. Fekhari, V. Chabridon, J. Muré and B. Iooss (2023). "Given-data probabilistic fatigue assessment for offshore wind turbines using Bayesian quadrature". In: *Data-Centric Engineering*, In press.

1.1 Introduction

As a sustainable and renewable energy source, offshore wind turbines (OWT) are likely to take a growing share of the global electric mix. However, to be more cost-effective, wind farm projects tend to move further from the coast, exploiting stronger and steadier wind resources. Going further offshore, wind turbines are subject to more severe and uncertain environmental conditions (i.e., wind and waves). In such conditions, their structural integrity should be certified. To do so, numerical simulation and probabilistic tools have to be used. In fact, according to Graf et al. (2016), for new environmental conditions or new turbine models, international standards such as IEC-61400-1 (2019) from the International Electrotechnical Commission and DNV-ST-0437 (2016) from Det Norske Veritas recommend performing over 2×10^5 simulations distributed over a grid. However, numerical simulations are computed by a costly hydro-servo-aero-elastic wind turbine model, making the design process time-consuming. In the following, the simulated output cyclic loads studied are aggregated over the simulation period to assess the mechanical fatigue damage at hot spots of the structure. To compute the risks associated with wind turbines throughout their lifespan, one can follow the steps of the universal framework for the treatment of uncertainties presented in the introduction of this manuscript Fig. 1. After specifying the problem (Step A), one can quantify the uncertainties related to site-specific environmental conditions represented by the random vector $\mathbf{X} \in \mathcal{D}_{\mathbf{X}} \subset \mathbb{R}^d, d \in \mathbb{N}^*$ (Step B). Then, one can propagate them through the OWT simulation model (Step C) denoted by $g : \mathcal{D}_{\mathbf{X}} \to \mathbb{R}, \mathbf{X} \mapsto Y = g(\mathbf{X})$, and estimate a relevant quantity of interest $\psi(Y) = \psi(g(\mathbf{X}))$ (e.g., a mean, a quantile, a failure probability). An accurate estimation of the quantity of interest $\psi(Y)$ relies on both a relevant quantification of the input uncertainty and an efficient sampling method.

Regarding Step B, when dealing with uncertain environmental conditions, a specific difficulty often arises from the complex dependence structure such variables may exhibit. Here, two cases may occur: either measured data are directly available (i.e., the "given-data" context) or a theoretical parametric form for the joint input probability distribution can be postulated. Such existing parametric joint distributions often rely on prior data fitting combined with expert knowledge. For example, several parametric approaches have been proposed in the literature to derive such formulations, ranging from fitting conditional distributions Vanem et al., 2023) to using vine copulas (Li and Zhang, 2020). When a considerable amount of environmental data is available, nonparametric approaches as the empirical Bernstein copula were studied in the Chapter **??** to capture complex dependence structures. Alternatively, an idea is to directly use

the data as an empirical representation of input uncertainties in order to avoid an additional inference error.

Step C usually focuses on propagating the input uncertainties in order to estimate the quantity of interest. Depending on the nature of $\psi(Y)$, one often distinguishes between two types of uncertainty propagation: a central tendency estimation (e.g., focusing on the output mean value or the variance) and a tail estimation (e.g., focusing on a high-order quantile or a failure probability). When uncertainty propagation aims at central tendency estimation, the usual methods can be split into two groups. First, those relying on sampling, i.e., mainly Monte Carlo sampling (Graf et al., 2016), quasi-Monte Carlo sampling (Müller and Cheng, 2018), geometrical subsampling (Kanner et al., 2018), or deterministic quadrature rules (Van den Bos, 2020). All these methods estimate the quantity directly on the numerical simulator's outputs. Second, those that rely on the use of surrogate models (or metamodels, see Fig. 1) to emulate the costly numerical model by a statistical model. Among a large panel of surrogates, one can mention, regarding wind energy applications, the use of polynomial chaos expansions (Dimitrov et al., 2018; Murcia et al., 2018), Gaussian process regression (Huchet, 2019; Teixeira et al., 2019a; Slot et al., 2020; Wilkie and Galasso, 2021), or artificial neural networks (Bai et al., 2023). When uncertainty propagation aims at studying the tail of the output distribution such as in risk or reliability assessment, one usually desires to estimate a quantile or a failure probability. In the wind energy literature, failure probability estimation has been largely studied, e.g., in time-independent reliability assessment (Zwick and Muskulus, 2015; Slot et al., 2020; Wilkie and Galasso, 2021) or regarding time-dependent problems (Abdallah et al., 2019; Lataniotis, 2019).

During the overall process described in Fig. 1, modelers and analysts often need to determine whether inputs are influential or not in order to prioritize their effort (in terms of experimental data collecting, simulation budget, or expert elicitation). Sometimes, they want to get a better understanding of the OWT numerical models' behavior or to enhance the input uncertainty modeling. All these questions are intimately related to the topic of sensitivity analysis (Saltelli et al., 2008; Da Veiga et al., 2021) and can be seen as an "inverse analysis" denoted by Step C' in Fig. 1. In the wind energy literature, one can mention, among others, some works related to Spearman's rank correlation analysis and the use of the Morris method in Velarde et al. (2019); Petrovska (2022). Going to variance-based analysis, the direct calculation of Sobol' indices after fitting a polynomial chaos surrogate model has been proposed in many works (e.g., in Murcia et al., 2018) while the use of distributional indices (e.g., based on the Kullback–Leibler divergence) has been investigated by Teixeira et al. (2019b).

The present chapter focuses on the problem of uncertainty propagation, and more specifically, on the mean fatigue damage estimation (i.e., $\psi(Y) = \mathbb{E}[g(\mathbf{X})]$). Such a problem is usually encountered, by engineers, during the design phase. Most of the time, current standards as well as common engineering practices make them use regular grids (Huchet, 2019). Altogether, one can describe three alternative strategies: (i) direct sampling on the numerical model (e.g., using Monte Carlo), (ii) sampling on a static surrogate model (e.g., using Gaussian process regression), or (iii) using an "active learning" strategy (i.e., progressively adding evaluations of the numerical

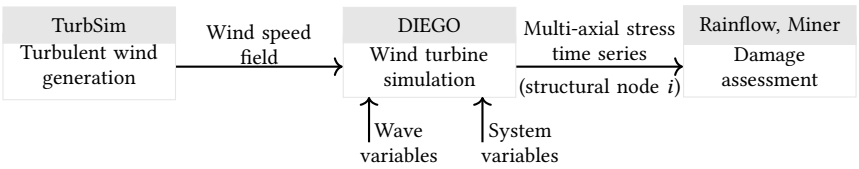

Figure 1.1 Diagram of the chained OWT simulation model.

model to enhance the surrogate model fitting process). In practice, fitting a surrogate model in the context of OWT fatigue damage can be challenging due to the nonlinearity of the code. Moreover, the surrogate model validation procedure complexifies the process. Finally, active learning strategies restrict the potential number of parallel simulations, which limits the use of HPC facilities. Thus, the main contribution of this chapter is to explore different ways to propagate uncertainties by directly evaluating the numerical model (i.e., without any surrogate model) with a relevant tradeoff between computational cost and accuracy. In the specific context of wind turbine fatigue damage, this work shows how to propagate uncertainties arising from a complex input distribution through a costly wind turbine simulator. The proposed work consists of evaluating the advantages and limits of kernel herding as a tool for given-data, fast, and fully-distributable uncertainty propagation in OWT simulators. Additionally, this sampling method is highly flexible, allowing one to complete an existing design of experiments. Such a property can be crucial in practice when the analyst is asked to include some specific points to the design (e.g., characteristic points describing the system's behavior required by experts or by standards, see Huchet, 2019).

The present chapter is organized as follows. Section 1.2 will present the industrial use case related to a wind farm operating in Teesside, UK. Then, Section 1.3 will introduce various kernel-based methods for central tendency estimation. Section 1.4 will analyze the results of numerical experiments obtained on both analytical and industrial cases. Finally, the last section will present some discussions and draw some conclusions.

1.2 Treatment of uncertainties on the Teesside wind farm

An OWT is a complex system interacting with its environment. To simulate the response of this system against a set of environmental solicitations, multi-physics numerical models are developed. In the present chapter, the considered use case consists of a chain of three numerical codes executed sequentially. As illustrated in Fig. 1.1, a simulation over a time period is the sequence of, first, a turbulent wind speed field generation, then a wind turbine simulation (computing various outputs including mechanical stress), and finally, a post-processing phase to assess the fatigue damage of the structure.

1.2.1 Numerical simulation model

This subsection generally describes the modeling hypotheses considered in the industrial use case, further details regarding wind turbines modeling a provided in Chapter **??** of this manuscript. The first block of the chain consists of a turbulent wind field simulator called "TurbSim" (developed by Jonkman, 2009 from the National Renewable Energy Laboratory, USA) that uses, as a turbulence model, a Kaimal spectrum (Kaimal et al., 1972) (as recommended by the IEC-61400-1, 2019). Moreover, to extrapolate the wind speed vertically, the shear is modeled by a power law. Since the wind field generation shows inherent stochasticity, each 10-minute long simulation is repeated with different pseudo-random seeds and one averages the estimated damage over these repetitions. This question has been widely studied by some authors, (e.g., Slot et al., 2020), who concluded that the six repetitions recommended by the IEC-61400-1 (2019) may be insufficient to properly average this stochasticity. Thus, in the following, the simulations are repeated eleven times (picking an odd number also directly provides the median value over the repetitions). This number of repetitions was chosen to suit the maximum number of simulations and the storage capacity of the generated simulations.

As a second block, one finds the "DIEGO" software (for "Dynamique Intégrée des Éoliennes et Génératrices Offshore"[1]) which is developed by EDF R&D (Kim et al., 2022) to simulate the aero-hydro-servo-elastic behavior of OWTs. It takes the turbulent wind speed field generated by TurbSim as input and computes the dynamical behavior of the system (including the multiaxial mechanical stress at different nodes of the structure). For the application of interest here, the control system is modeled by the open-source DTU controller (Hansen and Henriksen, 2013), and no misalignment between the wind and the OWT is assumed. As for the waves, they are modeled in DIEGO using a JONSWAP spectrum (named after the 1975 Joint North Sea Wave Project). The considered use case here consists of a DIEGO model of a Siemens SWT 2.3MW bottom-fixed turbine on a monopile foundation (see the datasheet in Table 1.1), currently operating in Teesside, UK (see the wind farm layout and wind turbine diagram in Fig. 1.2). Although wind farms are subject to the wake effect, affecting the behavior and performance of some turbines in the farm, this phenomenon is not considered in this chapter. To avoid numerical perturbations and reach the stability of the dynamical system, our simulation period is extended to 1000 seconds and the first 400 seconds are cropped in the post-processing step. This chained OWT numerical simulation model has been deployed on an EDF R&D HPC facility to benefit from parallel computing speed up (a single simulation on one CPU takes around 20 minutes).

1.2.2 Measured environmental data

During the lifespan of a wind farm project, environmental data is collected at different phases. In order to decide on the construction of a wind farm, meteorological masts, and wave buoys are usually installed on a potential site for a few years. After its construction, each wind turbine

[1] In English, "Integrated Dynamics of Wind Turbines and Offshore Generators".

Table 1.1 Teesside Offshore Wind turbine datasheet

Siemens SWT-2.3–93	
Rated power	2.3 MW
Rotor diameter	93 m
Hub height	83 m
Cut-in, cut-out wind speed	4 m/s, 25 m/s

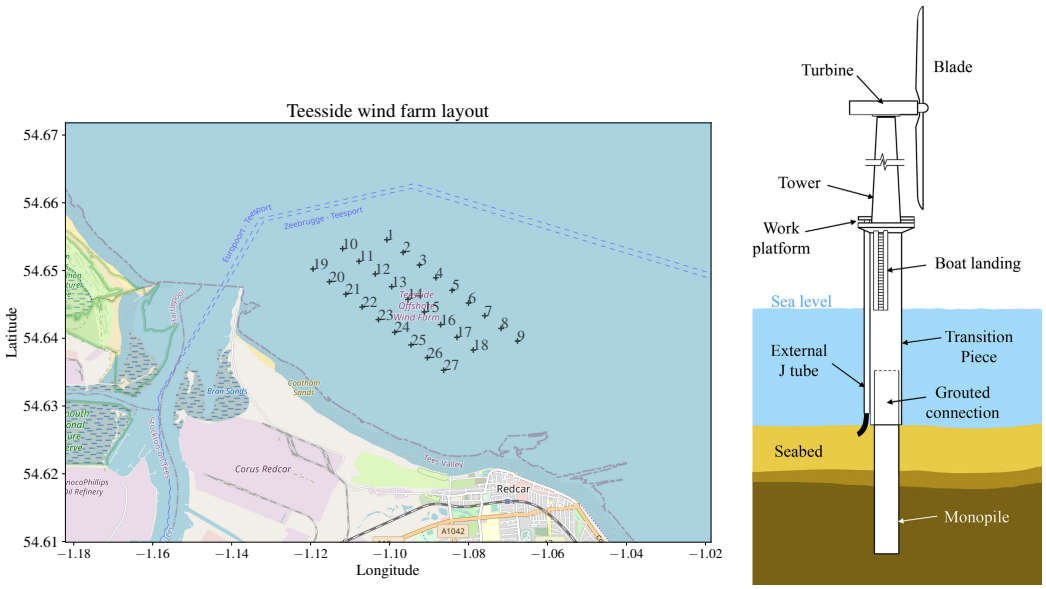

Figure 1.2 Teesside wind farm layout (left). Monopile OWT diagram (Chen et al., 2018) (right)

Variable	Notation	Unit	Description
Mean wind speed	U	$\mathrm{m.s}^{-1}$	10-min. average horizontal wind speed
Wind turbulence	σ_U	$\mathrm{m.s}^{-1}$	10-min. wind speed standard deviation
Wind direction[2]	θ_{wind}	deg.	10-min. average wind direction
Significant wave height	H_s	m	Significant wave height
Peak wave period	T_p	s	Peak 1-hour spectral wave period
Wave direction	θ_{wave}	deg.	10-min. average wave direction

Table 1.2 Description of the environmental data.

is equipped with monitoring instruments (e.g., cup anemometers). In total, five years of wind data have been collected on the turbines which are not affected by the wake on this site. Their acquisition system (usually called "SCADA" for "Supervisory Control And Data Acquisition") has a sampling period of ten minutes. The wave data arise from a buoy placed in the middle of the farm. These data describe the physical features listed in Table 1.2. A limitation of the present study is that it controller-induced uncertainty (like wind misalignment) is not considered.

The Teesside farm is located close to the coast, making the environmental conditions very different depending on the direction (see the wind farm layout in Fig. 1.2). Since measures are also subject to uncertainties, a few checks were made to ensure that the data were physically consistent. Truncation bounds were applied since this study is focused on central tendency estimation (i.e., mean behavior) rather than extreme values. In practice, this truncation only removes extreme data points (associated with storm events). In addition, a simple trigonometric transform is applied to each directional feature to take into account their cyclic structure. Finally, the remaining features are rescaled (i.e., using a min-max normalization).

Teesside's environmental data is illustrated by its copulogram in Fig. 1.3, a graphical tool presented in Section **??** to visualize multivariate data. The copulogram exhibits the marginals with univariate kernel density estimation plots (in the diagonal), and the dependence structure with scatter plots in the normalized rank space (in the upper triangle). Looking at data in the rank space instead of the initial space allows one to observe the ordinal associations between variables. The scatter plots of normalized ranks are actually a representation of the empirical copula density. Two independent variables will present a uniformly distributed scatter plot in the rank space. In the lower triangular matrix, the scatter plots are set in the physical space, merging the effects of the marginals and the dependencies (as in the usual visualization offered by the matrix plot). Since the dependence structure is theoretically modeled by an underlying copula, this plot is called *copulogram*, generalizing the well-known "correlogram" to nonlinear dependencies. It gives a synthetic and empirical decomposition of the dataset that was implemented in a new open-source Python package named `copulogram`[3].

On Fig. 1.3, a large sample $\mathcal{S} \subset \mathcal{D}_{\mathbf{X}}$ (with size $N = 10^4$) is randomly drawn from the entire Teesside data (with size $N_{\mathrm{Teesside}} = 2 \times 10^5$), and plotted in grey. In the same figure, the orange

[2]Note that the two directional variables could be replaced by a wind-wave misalignment variable for a bottom-fixed technology, however, our framework can be directly transposed to floating models.

[3]GitHub repository: https://github.com/efekhari27/copulogram

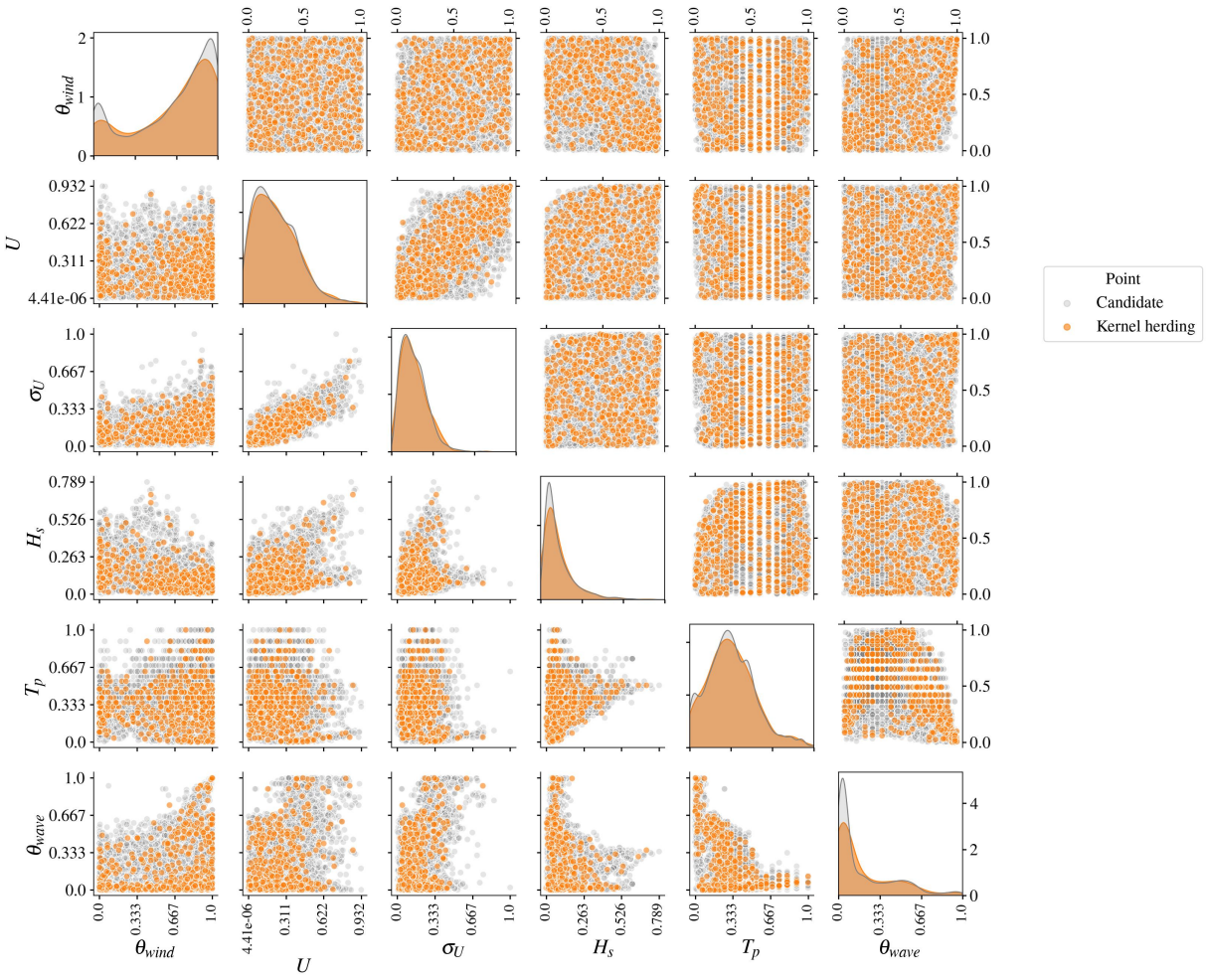

Figure 1.3 Copulogram of the Teesside measured data ($N = 10^4$ in grey), kernel herding subsample ($n = 500$ in orange). Marginals are represented by univariate kernel density estimation plots (diagonal), the dependence structure with scatter plots in the rank space (upper triangle). Scatter plots on the bottom triangle are set in the physical space.

matrix plot is a subsample of the sample $\mathcal{S}$, selected by kernel herding, a method minimizing some discrepancy measure with the sample $\mathcal{S}$ that will be presented in Section 1.3). For this example, generating the kernel herding subsample takes under one minute, which is negligible compared with the simulation time of OWT models. Visually, this orange subsample seems to be representative of the original sample both in terms of marginal distributions and dependence structure. In the following study, the large samples $\mathcal{S}$ will be considered as an empirical representation of the distribution of the random vector $\mathbf{X} \in \mathcal{D}_{\mathbf{X}}$, with probability density function $f_{\mathbf{X}}$, and called *candidate set*. Kernel herding allows direct subsampling from this large and representative dataset, instead of fitting a joint distribution and generating samples from it. Indeed, fitting a joint distribution would introduce an additional source of error in the uncertainty propagation process. Note that a proper parametric model fit would be challenging for complex dependence structures such as the one plotted on Fig. 1.3. As examples of works

that followed this path, one can mention the work of Li and Zhang (2020) who built a parametric model of a similar multivariate distribution using vine copulas.

For a similar purpose and to avoid some limits imposed by the parametric framework, a nonparametric approach coupling empirical Bernstein copula fitting with kernel density estimation of the marginals is proposed in subSection 1.2.3.

1.2.3 Non parametric fit with empirical Bernstein copula

Instead of directly subsampling from a dataset such as the one from Fig. 1.3, one could first infer a multivariate distribution and generate a sample from it. However, accurately fitting such complex multivariate distributions is challenging. The amount of available data is large enough to make nonparametric inference a viable option.

The Sklar theorem (Durante and Sempi, 2015) states that the multivariate distribution of any random vector $\mathbf{X} \in \mathbb{R}^d, d \in \mathbb{N}^*$ can be broken down into two objects:

1. A set of univariate marginal distributions to describe the behavior of the individual variables;

2. A function describing the dependence structure between all variables, called a *copula*.

This theorem states that considering a random vector $\mathbf{X} \in \mathbb{R}^d$, with its cumulative distribution function F and its marginals $\{F_i\}_{i=1}^d$, there exists a copula $C : [0,1]^d \to [0,1]$, such that:

$$F(x_1, \ldots, x_d) = C\left(F_1(x_1), \ldots, F_d(x_d)\right). \tag{1.1}$$

It allows us to divide the problem of fitting a joint distribution into two independent problems: fitting the marginals and fitting the copula. The empirical Bernstein copula is a nonparametric copula approximation method introduced and applied on similar data in Chapter **??**. Provided a large enough learning set $\mathbf{X}_n$ (over five years in the present case), the EBC combined with kernel density estimation for the marginals enable to fit well the environmental joint distribution related to the dataset in Fig. 1.3. Moreover, the densities of the EBC are available in an explicit form, making Monte Carlo or quasi-Monte Carlo generation easy. As discussed in Chapter **??**, this method is sensitive to the chosen polynomial orders $\{m_j\}_{j=1}^d$ and the learning set size.

1.2.4 Fatigue assessment

As described in Fig. 1.1, a typical DIEGO simulation returns a 10-minute multiaxial stress time series at each node $i \in \mathbb{N}$ of the 1D meshed structure. Since classical fatigue laws are established for uniaxial stresses, the first step is to compute one equivalent Von Mises stress time series at each structural node. The present section recalls the main concepts but fatigue assessment is further discussed in Section **??**.

The foundation and the tower of an OWT are a succession of tubes with various sections connected by bolted or welded joints. Our work focuses on the welded joints at the mudline

level, identified as a critical area for the structure. This hypothesis is confirmed in the literature by different contributions, see e.g., the results of Müller and Cheng (2018) in Figure 12, or Katsikogiannis et al. (2021). At the top of the turbine, the fatigue is commonly studied at the blade root, which was not studied here since the blades in composite material have different properties (see e.g., Dimitrov (2013)). Note that the OWT simulations provide outputs allowing us to similarly study any node along the structure (without any additional computational effort).

To compute fatigue in a welded joint, the external circle of the welding ring is discretized for a few azimuth angles $\theta \in \mathbb{R}_+$ (see the red points in the monopile cross-section on the right in Fig. 1.4). The equivalent Von Mises stress time series is then reported on the external welding ring for an azimuth θ. According to Li and Zhang (2020) and our own experience, the most critical azimuth angles are roughly aligned with the main wind and wave directions (whose distributions are illustrated in Fig. 1.4). Looking at these illustrations, the wind and wave conditions have a very dominant orientation, which is explained by the closeness of the wind farm to the shore. Then, it is assumed that azimuth angles in these directions will be more solicited, leading to higher fatigue damage. To assess fatigue damage, rainflow counting (Dowling, 1972) first identifies the stress cycles and their respective amplitudes (using the implementation of the ASTM E1049-85 rainflow cycle counting algorithm from the Python package `rainflow`[4]). For each identified stress cycle of amplitude $s \in \mathbb{R}_+$, the so-called "Stress vs. Number of cycles" curve (also called the "SN curve" or "Wöhler curve") allows one to estimate the number N_c of similar stress cycles necessary to reach fatigue ruin. The SN curve, denoted by $W(\cdot)$ is an affine function in the log-log scale with slope $-m$ and y-intercept a:

$$N_c(s) = as^{-m},\ a \in \mathbb{R}_+,\ m \in \mathbb{R}_+. \tag{1.2}$$

Finally, a usual approach to compute the damage is to aggregate the fatigue contribution of each stress cycle identified using Miner's rule. Damage occurring during a 10-minute operating time is simulated and then scaled up to the OWT lifetime. More details regarding damage assessment and the Wöhler curve used are available in Section 2.4.6 from (DNV-RP-C203, 2016). For a realization $\mathbf{x} \in \mathcal{D}_{\mathbf{X}}$ of environmental conditions, at a structural node i, an azimuth angle θ; $k \in \mathbb{N}$ stress cycles of respective amplitude $\{s_{i,\theta}^{(j)}(\mathbf{x})\}_{j=1}^{k}$ are identified. Then, Miner's rule (Fatemi and Yang, 1998) defines the damage function $g_{i,\theta}(\mathbf{x}) : \mathcal{D}_{\mathbf{X}} \to \mathbb{R}_+$ by:

$$g_{i,\theta}(\mathbf{x}) = \sum_{j=1}^{k} \frac{1}{N_c\left(s_{i,\theta}^{(j)}(\mathbf{x})\right)}. \tag{1.3}$$

As defined by the DNV standards for OWT fatigue design (DNV-RP-C203, 2016), the quantity of interest in the present chapter is the "mean damage" $d_c^{i,\theta}$ (also called "cumulative damage"),

[4] https://github.com/iamlikeme/rainflow

computed at a node i, for an azimuth angle θ:

$$d_c^{i,\theta} = \mathbb{E}[g_{i,\theta}(\mathbf{X})] = \int_{\mathcal{D}_\mathbf{X}} g_{i,\theta}(\mathbf{x}) f_\mathbf{X}(\mathbf{x})\, \mathrm{d}\mathbf{x}\,. \tag{1.4}$$

To get a preview of the distribution of this output random variable $g_{i,\theta}(\mathbf{X})$, a histogram of a large Monte Carlo simulation ($N_{\text{ref}} = 2000$) is represented in Fig. 1.5 (with a log scale). In this case, the log-damage histogram presents a little asymmetry but it is frequently modeled by a normal distribution (see e.g., Teixeira et al. (2019b)).

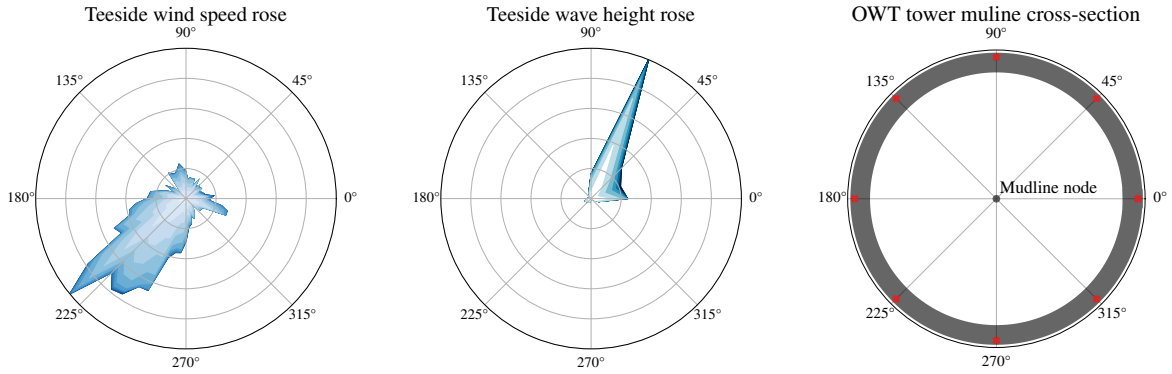

Figure 1.4 Angular distribution of the wind and waves with a horizontal cross-section of the OWT structure and the mudline. Red crosses represent the discretized azimuths for which the fatigue is computed

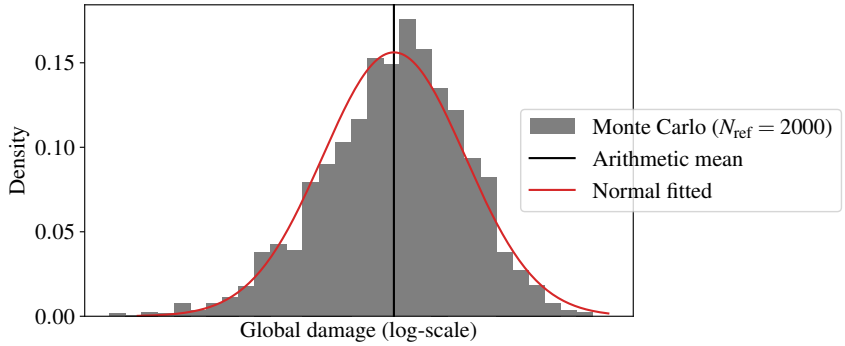

Figure 1.5 Histogram of the log-damage, at mudline, azimuth 45 deg. (Monte Carlo reference sample)

1.3 Numerical integration procedures for mean damage estimation

The present section explores different methods aiming at approximating the integral of a function against a probability measure. In the case of OWT mean damage estimation, these

methods can be used for defining efficient design load cases. This problem is equivalent to the central tendency estimation of $\mathbf{Y} = g(\mathbf{X})$, the image of the environmental random variable $\mathbf{X}$ by the damage function $g(\cdot) : \mathcal{D}_\mathbf{X} \to \mathbb{R}$ (see e.g., Eq. (1.4)). Considering a measurable space $\mathcal{D}_\mathbf{X} \subset \mathbb{R}^d, d \in \mathbb{N}^*$, associated with a known probability measure π, this section studies the approximation of integrals of the form $\int_{\mathcal{D}_\mathbf{X}} g(\mathbf{x})\mathrm{d}\pi(\mathbf{x})$.

1.3.1 Quadrature rules and quasi-Monte Carlo methods

Numerical integration authors may call this generic problem *probabilistic integration* (Briol et al., 2019). In practice, this quantity of interest is estimated on an n-sized set of damage realizations $\mathbf{y}_n = \left\{g(\mathbf{x}^{(1)}), \ldots, g(\mathbf{x}^{(n)})\right\}$ of an input sample $\mathbf{X}_n = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\right\}$. A weighted arithmetic mean of the realizations $\left\{g(\mathbf{x}^{(1)}), \ldots, g(\mathbf{x}^{(n)})\right\}$ is called a *quadrature rule* with a set of unconstrained weights $\mathbf{w}_n = \{w_1, \ldots, w_n\} \in \mathbb{R}^n$:

$$I_\pi(g) := \int_{\mathcal{D}_\mathbf{X}} g(\mathbf{x})\mathrm{d}\pi(\mathbf{x}) \approx \sum_{i=1}^{n} w_i g(\mathbf{x}^{(i)}). \tag{1.5}$$

Our numerical experiment framework often implies that the function g is costly to evaluate, making the realization number limited. For a given sample size n, our goal is to find a set of tuples $\left\{\mathbf{x}^{(i)}, w_i\right\}_{i=1}^n$ (i.e., quadrature rule), giving the best approximation of our quantity. In the literature, a large panel of numerical integration methods has been proposed to tackle this problem. For example, Van den Bos (2020) recently developed a quadrature rule based on arbitrary sample sets which was applied to a similar industrial OWT use case.

Alternatively, sampling methods rely on generating a set of points $\mathbf{X}_n$ drawn from the input distribution to compute the arithmetic mean of their realizations (i.e., uniform weights $\left\{w_i = \frac{1}{n}\right\}_{i=1}^n$). Among them, low-discrepancy sequences, also called "quasi-Monte Carlo" sampling (e.g., Sobol', Halton, Faure sequences) are known to improve the standard Monte Carlo convergence rate and will be used as a deterministic reference method in the following numerical experiments (Leobacher and Pillichshammer, 2014).

Quantization of probability measures and quadrature When dealing with probabilistic integration such as Eq. (1.5), a quadrature rule is a finite representation of a continuous measure π by a discrete measure $\zeta_n = \sum_{i=1}^n w_i \delta(\mathbf{x}^{(i)})$ (weighted sum of Dirac distributions at the design points $\mathbf{X}_n$). In the literature, this procedure is also called *quantization* of a continuous measure π. Overall, numerical integration is a particular case of probabilistic integration against a uniform input measure. For uniform measures, the Koksma-Hlawka inequality (Morokoff and Caflisch, 1995) provides a useful upper bound on the absolute error of a quadrature rule:

$$\left| \int_{[0,1]^d} g(\mathbf{x})\mathrm{d}\mathbf{x} - \frac{1}{n}\sum_{i=1}^{n} g(\mathbf{x}^{(i)}) \right| \leq V(g) D_n^*(\mathbf{X}_n). \tag{1.6}$$

As presented in Oates (2021), $V(g) = \sum_{\mathfrak{u}\subseteq\{1,\dots,p\}} \int_{[0,1]^{\mathfrak{u}}} \left|\frac{\partial^{\mathfrak{u}} g}{\partial \mathbf{x}_{\mathfrak{u}}}(\mathbf{x}_{\mathfrak{u}}, 1)\right| \mathrm{d}\mathbf{x}$, quantifies the complexity of the integrand, while $D_n^*(\mathbf{X}_n)$ evaluates the discrepancy to uniformity of the design $\mathbf{X}_n$. Therefore, the Koksma-Hlawka inequality shows that the quadrature rule's accuracy relies on the good quantization of π by $\mathbf{X}_n$. For a uniform measure π, the star discrepancy $D_n^*(\mathbf{X}_n)$ is a metric assessing how far from uniformity a sample $\mathbf{X}_n$ is. When generalizing to a non-uniform measure, a good quantization of π should also lead to a good approximation of the quantity.

1.3.2 Kernel herding sampling

Quasi-Monte Carlo sampling methods widely rely on a metric of uniformity, called *discrepancy*. To go beyond uniform measures, Appendix ?? introduces a kernel-based discrepancy, generalizing the discrepancy concept to non-uniform measures. This tool, name the maximum mean discrepancy (MMD) allows comparing multivariate distributions by embedding them in a specific function space. In this manuscript, the MMD was employed as a tool for statistical testing and quantifying the perturbations of distributions in Chapter ??, and for sensitivity analysis in Section ??.

Herin, the MMD is used to build a quadrature rule by sampling from a know measure. In other words, to quantize a known target measure π by a design sample $\mathbf{X}_n$. For practical reasons, the design construction is done sequentially. Sequential strategies have also been used to learn and validate regression models for statistical learning (see Fekhari et al. (2023)). Moreover, since each realization is supposed to be obtained at the same unitary cost, the quadrature weights are first fixed as uniform during the construction of the design $\mathbf{X}_n$.

Kernel herding (KH), proposed by Chen et al. (2010), is a sampling method that offers a quantization of the measure π by minimizing a squared MMD when adding points iteratively. With a current design $\mathbf{X}_n$ and its corresponding discrete distribution with uniform weights $\zeta_n = \frac{1}{n}\sum_{i=1}^n \delta(\mathbf{x}^{(i)})$, a KH iteration is as an optimization of the following criterion, selecting the point $\mathbf{x}^{(n+1)} \in \mathcal{D}_{\mathbf{X}}$:

$$\mathbf{x}^{(n+1)} \in \underset{\mathbf{x}\in\mathcal{D}_{\mathbf{X}}}{\arg\min} \left\{ \mathrm{MMD}\left(\pi, \frac{1}{n+1}\left(\delta(\mathbf{x}) + \sum_{i=1}^{n} \delta(\mathbf{x}^{(i)})\right)\right)^2 \right\}. \tag{1.7}$$

In the literature, two formulations of this optimization problem can be found. The first one uses the Frank-Wolfe algorithm (or "conditional gradient algorithm") to compute a linearization of the problem under the convexity hypothesis (see Lacoste-Julien et al. (2015) and Briol et al. (2015) for more details). The second one is a straightforward greedy optimization. Due to the combinatorial complexity, the greedy formulation is tractable for sequential construction only.

Let us develop the MMD criterion from Eq. (??):

$$\mathrm{MMD}\left(\pi, \frac{1}{n+1}\left(\delta(\mathbf{x}) + \sum_{i=1}^{n}\delta(\mathbf{x}^{(i)})\right)\right)^2 = \varepsilon_\pi - \frac{2}{n+1}\sum_{i=1}^{n+1} P_\pi\left(\mathbf{x}^{(i)}\right) + \frac{1}{(n+1)^2}\sum_{i,j=1}^{n+1} k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) \tag{1.8a}$$

$$= \varepsilon_\pi - \frac{2}{n+1}\left(P_\pi(\mathbf{x}) + \sum_{i=1}^{n} P_\pi\left(\mathbf{x}^{(i)}\right)\right) \tag{1.8b}$$

$$+ \frac{1}{(n+1)^2}\left(\sum_{i,j=1}^{n} k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) + 2\sum_{i=1}^{n} k\left(\mathbf{x}^{(i)}, \mathbf{x}\right) - k(\mathbf{x}, \mathbf{x})\right). \tag{1.8c}$$

In the previously developed expression, only a few terms actually depend on the next optimal point $\mathbf{x}^{(n+1)}$ since the target energy, denoted by ε_π, and $k(\mathbf{x}, \mathbf{x}) = \sigma^2$ are constant (by taking a stationary kernel). Therefore, the greedy minimization of the MMD can be equivalently written as:

$$\mathbf{x}^{(n+1)} \in \underset{\mathbf{x}\in\mathcal{D}_\mathrm{X}}{\arg\min}\left\{\frac{1}{n+1}\sum_{i=1}^{n} k\left(\mathbf{x}^{(i)}, \mathbf{x}\right) - P_\pi(\mathbf{x})\right\} = \underset{\mathbf{x}\in\mathcal{D}_\mathrm{X}}{\arg\min}\left\{\frac{n}{n+1}P_{\zeta_n}(\mathbf{x}) - P_\pi(\mathbf{x})\right\}. \tag{1.9}$$

Remark 1. For the sequential and uniformly weighted case, the formulation in Eq. (1.9) is almost similar to the Frank-Wolfe formulation. Our numerical experiments showed that these two versions generate very close designs, especially as n becomes large. Pronzato and Rendas (2021a) express the Frank-Wolfe formulation in the sequential and uniformly weighted case as follows:

$$\mathbf{x}^{(n+1)} \in \underset{\mathbf{x}\in\mathcal{D}_\mathrm{X}}{\arg\min}\left\{P_{\zeta_n}(\mathbf{x}) - P_\pi(\mathbf{x})\right\}. \tag{1.10}$$

Remark 2. In practice, the optimization problem is solved by a brute-force approach on a fairly dense finite subset $\mathcal{S} \subseteq \mathcal{D}_\mathrm{X}$ of candidate points with size $N \gg n$ that emulates the target distribution, also called the "candidate set". This sample is also used to estimate the target potential $P_\pi(\mathbf{x}) \approx \frac{1}{N}\sum_{i=1}^{N} k\left(\mathbf{x}^{(i)}, \mathbf{x}\right)$.

The diagram illustrated in Fig. 1.6 summarizes the main steps of a kernel herding sampling algorithm. One can notice that the initialization can either be done using a median point (maximizing the target potential) or from any existing design of experiments. This second configuration showed to be practical when the analyst must include some characteristic points in the design (e.g., points with a physical interpretation).

As explained previously, choosing the kernel defines the function space on which the worst-case function is found (see Eq. (??)). Therefore, this sampling method is sensitive to the kernel's choice. A kernel is defined, both by the choice of its parametric family (e.g., Matérn, squared exponential) and the choice of its tuning. The so-called "support points" method developed by Mak and Joseph (2018) is a special case of kernel herding that uses the

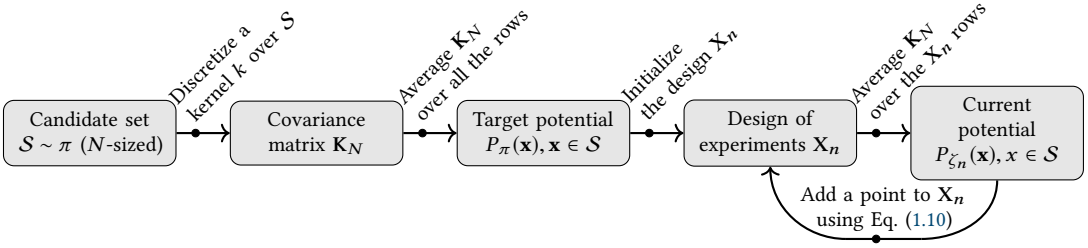

Figure 1.6 Greedy kernel herding algorithm

Energy-distance	$k_E(\mathbf{x},\mathbf{x}') = \frac{1}{2}\left(\|\mathbf{x}\| + \|\mathbf{x}'\| - \|\mathbf{x}-\mathbf{x}'\|\right)$	
Squared exponential	$k_G(\mathbf{x},\mathbf{x}') = \prod_{i=1}^{p} k_{\theta_i}(x_i - x_i')$	$k_\theta(x-x') = \exp\left(-\frac{(x-x')^2}{2\theta^2}\right)$
Matérn ($\nu = 5/2$)	$k_M(\mathbf{x},\mathbf{x}') = \prod_{i=1}^{p} k_{5/2,\theta_i}(x_i - x_i')$	$k_{5/2,\theta}(x-x') = \left(1 + \frac{\sqrt{5}}{\theta}\lvert x-x'\rvert + \frac{5}{3\theta^2}(x-x')^2\right)\exp\left(-\frac{\sqrt{5}}{\theta}\lvert x-x'\rvert\right)$

Table 1.3 Kernels considered in the following numerical experiments.

characteristic and parameter-free "energy-distance" kernel (introduced by Székely and Rizzo (2013)). In the following numerical experiments, the energy-distance kernel will be compared with an isotropic tensor product of a Matérn kernel (with regularity parameter $\nu = 5/2$ and correlation lengths θ_i), or a squared exponential kernel (with correlation lengths θ_i) defined in Table 1.3. Since the Matérn and squared exponential kernels are widely used for Gaussian process regression (Rasmussen and Williams, 2006), they were naturally picked to challenge the energy-distance kernel. The correlation lengths for the squared exponential and Matérn kernels are set using the heuristic given in Fekhari et al. (2023), $\theta_i = n^{-1/d}, i \in \{1,\dots,d\}$.

Fig. 1.7 represents the covariance structure of the three kernels. One can notice that the squared exponential and Matérn $\nu = 5/2$ kernels are closer to one another than they are to the energy-distance. In fact, as ν tends to infinity, the Matérn kernel tends toward the squared exponential kernel (which has infinitely differentiable sample paths, see Rasmussen and Williams (2006)). For these two stationary kernels, the correlation length controls how fast the correlation between two points decreases as their distance from one another increases.

Meanwhile, the energy distance is not stationary (but still positive and semi-definite). Therefore, its value does not only depend on the distance between two points but also on the norm of each of the points. Interestingly, the energy-distance kernel is almost similar to the kernel used by Hickernell (1998) to define a widely-used space-filling metric called the centered L^2-discrepancy. A presentation of these kernel-based discrepancies from the design of experiment point of view is also provided in Chapter Two from Fang et al. (2018).

To illustrate the kernel herding sampling of a complex distribution, Fig. 1.8 shows three nested samples (orange crosses for different sizes $n \in \{10, 20, 40\}$) of a mixture of Gaussian distributions with complex nonlinear dependencies (with density represented by the black isoprobability contours). In this example, the method seems to build a parsimonious design

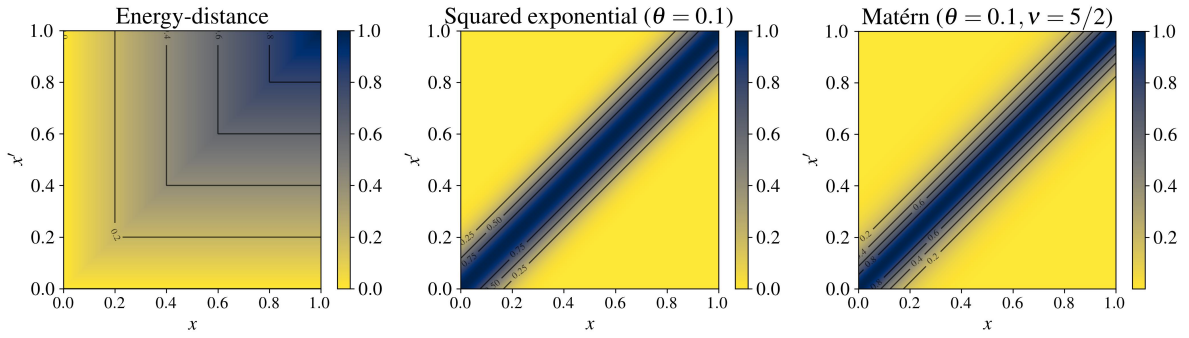

Figure 1.7 Kernel illustrations (left to right: energy-distance, squared exponential, and Matérn 5/2)

between each mode of the distribution (by subsampling directly without any transformation). The candidate set (in light grey) was generated by a large quasi-Monte sample of the underlying Gaussian mixture. In this two-dimensional case, this candidate set is sufficient to estimate the target potential P_π. However, the main bottleneck of kernel herding is the estimation of the potentials, which becomes costly in high dimensions.

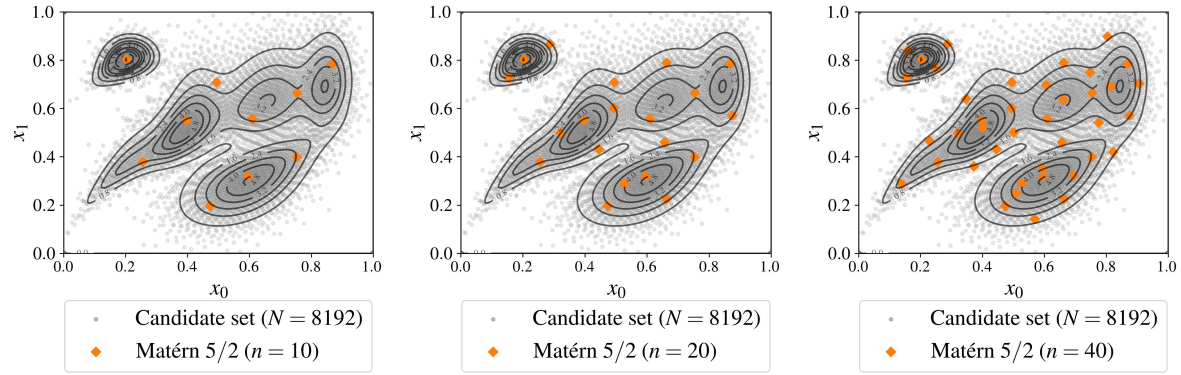

Figure 1.8 Sequential kernel herding for increasing design sizes ($n \in \{10, 20, 40\}$) built on a candidate set of $N = 8196$ points drawn from a complex Gaussian mixture π

Other approaches take advantage of the progressive knowledge acquired sequentially from the outputs to select the following points in the design. These methods are sometimes called "active learning" or "adaptive strategies" (Fuhg. et al., 2021). Many of them rely on a sequentially updated Gaussian process (or Kriging) metamodel. To solve a probabilistic integration problem, the concept of Bayesian quadrature is introduced in the following.

1.3.3 Bayesian quadrature

Gaussian processes for Bayesian quadrature Kernel methods and Gaussian processes present a lot of connections and equivalences, thoroughly reviewed by Kanagawa et al. (2018). In numerical integration, Gaussian processes have been used to build quadrature rules in the seminal paper of O'Hagan (1991), introducing the concept of *Bayesian quadrature* (BQ). Let us

recall the probabilistic integration problem $I_\pi(g) = \int_{\mathcal{D}_\mathbf{X}} g(\mathbf{x})\mathrm{d}\pi(\mathbf{x})$ (stated in Eq. (1.5)). From a general point of view, this quantity could be generalized by composing g with another function ψ (e.g., other moments, quantiles, exceedance probabilities). The quantity of interest then becomes, $I_\pi(\psi(g))$, for example when ψ is a monomial, it gives a moment of the output distribution.

Let us assume, adopting a Bayesian point of view, that G is a stochastic process describing the uncertainty affecting the knowledge about the true function g. Let G be a Gaussian process (GP) prior with a zero trend (denoted by $\mathbf{0}$) to ease the calculation, and a stationary covariance kernel (denoted by $k(\cdot,\cdot)$). The conditional posterior $G_n := (G|\mathbf{y}_n) \sim \mathrm{GP}(\eta_n, s_n^2)$ has been conditioned on the function observations $\mathbf{y}_n = \left[g\left(\mathbf{x}^{(1)}\right), \ldots, g\left(\mathbf{x}^{(n)}\right)\right]^\top$ computed from the input design $\mathbf{X}_n$ and is fully defined by the well-known "Kriging equations" (see e.g., Rasmussen and Williams (2006)):

$$\begin{cases} \eta_n(\mathbf{x}) & := \mathbf{k}_n^\top(\mathbf{x})\mathbf{K}_n^{-1}\mathbf{y}_n \\ s_n^2(\mathbf{x}) & := k_n(\mathbf{x},\mathbf{x}) - \mathbf{k}_n^\top(\mathbf{x})\mathbf{K}_n^{-1}\mathbf{k}_n(\mathbf{x}) \end{cases} \tag{1.11}$$

where $\mathbf{k}_n(\mathbf{x})$ is the column vector of the covariance kernel evaluations $[k_n(\mathbf{x},\mathbf{x}^{(1)}), \ldots, k_n(\mathbf{x},\mathbf{x}^{(n)})]$ and $\mathbf{K}_n$ is the $(n\times n)$ variance-covariance matrix such that the (i,j)-element is $\{\mathbf{K}_n\}_{i,j} = k_n(\mathbf{x}^{(i)},\mathbf{x}^{(j)})$.

In BQ, the main object is the random variable $I_\pi(G_n)$. According to Briol et al. (2019), its distribution on $\mathbb{R}$ is the pushforward of G_n through the integration operator $I_\pi(\cdot)$, sometimes called *posterior distribution*:

$$I_\pi(G_n) = \int_{\mathcal{D}_\mathbf{X}} (G(\mathbf{x})|\mathbf{y}_n)\mathrm{d}\pi(\mathbf{x}) = \int_{\mathcal{D}_\mathbf{X}} G_n(\mathbf{x})\mathrm{d}\pi(\mathbf{x})\,. \tag{1.12}$$

Fig. 1.9 provides a one-dimensional illustration of the Bayesian quadrature of an unknown function (dashed black curve) against a given input measure π (with corresponding grey distribution at the bottom). For an arbitrary design, one can fit a Gaussian process model, interpolating the function observations (black crosses). Then, multiple trajectories of this conditioned Gaussian process G_n are drawn (orange curves) whilst its mean function, also called "predictor", is represented by the red curve. Therefore, the input measure π is propagated through the conditioned Gaussian process to obtain the random variable $I_\pi(G_n)$, with distribution represented on the right plot (brown curve). Again on the right plot, remark how the mean of this posterior distribution (brown line) is closer to the reference output expected value (dashed black line) than the arithmetic mean of the observations (black line). This plot was inspired by the paper of Huszár and Duvenaud (2012).

Optimal weights computed by Bayesian quadrature Taking the random process G_n as Gaussian conveniently implies that its posterior distribution $a_\pi(G_n)$ is also Gaussian. This comes from the linearity of the infinite sum of realizations of a Gaussian process. The posterior distribution is described in a closed form through its mean and variance by applying Fubini's theorem (see the supplementary materials from Briol et al. (2019) for the proof regarding the

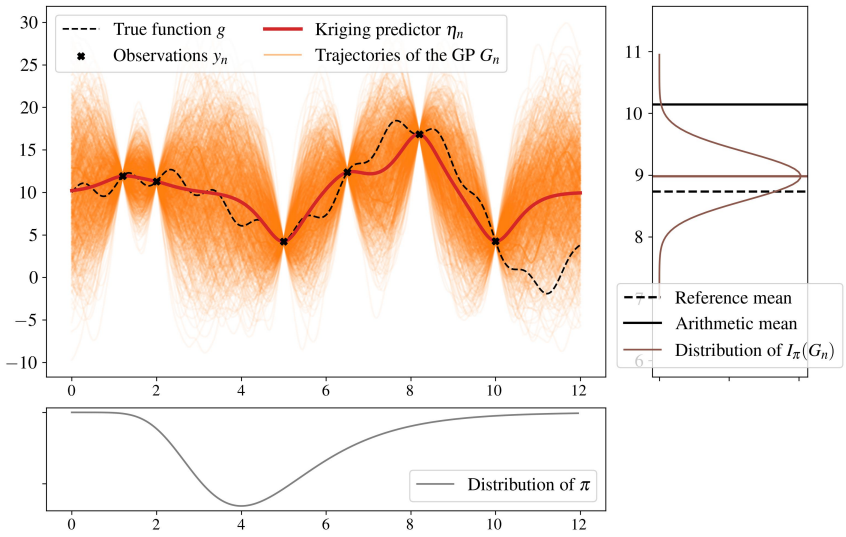

Figure 1.9 Bayesian quadrature on a one-dimensional case

variance):

$$\overline{y}_n^{\mathrm{BQ}} := \mathbb{E}\left[I_\pi(G_n)|\mathbf{y}_n\right] = \int_{\mathcal{D}_\mathbf{X}} \eta_n(\mathbf{x})\mathrm{d}\pi(\mathbf{x}) = \left[\int_{\mathcal{D}_\mathbf{X}} \mathbf{k}_n^\top(\mathbf{x})\mathrm{d}\pi(\mathbf{x})\right]\mathbf{K}_n^{-1}\mathbf{y}_n = P_\pi(\mathbf{X}_n)\mathbf{K}_n^{-1}\mathbf{y}_n, \tag{1.13}$$

$$\left(\sigma_n^{\mathrm{BQ}}\right)^2 := \mathrm{Var}\left(I_\pi(G_n)\right) = \iint_{\mathcal{D}_{\mathbf{X}^2}} k_n(\mathbf{x},\mathbf{x}')\mathrm{d}\pi(\mathbf{x})\mathrm{d}\pi(\mathbf{x}') = \varepsilon_\pi - P_\pi(\mathbf{X}_n)\mathbf{K}_n^{-1}P_\pi(\mathbf{X}_n)^\top. \tag{1.14}$$

Where $P_\pi(\mathbf{X}_n)$ is the row vector of potentials $\left[\int k_n(\mathbf{x},\mathbf{x}^{(1)})\mathrm{d}\pi(\mathbf{x}),\ldots,\int k_n(\mathbf{x},\mathbf{x}^{(n)})\mathrm{d}\pi(\mathbf{x})\right]$, and ε_π is given in Eq. (**??**). As in the one-dimensional example presented in Fig. 1.9, the expected value of $I_\pi(G_n)$ expressed in Eq. (1.13) is a direct estimator of the quantity of interest Eq. (1.5). The so-called "Bayesian quadrature estimator" appears to be a simple linear combination of the observations by taking the row vector of "optimal weights" as:

$$\mathbf{w}_{\mathrm{BQ}} := P_\pi(\mathbf{X}_n)\mathbf{K}_n^{-1} \tag{1.15}$$

For any given sample, an optimal set of weights can be computed, leading to the mean of the posterior distribution. Remark here that this enhancement depends on the evaluation of the inverse variance-covariance matrix $\mathbf{K}_n^{-1}$, which can present numerical difficulties, either when design points are too close, making the conditioning bad. Moreover, a prediction interval on the BQ estimator can be computed since the posterior distribution is Gaussian, with a variance expressed in closed-form in Eq. (1.14). The expressions in Eq. (1.13) and Eq. (1.14) were extended to Gaussian processes in the case of constant and linear trends in Pronzato and Zhigljavsky (2020). In the following numerical experiments, the expression with a hypothesis of constant

trend β_n is used, which leads to:

$$\mathbb{E}\left[I_\pi(G_n)\right] = \beta_n + P_\pi(\mathbf{X}_n)\mathbf{K}_n^{-1}\left(\mathbf{y}_n - \beta_n \mathbf{1}_n\right). \tag{1.16}$$

Then, an a posteriori 95% prediction interval around the mean Bayesian estimator is directly given by:

$$\overline{y}_n^{\mathrm{BQ}} \in \left[\overline{y}_n^{\mathrm{BQ}} - 2\sigma_n^{\mathrm{BQ}}, \overline{y}_n^{\mathrm{BQ}} + 2\sigma_n^{\mathrm{BQ}}\right]. \tag{1.17}$$

Variance-based Bayesian quadrature rule The link between the posterior variance and the squared MMD has been first made by Huszár and Duvenaud (2012) in their Proposition 1: the expected variance in the Bayesian quadrature $\mathrm{Var}\left(I_\pi(G_n)\right)$ is the MMD between the target distribution π and $\zeta_n = \sum_{i=1}^n \mathbf{w}_{\mathrm{BQ}}^{(i)} \delta(\mathbf{x}^{(i)})$. The proof is reproduced below (as well as in Proposition 6.1 from Kanagawa et al. (2018)):

$$\begin{aligned}
\mathrm{Var}\left(I_\pi(G_n)\right) &= \mathbb{E}\left[\left(I_\pi(G_n) - I_{\zeta_n}(G_n)\right)^2\right] && \text{(1.18a)}\\
&= \mathbb{E}\left[\left(\langle G_n, P_\pi\rangle_{\mathcal{H}(k)} - \left\langle G_n, P_{\zeta_n}\right\rangle_{\mathcal{H}(k)}\right)^2\right] && \text{(1.18b)}\\
&= \mathbb{E}\left[\left\langle G_n, P_\pi - P_{\zeta_n}\right\rangle_{\mathcal{H}(k)}^2\right] && \text{(1.18c)}\\
&= \|P_\pi - P_{\zeta_n}\|_{\mathcal{H}(k)}^2 && \text{(1.18d)}\\
&= \mathrm{MMD}(\pi, \zeta_n)^2. && \text{(1.18e)}
\end{aligned}$$

Note that the transition from equation (27c) to (27d) relies on the property stating that if G is a standard Gaussian process then $\forall g \in \mathcal{H}(k) : \langle G, g\rangle_{\mathcal{H}(k)} \sim \mathcal{N}(0, \|g\|_{\mathcal{H}(k)}^2)$. The method that sequentially builds a quadrature rule by minimizing this variance is called by the authors "Sequential Bayesian Quadrature" (SBQ). According to the previous proof, this criterion can be seen as an optimally-weighted version of the kernel herding criterion, as stated in the title of the paper from Huszár and Duvenaud (2012). Later, Briol et al. (2015) proved the weak convergence of $I_\pi(G_n)$ towards the target integral. Closer to wind turbines applications, Huchet (2019) and Huchet et al. (2019) introduced the "Adaptive Kriging Damage Assessment" method: a Kriging-based method for mean damage estimation that is very close to SBQ. However, This type of method inherits the limits from both KH and BQ since it searches for optimal design points among a candidate set and computes an inverse variance-covariance matrix. These numerical operations both scale hardly in high dimension.

Remark 3. Every quadrature method introduced in this section has been built without any observation of the possibly costly function g. Therefore, they cannot be categorized as active learning approaches. Contrarily, Kanagawa and Hennig (2019) presents a set of methods for BQ with transformations (i.e., adding a positivity constraint on the function g), which are truly active learning methods.

1.4 Numerical experiments

This section presents numerical results computed on two different analytical toy cases, respectively in dimension 2 (toy case #1) and dimension 10 (toy case #2), with easy-to-evaluate functions $g(\cdot)$ and associated input distributions π. Therefore, reference values can easily be computed with great precision. For each toy case, a large reference Monte Carlo sample ($N_{\text{ref}} = 10^8$) is taken. This first benchmark compares the mean estimation of toy cases given by a quasi-Monte Carlo technique (abbreviated by QMC in the next figures) which consists herein using a Sobol' sequence, and kernel herding with the three kernels defined in Table 1.3. Notice that the quasi-Monte Carlo designs are first generated on the unit hypercube and then, transformed using the generalized Nataf transformation to follow the target distribution (Lebrun and Dutfoy, 2009). Additionally, the performances of kernel herding for both uniform and optimally-weighted Eq. (1.16) estimators are compared.

All the following results and methods (i.e., the kernel-based sampling and BQ methods) have been implemented in a new open-source Python package named `otkerneldesign`[5]. This development mostly relies on the open source software OpenTURNS[6] ("Open source initiative for the Treatment of Uncertainties, Risks'N Statistics") devoted to uncertainty quantification and statistical learning (Baudin et al., 2017). Finally, note that the numerical experiments for the toy cases are available in the Git repository named `ctbenchmark`[7].

1.4.1 Illustration on analytical toy-cases

The toy cases were chosen to cover a large panel of complex probabilistic integration problems, completing the ones from Fekhari et al. (2022). To assess the complexity of numerical integration problems, Owen (2003) introduced the concept of the "effective dimension" of an integrand function (number of the variables that actually impact the integral). The author showed that functions built on sums yield a low effective dimension (unlike functions built on products). In the same vein, Kucherenko et al. (2011) build three classes of integrand sorted by difficulty depending on their effective dimension:

- *class A*: problem with a few dominant variables.
- *class B*: problem without unimportant variables, and important low-order interaction terms.
- *class C*: problems without unimportant variables, and important high-order interaction terms.

The 10-dimensional "GSobol function" (toy case #2) with a set of coefficient $\{a_i = 2\}_{i=1}^{10}$ has an effective dimension equal to 10 and belongs to the hardest class C from Kucherenko et al. (2011).

[5] https://efekhari27.github.io/otkerneldesign/master/index.html
[6] https://openturns.github.io/www/
[7] https://github.com/efekhari27/ctbenchmark

Table 1.4 Analytical toy-cases

Toy-case #1	$dim = 2$	$g_1(\mathbf{x}) = x_1 + x_2$	Gaussian mixture from Fig. 1.8
Toy-case #2	$dim = 10$	$g_2(\mathbf{x}) = \prod_{i=1}^{10} \frac{\lvert 4x_i - 2\rvert + a_i}{1+a_i}, \{a_i = 2\}_{i=1}^{10}$	Gaussian $\mathcal{N}(\mathbf{0.5}, \mathbf{I}_{10})$

In the case of the two-dimensional Gaussian mixture problem, the complexity is carried by the mixture of Gaussian distributions with highly nonlinear dependencies. Probabilistic integration results are presented in Fig. 1.10 (toy case #1) and Fig. 1.11 (toy case #2). Kernel herding samples using the energy-distance kernel are in red, while quasi-Monte Carlo samples built from Sobol' sequences are in grey. Convergences of the arithmetic means are plotted on the left and MMDs on the right. The respective BQ estimators of the means are plotted in dashed lines.

Remark 4. Different kernels are used in these numerical experiments. First, the generation kernel, used by the kernel herding algorithm to generate designs (with the heuristic tuning defined in Section 1.3.2). Second, the BQ kernel allows computation of the optimal weights (arbitrarily set up as a Matérn 5/2 with the heuristic tuning). Third, the evaluation kernel, which must be common to allow a fair comparison of the computed MMD results (same as the BQ kernel).

Results analysis for toy case #1. Convergence plots are provided in Fig. 1.10. KH consistently converges faster than quasi-Monte Carlo in this case, especially for small sizes in terms of MMD. BQ weights tend to reduce the fluctuations in the mean convergence, which ensures better performance for any size. Overall, applying the weights enhances the convergence rate.

Results analysis for toy case #2. Convergence plots are provided in Fig. 1.11. Although quasi-Monte Carlo is known to suffer the "curse of dimensionality", KH does not outperform it drastically in this example. In fact, KH with uniform weights performs worse than quasi-Monte Carlo while optimally-weighted KH does slightly better. Moreover, the results confirm that $\text{MMD}_{\text{BQ}} < \text{MMD}_{\text{unif}}$ for all our experiments. The application of optimal weights to the quasi-Monte Carlo sample slightly improves the estimation in this case. Note that the prediction interval around the BQ estimator is not plotted for the sake of readability.

In these two toy cases, the MMD is shown to quantify numerical integration convergence well, which illustrates the validity of the inequality given in Eq. (**??**), similar to the Koksma-Hlawka inequality (as recalled in Eq. (1.6)).

1.4.2 Application to the Teesside wind turbine fatigue estimation

Let us summarize the mean damage estimation strategies studied in this chapter. The diagram represented in Fig. 1.12 describes the different workflows computed. The simplest workflow is represented by the grey horizontal sequence. It directly subsamples a design of experiments from a large and representative dataset (previously referred to as candidate set). This workflow simply estimates the mean damage by computing an arithmetic average of the outputs.

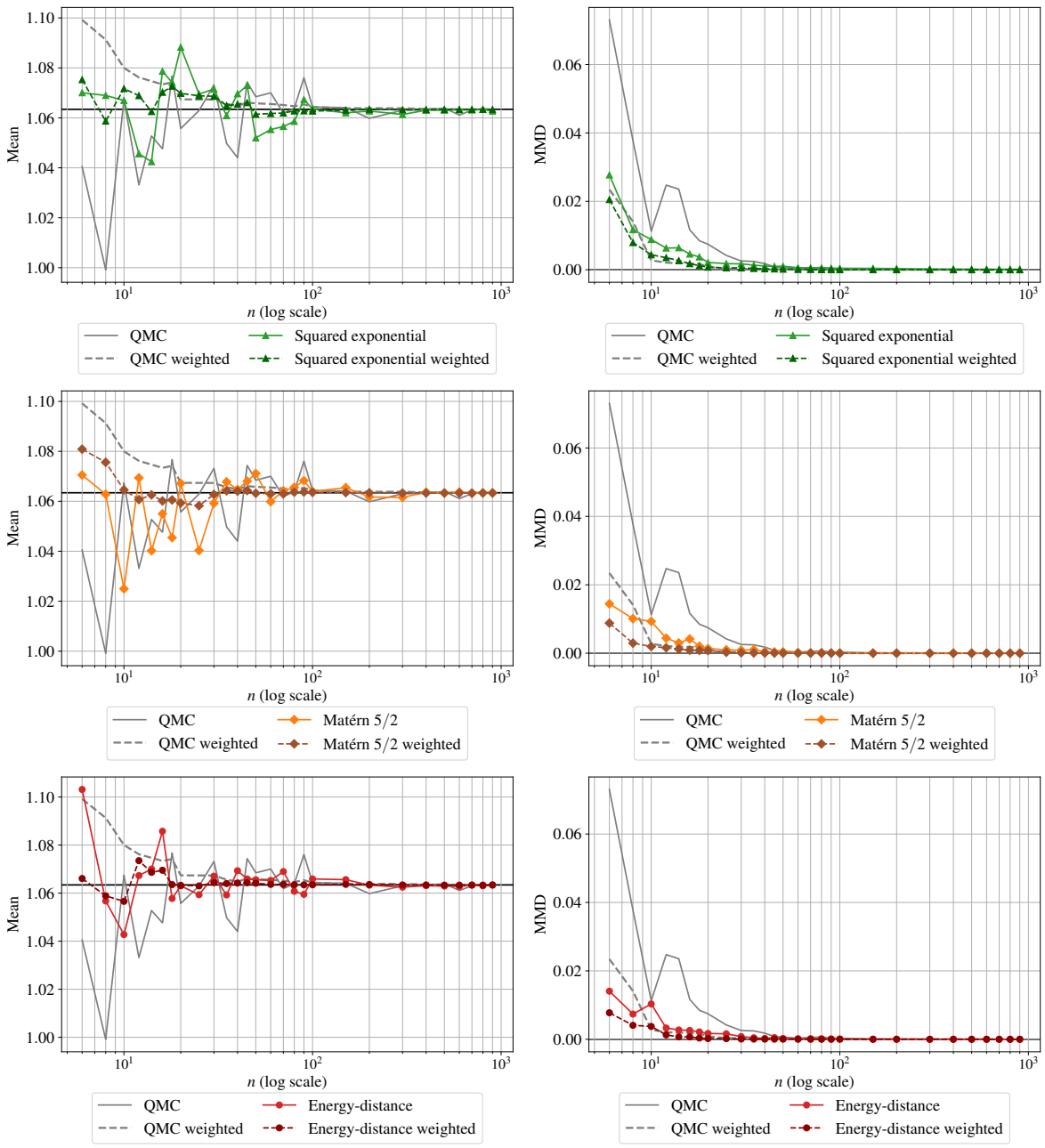

Figure 1.10 Analytical benchmark results on the toy-case #1

Alternatively, one can respectively fit a joint distribution and sample from it. In our case, this distribution is only known empirically via the candidate set. Since its dependence structure is complex (see Fig. 1.3), a parametric method might fit the distribution poorly (and therefore lead to a poor estimation of the quantity). Then, a nonparametric fit using the empirical Bernstein copula (introduced in Section 1.2.3) coupled with a kernel density estimation on each marginal is applied to the candidate set (with the EBC parameter $m = 100 > m_{\text{MISE}}$ to avoid bias, see Lasserre (2022) p.117). The sampling on this hybrid joint distribution is realized with a quasi-Monte Carlo method. A Sobol' low-discrepancy sequence generates a uniform sample in the unit hypercube, which can then be transformed according to a target distribution. Remember

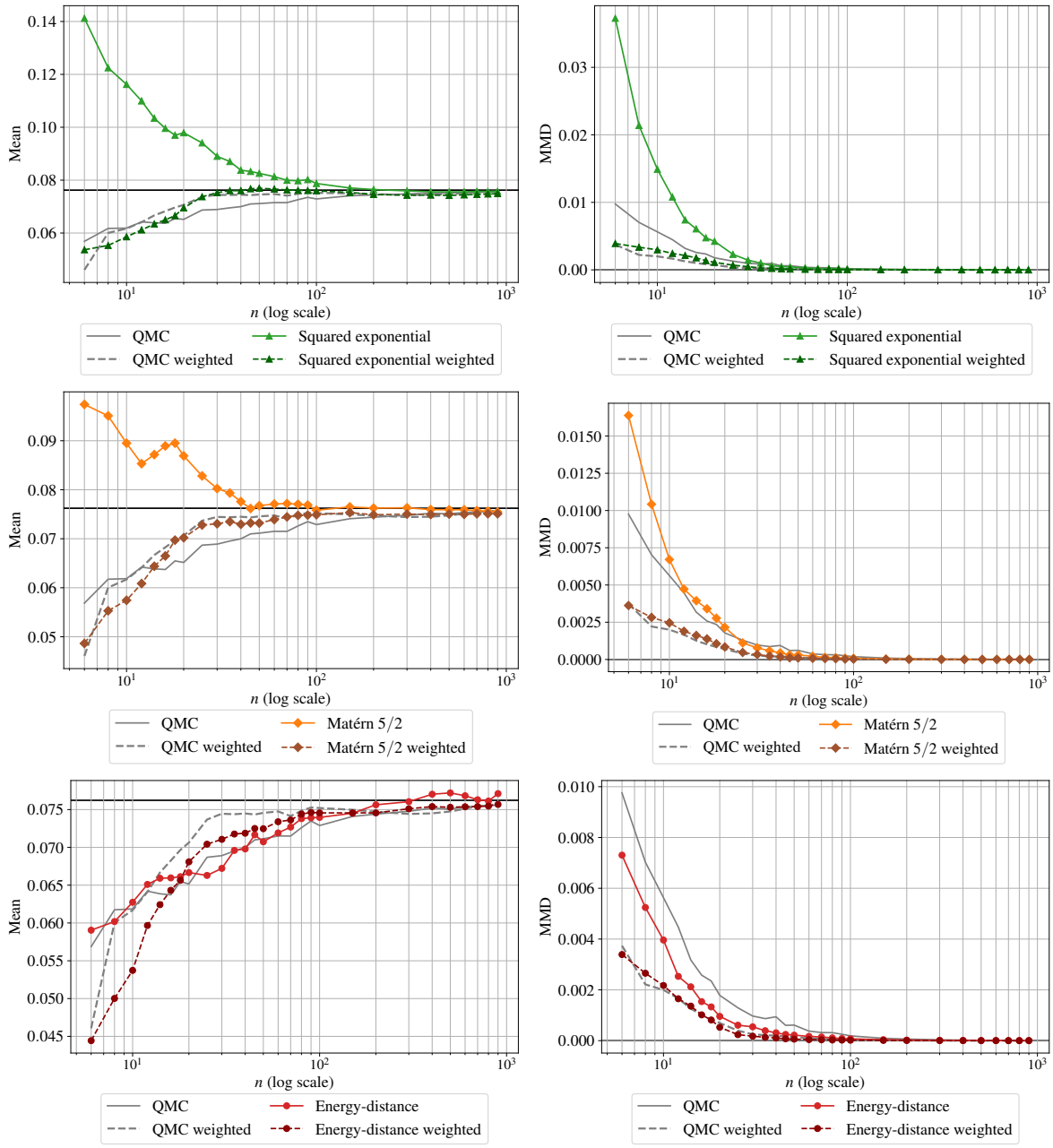

Figure 1.11 Analytical benchmark results on the toy-case #2

that quasi-Monte Carlo sampling is also sensitive to the choice of a low-discrepancy sequence, each presenting different properties (e.g., Sobol', Halton, Faure, etc.). Finally, the estimation by an arithmetic mean can be replaced by an optimally weighted mean. To do so, optimal weight must be computed, using the formulas introduced in Eq. (1.15).

The copulogram in Fig. 1.13 illustrates the intensity of the computed damages, proportionally to the color scale. Note that the numerical values of the damage scale are kept confidential since it models the state of an operating asset. Before analyzing the performance of the KH on this industrial application, let us notice that the copulogram Fig. 1.13 seems to be in line with the global sensitivity analysis presented in Murcia et al. (2018) and Li and Zhang (2020).

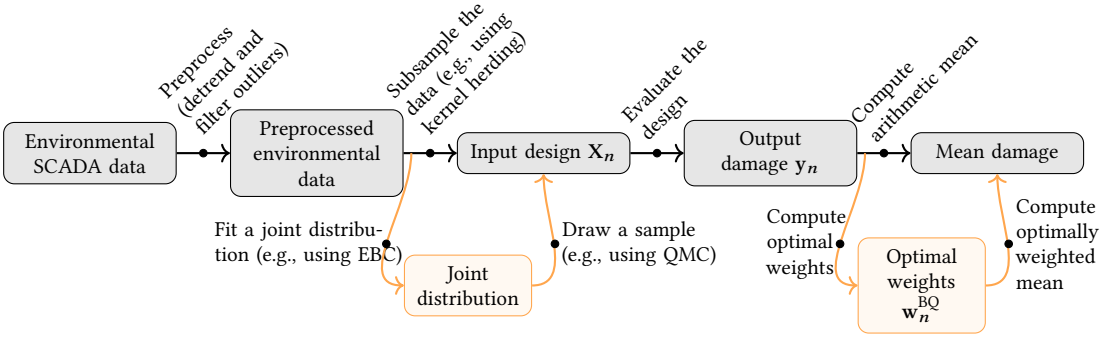

Figure 1.12 Mean damage estimation workflows for the industrial use case. The orange parts represent optional alterations to the workflow: the first one is an alternative to input data subsampling where the underlying distribution is sampled from, the second one improves mean damage calculation by using optimal weights over the output data

In particular, the fact that the scatter plot of mean wind speed vs. turbulence wind speed is the main factor explaining the variance of the output $Y = g(\mathbf{X})$. Judging from these references, the numerical model does not seem to have a highly effective dimension, however, the input dependence structure is challenging and the damage assessment induces strong nonlinearities (see Eq. (1.2)).

The results presented are compared in the following to a large reference Monte Carlo sample (size 2000) with a confidence interval computed by bootstrap (see Fig. 1.14). This reference is represented by a horizontal line intersecting with the most converged Monte Carlo estimation. Once again, the mean damage scale is hidden for confidentiality reasons, but all the plots are represented for the same vertical scale. The performance of the KH is good: it quickly converges towards the confidence interval of the Monte Carlo obtained with the reference sample. In addition, the Bayesian quadrature estimator also offers a posteriori prediction interval, which can reassure the user. The BQ prediction intervals are smaller than the ones obtained by bootstrap on the reference Monte Carlo sample.

To provide more representative results, note that a set of scale parameters is computed with a kriging procedure to define the kernel used to compute BQ intervals. Since other methods do not generate independent samples, bootstrapping them is not legitimate. Contrarily to the other kernels, we observe that the energy-distance kernel presents a small bias with the MC reference for most of the azimuth angles computed in this experiment. Meanwhile, combining nonparametric fitting with quasi-Monte Carlo sampling also delivers good results as long as the fitting step does not introduce a bias. In our case, any potential bias due to poor fitting would be the result of a poorly tuned empirical Bernstein copula. Fortunately, Nagler et al. (2017) formulated recommendations regarding how to tune empirical Bernstein copulas. We follow these recommendations in the present work.

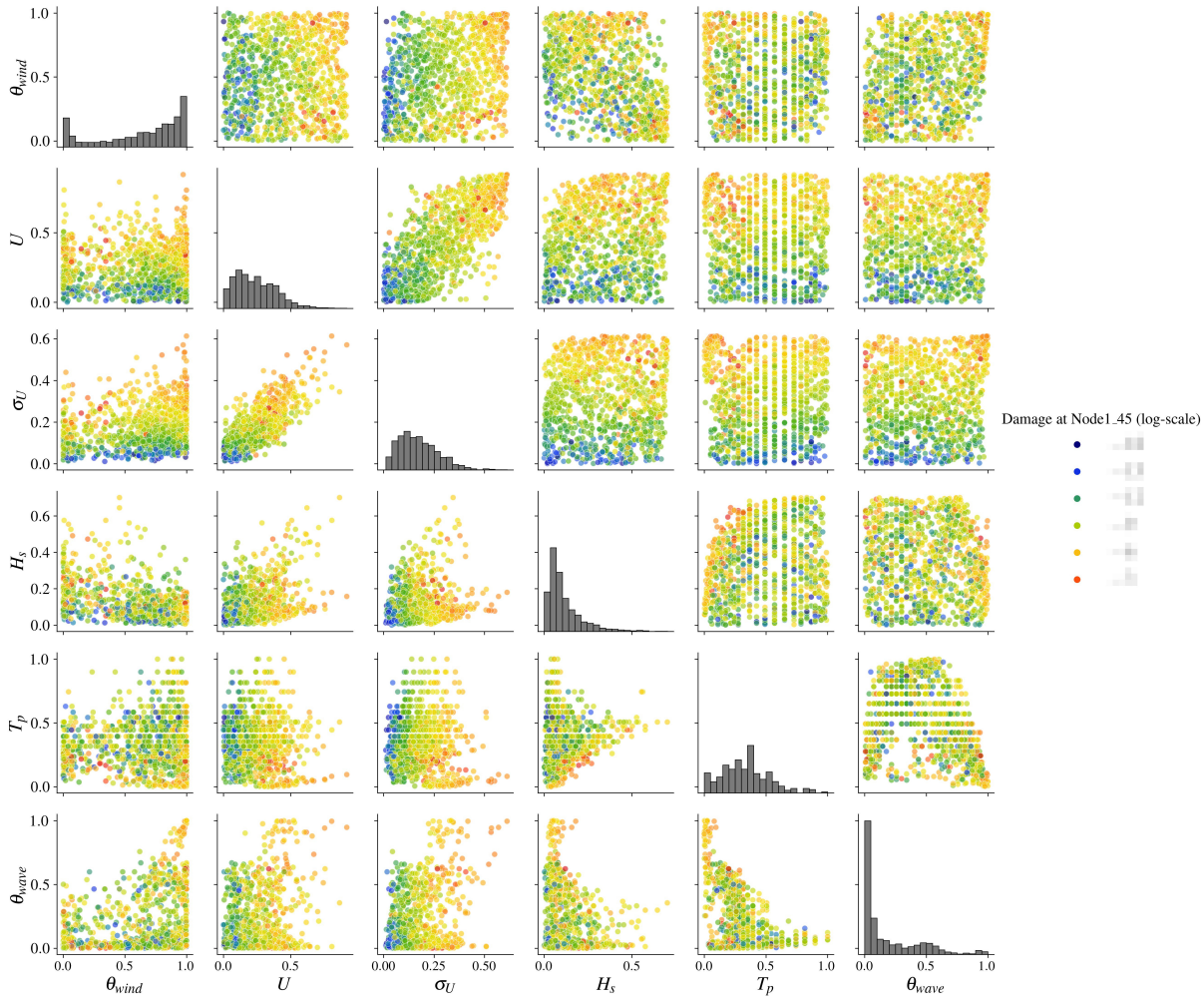

Figure 1.13 Copulogram of the kernel herding design of experiments with corresponding outputs in color (log-scale) on the Teesside case ($n = 10^3$). The color scale ranges from blue for the lowest values to red for the largest. Marginals are represented by histograms (diagonal), the dependence structure with scatter plots in the ranked space (upper triangle). Scatter plots on the bottom triangle are set in the physical space.

1.5 Conclusion

Wind energy assets are subject to highly uncertain environmental conditions. Uncertainty propagation through numerical models is performed to ensure their structural integrity (and energy production). For this case, the method recommended by the standards (regular grid sampling) is intractable for even moderate-fidelity simulators. In practice, such an approach can lead to poor uncertainty propagation, especially when facing simulation budget constraints.

In the present chapter, a real industrial wind turbine fatigue estimation use case is investigated, considering site-specific data. As a perspective, other sites with different environmental conditions could be studied. This use case induces two practical constraints: first, usual active learning methods are hard to set up on such a model (mainly due to the nonlinearity of the variable of interest), and they restrict the use of high-performance computing facilities; second,

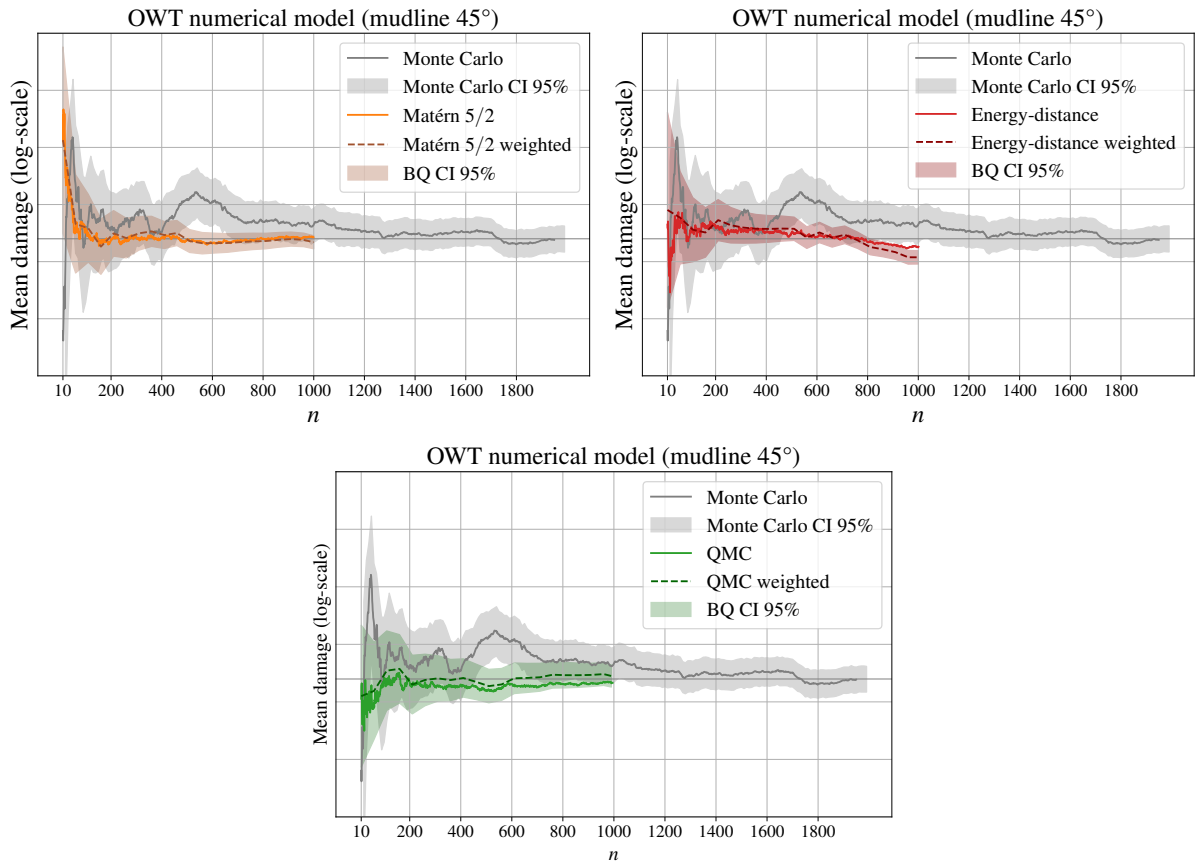

Figure 1.14 Mean estimation convergence (at the mudline, azimuth $\theta = 45$ deg.) on the Teesside case. Monte Carlo confidence intervals are all computed by bootstrap

the input distribution of the environmental conditions presents a complex dependence structure which is hard to infer with common parametric approaches.

In this work, the association of kernel herding sampling with Bayesian quadrature for central tendency estimation is explored theoretically and numerically. This method fits with the practical constraints induced by the industrial use case. To be more specific, the kernel herding method easily subsamples the relevant points directly from a given dataset (here, from the measured environmental data). Moreover, the method is fully compatible with intensive high-performance computer use. Moreover, the present work outlined an upper bound based on the maximum mean discrepancy (MMD) on numerical integration absolute error. Kernel herding and Bayesian quadrature both aim at finding the quadrature rule minimizing the MMD, and therefore the absolute integration error. The numerical experiments confirm that the MMD is an appropriate criterion since it leads to results being better or equivalent to quasi-Monte Carlo sampling. Finally, the proposed numerical benchmark relies on a Python package, called `otkerneldesign`, which implements the methods and allows anyone to reproduce the results.

The limits of the proposed method are reached when the input dimension of the problem increases, requiring a larger candidate set and therefore a larger covariance matrix. Moreover,

the numerical experiments show that the method can be sensitive to the choice of the kernel and its tuning (although good practices can be derived). From a methodological viewpoint, further interpretation of the impact of the different kernels could be explored. Besides, extensions of kernel herding sampling for quantile estimation could be investigated, in a similar fashion as the work on randomized quasi-Monte Carlo for quantiles proposed by Kaplan et al. (2019). Kernel herding could also be used to quantize conditional distributions, using the so-called "conditional kernel mean embedding" concept reviewed by Klebanov et al. (2020). Finally, regarding the industrial use case, the next step should be to perform a reliability analysis by considering another group of random variables (related to the wind turbine) or to explore the possibilities offered by reliability-oriented sensitivity analysis in the context of kernel-based indices, as studied in Marrel and Chabridon (2021).

Chapter 2

Kernel-based surrogate models validation

This chapter is adapted from the following reference:

E. Fekhari, B. Iooss, J. Muré, L. Pronzato and M.J. Rendas (2023). "Model predictivity assessment: incremental test-set selection and accuracy evaluation". In: *Studies in Theoretical and Applied Statistics*, pages 315–347. Springer.

2.1 Introduction

The development of methods to validate and certify the predictivity of supervised learning models is essential to the industry. Estimating the predictivity of these models can either be done by cross-validation or using a suitably selected test sample (as introduced in Section **??**). Both in a given-data context (i.e., machine learning) or a simulated data context (i.e., computer experiment), guarantees on the validation procedure are increasingly asked. Certain risk-averse industries (e.g., nuclear) impose to establish these guarantees from independent test-sets, i.e., data set that has not been used either to train or to select the learning model (Borovicka et al., 2012; Xu and Goodacre, 2018; Iooss, 2021). Using the prediction residuals on this test set, an independent evaluation of the proposed leaning 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 present chapter introduces methods to choose a "good" test set, either within a given dataset or within the input space of the model, as recently motivated in Iooss (2021); Joseph and Vakayil (2022). The construction of test sets is studied as an uncertainty propagation of the learning model's error, on which an average error may be estimated using the Bayesian quadrature methods introduced in Chapter 1 for mean estimation.

A first choice concerns the size of the test set. No optimal choice exists, and, when only a finite dataset is available, classical machine learning (ML) handbooks (Hastie et al., 2009; Goodfellow et al., 2016) 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. This point is not formally addressed in the following (see Xu and Goodacre, 2018 for a numerical study of this issue). 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 (Hastie et al., 2009). 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 computationally expensive computer experiments on which a model will be identified (Fang et al., 2006; Pronzato and Müller, 2012).

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 Borovicka et al. (2012). For example, the CADEX and DUPLEX

algorithms (Kennard and Stone, 1969; Snee, 1977) can sequentially extract points from a database to include them in a test sample, using an inter-point distance criterion.

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 learned 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 chapter shows, test sets built in this manner do enable a good assessment of the quality of models learned 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 (Sheikholeslami and Razavi, 2017; Shang and Apley, 2020). Different empirical strategies have been proposed for incremental space-filling design (Iooss et al., 2010; Crombecq et al., 2011; Li et al., 2017), which basically entail the addition of new points in the zones poorly covered by the current design. Shang and Apley (2020) have recently proposed an improvement of the CADEX algorithm, called the "fully-sequential space-filling" (FSSF) design. Nogales Gómez et al. (2021) also proposed an improved version of such design enforcing boundary avoidance. Although they are developed for different purposes, nested space filling designs (Qian et al., 2009) and sliced space filling designs (Qian and Wu, 2009) can also be used to build sequential designs.

This work provides insights into these subjects in two main directions: *(i)* definition of a 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 Mak and Joseph (2018) and kernel herding Chen et al. (2010), the latter two algorithms being typically used to provide a representative sample of a desired theoretical or empirical distribution. Besides, this chapter presents a numerical benchmark analysis comparing the behavior of the three algorithms on a selected set of test cases and an industrial case.

This chapter is organized as follows. Section 2.2 defines the predictivity criterion considered and proposes different methods for its estimation. Section 2.3 presents the algorithms used for test-point selection. Our numerical results are presented in Sections 2.4 and 2.5: in Section 2.4 a test set is freely chosen within the entire input space, while in Section 2.5 an existing data set can be split into a training sample and a test set. Finally, Section 2.6 concludes and outlines some perspectives.

2.2 Predictivity assessment criteria for an ML model

In this section, a new criterion to assess the predictive performance of a model is proposed, enhancing a standard model quality metric by suitably weighting the errors observed on the test set. Let us 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. $\mathbf{x}' \in \mathcal{X}$). Considering the training sample denoted by $(\mathbf{X}_m, \mathbf{y}_m)$, with $\mathbf{y}_m = [y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(m)})]^\top$. The test sample is

denoted by $(\mathbf{X}_n, \mathbf{y}_n) = (\mathbf{x}^{(i)}, y(\mathbf{x}^{(i)}))_{1 \leq i \leq n}$. Remember that the intersection between these two samples is empty, $\mathbf{X}_m \cap \mathbf{X}_n = \varnothing$.

2.2.1 The predictivity coefficient

Let us denote by $\eta_m(\mathbf{x})$ the prediction at point $\mathbf{x}$ of a model learned using $(\mathbf{X}_m, \mathbf{y}_m)$ (Hastie et al., 2009; Rasmussen and Williams, 2006). A classical measure for assessing the predictive ability of η_m, in order to evaluate its validity, is the *predictivity coefficient*. Considering the probability measure μ 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 (i.e., ideal) value of the predictivity is defined as the following normalization of the Integrated Square Error (ISE):

$$Q^2_\mu(\eta_m) = 1 - \frac{\mathrm{ISE}_\mu(\eta_m)}{\mathrm{Var}_\mu(\eta_m)}, \tag{2.1}$$

where

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

The ideal predictivity $Q^2_{\mathrm{ideal}}(\mu)$ is usually estimated by its empirical version calculated over the test sample $(\mathbf{X}_n, \mathbf{y}_n)$, see (Da Veiga et al., 2021, p. 32):

$$\widehat{Q}^2_n = 1 - \frac{\sum_{i=1}^n \left[y(\mathbf{x}^{(i)}) - \eta_m(\mathbf{x}^{(i)}) \right]^2}{\sum_{i=1}^n \left[y(\mathbf{x}^{(i)}) - \overline{y}_n \right]^2}, \tag{2.2}$$

where $\overline{y}_n = (1/n) \sum_{i=1}^n y(\mathbf{x}^{(i)})$ 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 $\eta_m(\cdot)$. This estimator $\widehat{Q}^2_n$ is the *coefficient of determination* (also called "Nash-Sutcliffe criterion" Nash and Sutcliffe, 1970), which is a standard notion in regression studies (Kleijnen and Sargent, 2000; Iooss et al., 2010). 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 $\widehat{Q}^2_n$.

2.2.2 Weighting the test sample

The simplest way to estimate the $\mathrm{ISE}_\mu(\mathbf{X}_m, \mathbf{y}_m)$ (present on the numerator of the predictivity coefficient) is by computing the arithmetic mean of the squared residuals evaluated on the test set $\mathbf{X}_n$. Writing the equivalent discrete measure $\xi_n = \frac{1}{n}\sum_{i=1}^n \delta_{\mathbf{x}^{(i)}}$, with $\delta_{\mathbf{x}}$ the Dirac measure at $\mathbf{x}$, this estimate can be expressed as:

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

When the points $\mathbf{x}^{(i)}$ of the test set $\mathbf{X}_n$ are distant from the points of the training set $\mathbf{X}_m$, the squared prediction errors $|y(\mathbf{x}^{(i)}) - \eta_m(\mathbf{x}^{(i)})|^2$ tend to represent the worst possible error situations, and $\mathrm{ISE}_{\xi_n}(\eta_m)$ tends to overestimate $\mathrm{ISE}_\mu(\eta_m)$. 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.

Let us assume that the prediction error $\delta_m(\mathbf{x}) = y(\mathbf{x}) - \eta_m(\mathbf{x})$ is a realization of a Gaussian Process (GP) with mean $\widehat{\delta}_m(\mathbf{x})$ and covariance kernel $\sigma^2 K_{|m}$, $\delta_m(\mathbf{x}) \sim \mathrm{GP}(\widehat{\delta}_m(\mathbf{x}), \sigma^2 K_{|m})$

$$\left\{\begin{array}{l} \widehat{\delta}_m(\mathbf{x}) = \mathbf{k}_m^\top(\mathbf{x})\mathbf{K}_m^{-1}(\mathbf{y}_m - \boldsymbol{\eta}_m), \\ \sigma^2 K_{|m}(\mathbf{x},\mathbf{x}') = \mathbb{E}[\delta_m(\mathbf{x})\delta_m(\mathbf{x}')] = \sigma^2\left[K(\mathbf{x},\mathbf{x}') - \mathbf{k}_m^\top(\mathbf{x})\mathbf{K}_m^{-1}\mathbf{k}_m(\mathbf{x}')\right] . \end{array}\right. \tag{2.3}$$

Where $\boldsymbol{\eta}_m = [\eta_m(\mathbf{x}^{(1)}), \ldots, \eta_m(\mathbf{x}^{(m)})]^\top$, $\mathbf{k}_m(\mathbf{x})$ is the column vector $[K(\mathbf{x},\mathbf{x}^{(1)}), \ldots, K(\mathbf{x},\mathbf{x}^{(m)})]^\top$, and $\mathbf{K}_m$ is the $m \times m$ covariance 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. Note that in the case of a learning model η_m which interpolates the observations $\mathbf{y}_m$, the errors observed at the learning points $\mathbf{X}_m$ equal zero, leading finally to the posterior $\mathrm{GP}(0, \sigma^2 K_{|m})$ for $\delta_m(\mathbf{x})$.

The prediction model error above allows us to study how well $\mathrm{ISE}_\mu(\eta_m)$ is estimated using a test set $\mathbf{X}_n$. The expected squared error when estimating $\mathrm{ISE}_\mu(\eta_m)$ by $\mathrm{ISE}_{\xi_n}(\eta_m)$, is defined as $\overline{\Delta}^2(\xi_n, \mu; \eta_m)$:

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

Tonelli's theorem gives

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

Since $\mathbb{E}\left[U^2V^2\right] = 2\left(\mathbb{E}[UV]\right)^2 + \mathbb{E}\left[U^2\right]\mathbb{E}\left[V^2\right]$ for any one-dimensional normal centered random variables U and V. The expression can then be written as:

$$\overline{\Delta}^2(\xi_n, \mu; \eta_m) = \int_{\mathcal{X}^2} 2\,\mathbb{E}[\delta_m(\mathbf{x})\delta_m(\mathbf{x}')]^2 + \mathbb{E}[\delta_m^2(\mathbf{x})]\mathbb{E}[\delta_m^2(\mathbf{x}')]\,\mathrm{d}(\xi_n - \mu)(\mathbf{x})\mathrm{d}(\xi_n - \mu)(\mathbf{x}') \tag{2.4a}$$

$$\overline{\Delta}^2(\xi_n, \mu; \eta_m) = \int_{\mathcal{X}^2} \overline{K}_{|m}(\mathbf{x}, \mathbf{x}')\,\mathrm{d}(\xi_n - \mu)(\mathbf{x})\mathrm{d}(\xi_n - \mu)(\mathbf{x}') \tag{2.4b}$$

$$\overline{\Delta}^2(\xi_n, \mu; \eta_m) = \sigma^2\,\mathrm{MMD}^2_{\overline{K}_{|m}}(\xi_n, \mu). \tag{2.4c}$$

Interestingly, the last expression is equivalent to the maximum mean discrepancy (previously introduced in this manuscipt and further defined in Appendix **??**) between μ and ξ_n for a kernel $\overline{K}_{|m}(\cdot,\cdot)$. Note that σ^2 only appears as a multiplying factor in Eq. (2.4b), with the consequence that σ^2 does not impact the choice of a suitable ξ_n. The resulting kernel $\overline{K}_{|m}(\cdot,\cdot)$ is differently defined whether *(i)* the learning model $\eta_m(\mathbf{x})$ interpolates $\mathbf{y}_m$ or not *(ii)*:

$$\begin{cases} (i) \Rightarrow \overline{K}_{|m}(\mathbf{x}, \mathbf{x}') & := 2\,K^2_{|m}(\mathbf{x}, \mathbf{x}') + K_{|m}(\mathbf{x}, \mathbf{x})K_{|m}(\mathbf{x}', \mathbf{x}') \\ (ii) \Rightarrow \overline{K}_{|m}(\mathbf{x}, \mathbf{x}') & := 2\left[K_{|m}(\mathbf{x}, \mathbf{x}') + 2\,\widehat{\delta}_m(\mathbf{x})\widehat{\delta}_m(\mathbf{x}')\right]K_{|m}(\mathbf{x}, \mathbf{x}') \\ & \quad + \left[\widehat{\delta}^2_m(\mathbf{x}) + K_{|m}(\mathbf{x}, \mathbf{x})\right]\left[\widehat{\delta}^2_m(\mathbf{x}') + K_{|m}(\mathbf{x}', \mathbf{x}')\right]. \end{cases} \tag{2.5}$$

The main idea is to replace ξ_n, uniform on $\mathbf{X}_n$, by a nonuniform measure ζ_n supported on $\mathbf{X}_n$, $\zeta_n = \sum_{i=1}^n w_i \delta_{\mathbf{x}^{(i)}}$ with weights $\mathbf{w}_n = (w_1, \ldots, w_n)^\top$ chosen such that the estimation error $\overline{\Delta}^2(\xi_n, \mu; \eta_m)$, and thus $d^2_{\overline{K}_{|m}}(\zeta_n, \mu)$, is minimized. The squared MMD for the kernel $\overline{K}_{|m}$ between μ and the weighted measure ζ_n can be expressed as:

$$\mathrm{MMD}^2_{\overline{K}_{|m}}(\zeta_n, \mu) = \varepsilon_{\overline{K}_{|m},\mu} - 2\,\mathbf{w}_n^\top P_{\overline{K}_{|m},\mu}(\mathbf{X}_n) + \mathbf{w}_n^\top \overline{\mathbf{K}}_{|m}(\mathbf{X}_n)\mathbf{w}_n\,, \tag{2.6}$$

where $P_{\overline{K}_{|m},\mu}(\mathbf{X}_n)$ is the vector of potentials $\left[\int_{\mathcal{X}} \overline{K}_{|m}(\mathbf{x}, \mathbf{x}^{(1)})\mathrm{d}\pi(\mathbf{x}), \ldots, \int_{\mathcal{X}} \overline{K}_{|m}(\mathbf{x}, \mathbf{x}^{(n)})\mathrm{d}\pi(\mathbf{x})\right]^\top$, and $\overline{\mathbf{K}}_{|m}(\mathbf{X}_n)$ is the $n \times n$ covariance matrix such that $\{\overline{\mathbf{K}}_{|m}(\mathbf{X}_n)\}_{i,j} = \overline{K}_{|m}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, $\forall i, j = 1, \ldots, n$, and $\varepsilon_{\overline{K}_{|m},\mu} = \int_{\mathcal{X}^2} \overline{K}_{|m}(\mathbf{x}, \mathbf{x}')\,\mathrm{d}\mu(\mathbf{x})\mathrm{d}\mu(\mathbf{x}')$. The squared MMD defined in Eq. (2.6) is minimized for the following optimal weights $\mathbf{w}_n^*$:

$$\mathbf{w}_n^* = \overline{\mathbf{K}}^{-1}_{|m}(\mathbf{X}_n)\mathbf{p}_{\overline{K}_{|m},\mu}(\mathbf{X}_n)\,. \tag{2.7}$$

Therefore, an optimally weighted estimator of the predictivity coefficient supported on the test set $\mathbf{X}_n$ is defined as:

$$\widehat{Q}^2_{n*} = 1 - \frac{\sum_{i=1}^n w_i^* \left[y(\mathbf{x}^{(i)}) - \eta_m(\mathbf{x}^{(i)})\right]^2}{\frac{1}{n}\sum_{i=1}^n \left[y(\mathbf{x}^{(i)}) - \overline{y}_n\right]^2}, \tag{2.8}$$

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. Moreover, this approach does no constraint the weights, which

is works best in our experience than the different constrained versions (e.g., non-negativity, summing to one) studied in Pronzato and Rendas (2021b).

Remark 5. The optimal estimator $\widehat{Q}^2_{n*}$ focused on weighting numerator of the coefficient of determination defined in [ccc]. However, the variance's estimator on the denominator can also be optimally weighted. Let us write an alternative version of $\widehat{Q}^2_{n*}$

$$\widehat{Q}'^2_n = 1 - \frac{\sum_{i=1}^n \left[y(\mathbf{x}^{(i)}) - \eta_m(\mathbf{x}^{(i)})\right]^2}{\sum_{i=1}^n \left[y(\mathbf{x}^{(i)}) - \overline{y}_m\right]^2}, \tag{2.9}$$

which compares the performance on the test set of η_m and $\overline{y}_m = (1/m)\sum_{i=1}^m y(\mathbf{x}^i)$. Using similar developments as previously, is possible to also apply a weighting procedure to the denominator of $\widehat{Q}'^2_n$, in order to make it resemble its integral version $V'_\mu(\mathbf{y}_m) = \int_{\mathcal{X}} \left[y(\mathbf{x}) - \overline{y}_m\right]^2 \mathrm{d}\mu(\mathbf{x})$ (see Fekhari et al., 2023).

2.3 Test-set construction

In the previous section the test set was assumed as given, and a method was proposed to estimate the $\mathrm{ISE}_\mu(\eta_m)$ (with the learning model η_m, built on $\mathbf{X}_m$) by an optimally weighted sum of the residuals. The objective in this section is to construct a test set of size n, denoted by $\mathbf{X}_n = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\right\} \subset \mathcal{X}$. To evaluate the predictivity of a model learned on a training set $\mathbf{X}_m$, a strategy is to place the test points the furthest away from the training set, to obtain a space-filling design when gathering leaning and test set. The sampling methods used for building the test set should then be space-filling and incremental. The advantage of using an iterative construction is that it can be stopped as soon as the predictivity estimation 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. This section introduces different space-filling methods, later compared for test set construction.

2.3.1 Fully-Sequential Space-Filling design

The Fully-Sequential Space-Filling forward-reflected (FSSF-fr) algorithm Shang and Apley (2020) relies on the CADEX algorithm Kennard and Stone (1969) (also called the "coffee-house" method Müller (2007)). 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 $\mathcal{X} = [0,1]^d$, Shang and Apley (2020) recommends taking $\mathcal{S}$ 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 $\mathbf{X}_i = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(i)}\}$, select

$$\mathbf{x}^{(i+1)} \in \underset{\mathbf{x} \in \mathcal{S} \setminus \mathbf{X}_i}{\arg\max} \left[\min \left(\min_{j \in \{1,\dots,i\}} \|\mathbf{x} - \mathbf{x}^{(j)}\|, \sqrt{2}\, d\, \mathrm{dist}(\mathbf{x}, R(\mathbf{x})) \right) \right], \tag{2.10}$$

   where $R(\mathbf{x})$ is the symetric 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 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. While the standard coffee-house (greedy packing) algorithm simply uses $\mathbf{x}^{(i+1)} \in \arg\max_{\mathbf{x} \in \mathcal{S} \setminus \mathbf{X}_i} \min_{j \in \{1,\dots,i\}} \|\mathbf{x} - \mathbf{x}^{(j)}\|$. The factor $\sqrt{2}d$ in Eq. (2.10) proposed in Shang and Apley (2020) 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 Nogales Gómez et al. (2021).

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 the distribution μ of the model inputs is frequently not uniform. It is then desirable to select a test set representative of μ. Which can be achieved through the inverse transform of the CDF: 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 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, but is more complicated in other cases, see (Lemaire et al., 2009, 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.

2.3.2 Support points

Support points Mak and Joseph (2018) 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 Székely and Rizzo (2013),

$$K_E(\mathbf{x}, \mathbf{x}') = \frac{1}{2} \left(\|\mathbf{x}\| + \|\mathbf{x}'\| - \|\mathbf{x} - \mathbf{x}'\| \right). \tag{2.11}$$

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

$$\mathrm{MMD}^2_{K_E}(\xi_n, \mu) = \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}\|\mathbf{x}^{(i)} - \zeta\| - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\| - \mathbb{E}\|\zeta - \zeta'\|\,, \tag{2.12}$$

where ζ and ζ' are independently distributed with μ; see Sejdinovic et al. (2013). A key property of the energy-distance kernel is that it is characteristic Sriperumbudur et al. (2010): 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 $\mathbf{X}^s_n$ are then given by

$$\mathbf{X}^s_n \in \underset{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}}{\arg\min}\left(\frac{2}{nN}\sum_{i=1}^{n}\sum_{k=1}^{N}\|\mathbf{x}^{(i)} - \mathbf{x}'^{(k)}\| - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|\right). \tag{2.13}$$

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 Mak and Joseph (2018), 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.

The method was used to split a dataset into a training set and a test set in Joseph and Vakayil (2022), where the N points $\mathbf{X}_N$ in Eq. (2.13) are those from the dataset. Then $\mathbf{X}^s_n$ gives the test set and the other $N - n$ points are used for training. There is a serious additional difficulty though, as choosing $\mathbf{X}^s_n$ 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 $\mathbf{X}^s_n$ that corresponds to the closest points in $\mathbf{X}_N$ (for the Euclidean distance) to the continuous solution obtained Joseph and Vakayil (2022).

The direct determination of support points through Eq. (2.13) does not allow the construction of a nested sequence of test sets. One possibility would be to solve Eq. (2.13) 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 Eq. (2.12) for the candidate set $\mathcal{S} = \mathbf{X}_N$: at iteration i, the algorithm chooses

$$\mathbf{x}^s_{i+1} \in \underset{\mathbf{x}\in\mathcal{S}}{\arg\min}\left(\frac{1}{N}\sum_{k=1}^{N}\|\mathbf{x} - \mathbf{x}'^{(k)}\| - \frac{1}{i+1}\sum_{j=1}^{i}\|\mathbf{x} - \mathbf{x}^{(j)}\|\right). \tag{2.14}$$

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 2.5). Note that Joseph and Vakayil (2022) only studies the split of a given data set into a learning and test set while this chapter builds support points on the input space $\mathcal{X}$

Greedy MMD minimization can be applied to other kernels than the distance kernel Eq. (2.11), see Teymur et al. (2021); Pronzato (2022). In the next section the closely related method of kernel herding is recalled (KH) (Chen et al., 2010), after its presentation in Chapter 1 of the present manuscript.

2.3.3 Kernel herding

As introduced in Section 2.3.3, Lacoste-Julien et al. (2015) proposed a linearization of the MMD minimization using the Frank-Wolfe algorithm. Let us define a positive definite kernel K on $\mathcal{X}\times\mathcal{X}$, and consider $\xi_i = (1/i)\sum_{j=1}^{i}\delta_{\mathbf{x}^{(i)}}$ as the empirical measure for $\mathbf{X}_i$. In the sequential and uniformly weighted case, this iteration i of kernel herding is expressed as a difference of potentials:

$$\mathbf{x}_{i+1} \quad \in \quad \underset{\mathbf{x}\in\mathcal{S}}{\arg\min}\left[P_{\xi_i}(\mathbf{x}) - P_{\mu}(\mathbf{x})\right], \tag{2.15}$$

with $\mathcal{S}\subseteq\mathcal{X}$ a given candidate set and $P_{\xi_i}(\mathbf{x}) = (1/i)\sum_{j=1}^{i}K(\mathbf{x},\mathbf{x}^{(j)})$.

Once the targeted measure substituted by an empirical measure μ_N based on a sample $(\mathbf{x}'^{(k)})_{k=1\dots N}$ then complete estimation becomes: $P_{\mu_N}(\mathbf{x}) = (1/N)\sum_{k=1}^{N}K(\mathbf{x},\mathbf{x}'^{(k)})$, which gives

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

When K is the energy-distance kernel Eq. (2.11) the greedy support points from Eq. (2.14) are recovered with a factor $1/i$ instead of $1/(i+1)$ in the second sum.

The candidate set $\mathcal{S}$ in Eq. (2.15) is arbitrary and can be chosen as in Section 2.3.1. A neat advantage of kernel herding over support points is that the potential $P_\mu(\mathbf{x})$ 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 $P_{\mu_{[i]}}(x)$ is known explicitly for each $i\in\{1,\dots,d\}$. Indeed, for $\mathbf{x}=(x_1,\dots,x_d)\in\mathcal{X}$, $P_\mu(\mathbf{x}) = \prod_{i=1}^{d}P_{\mu_{[i]}}(x_i)$ (see Pronzato and Zhigljavsky, 2020). 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

$$K_{5/2,\theta}(x-x') = \left(1+\frac{\sqrt{5}}{\theta}|x-x'|+\frac{5}{3\theta^2}(x-x')^2\right)\exp\left(-\frac{\sqrt{5}}{\theta}|x-x'|\right), \tag{2.16}$$

the one-dimensional potentials are given in Appendix **??** 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. The values of the

correlation lengths empirically show a significant 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 Pronzato and Zhigljavsky, 2020).

2.3.4 Numerical illustration

As first numerical illustration, the FSSF-fr (denoted FSSF in the following), support points and kernel herding algorithms were applied in 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 2.1 and 2.2 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 (Santner et al., 2003). When μ is uniform, the initial design is a maximin Latin hypercube design (introduced in Section **??**) 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 Fig. 2.1 and Fig. 2.2 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. However, 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. In the next section, these designs are used for estimating the quality of the predictivity metric.

2.4 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 $f(\mathbf{x})$ has an easy to evaluate analytical expression, see Section 2.4.1. This allows a good estimation of Q_μ^2 (see Eq. (2.1)) by a large Monte Carlo sample (with size $M = 10^6$), which will serve as reference when assessing the performance of each of the other estimators.

The validation designs are built by FSSF, support points and kernel herding, presented in Sections 2.3.1, 2.3.2, and 2.3.3, and the performances obtained are compared for each one, considering the uniform and the weighted estimator of Section 2.2.2.

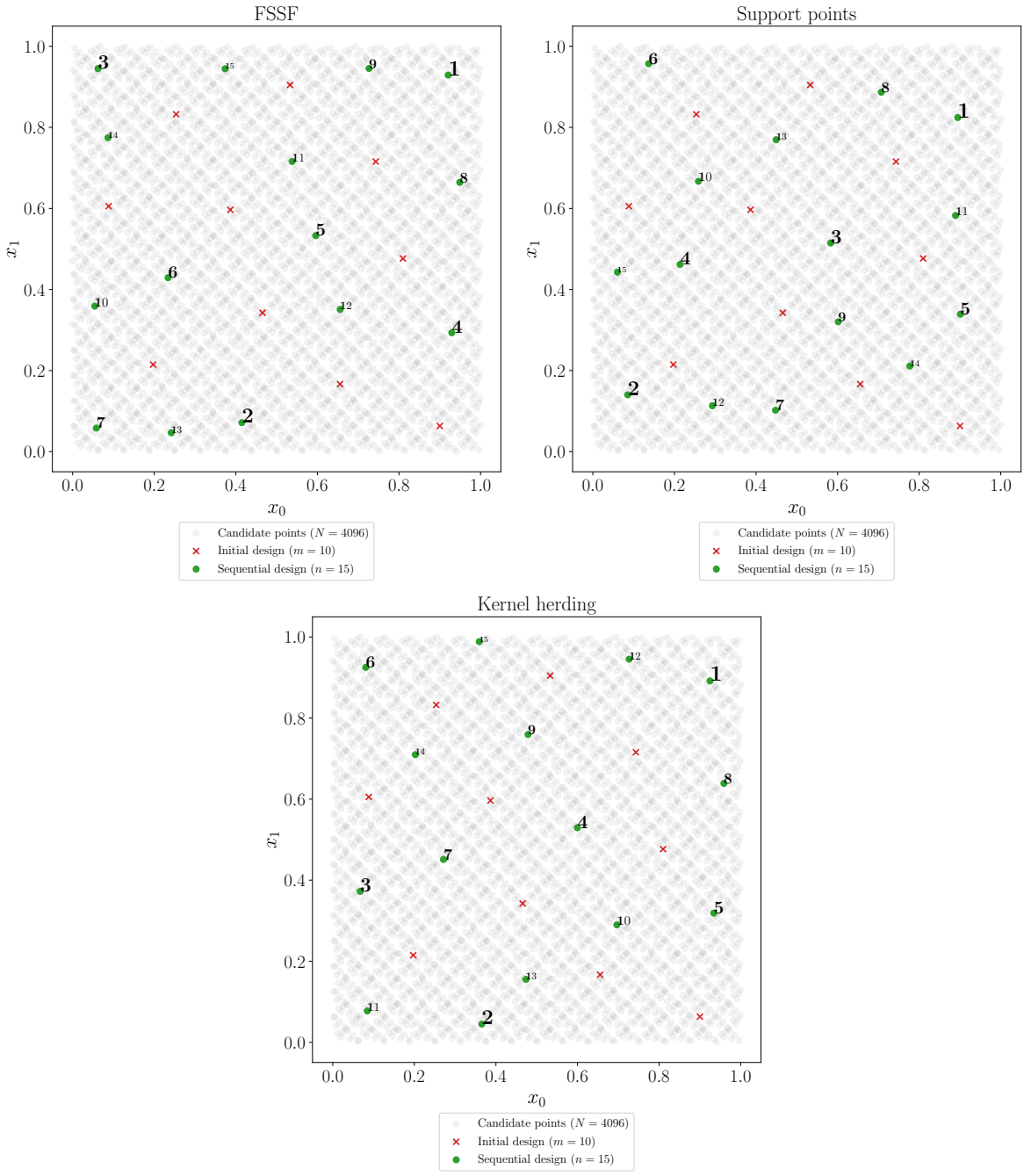

Figure 2.1 Additional points (ordered, green) complementing an initial design (red crosses), μ is uniform on $[0, 1]$, the candidate points are in gray.

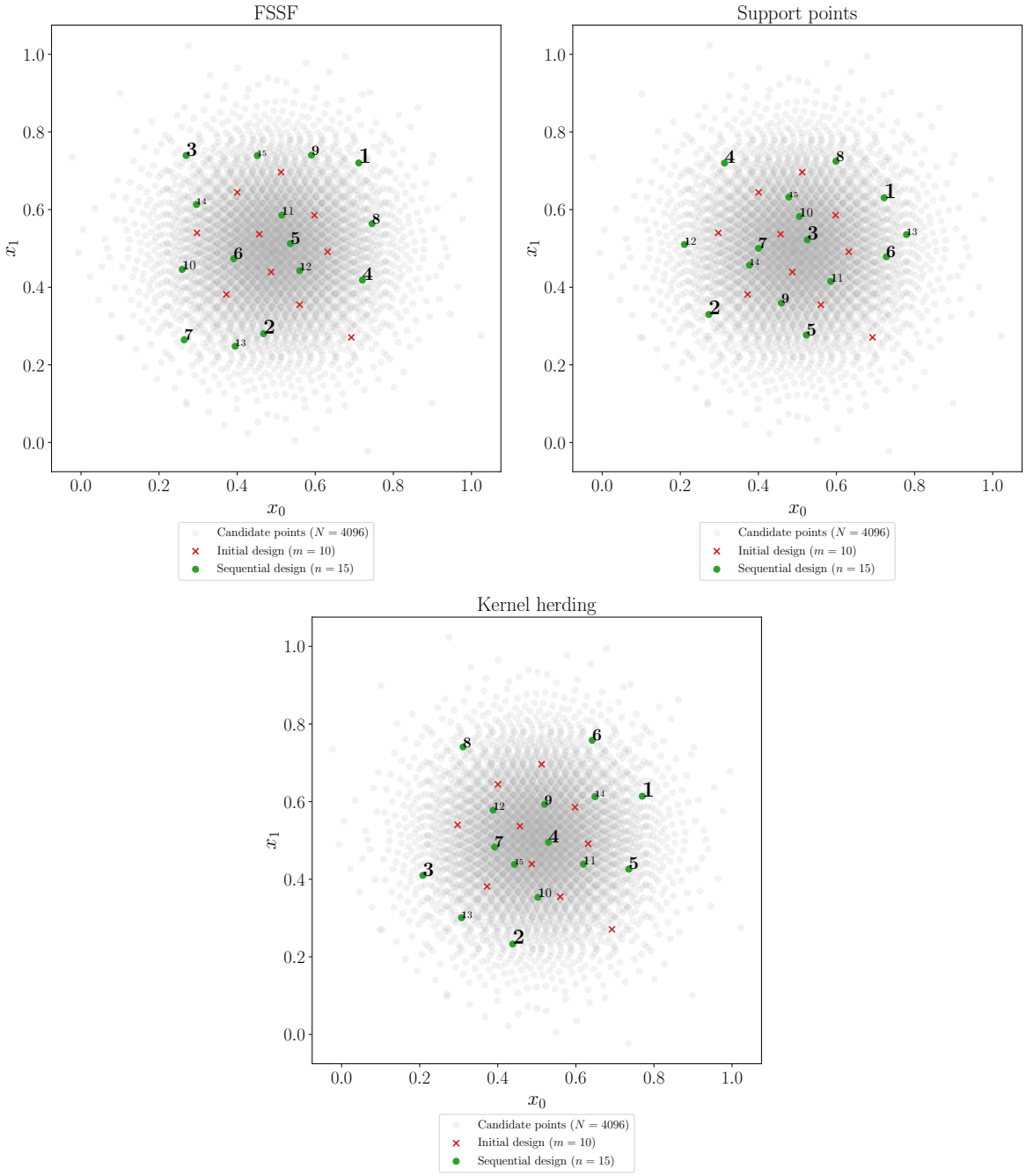

Figure 2.2 Additional points (ordered, green) complementing an initial design (red crosses), μ normal, the candidate points are in gray.
[Should we keep only two options in vertical or move the legends to the side?]

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 2.3.4. 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 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 S_N must be transformed as explained in section 2.3.1. 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 2.3.2).

For each test-case, a GP regression model is fitted to the m observations using ordinary Kriging Rasmussen and Williams (2006) (a GP model with constant mean), with an anisotropic Matérn kernel with regularity parameter $5/2$, 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 Baudin et al. (2017). The kernel used for kernel herding is different and corresponds to the tensor product of one-dimensional Matérn kernels Eq. (2.16), so that the potentials $P_\mu(\cdot)$ are known explicitly (see Appendix **??**); 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 $Q^2 > 0.9$, for each test-case, three different sizes m of the training set are selected 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 2.3, showing that the function is rather smooth, even if its behavior 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: $m \in \{5, 15, 30\}$.

Test-case 2. The second test function, plotted in the right panel of Figure 2.3 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 $m \in \{8, 15, 30\}$.

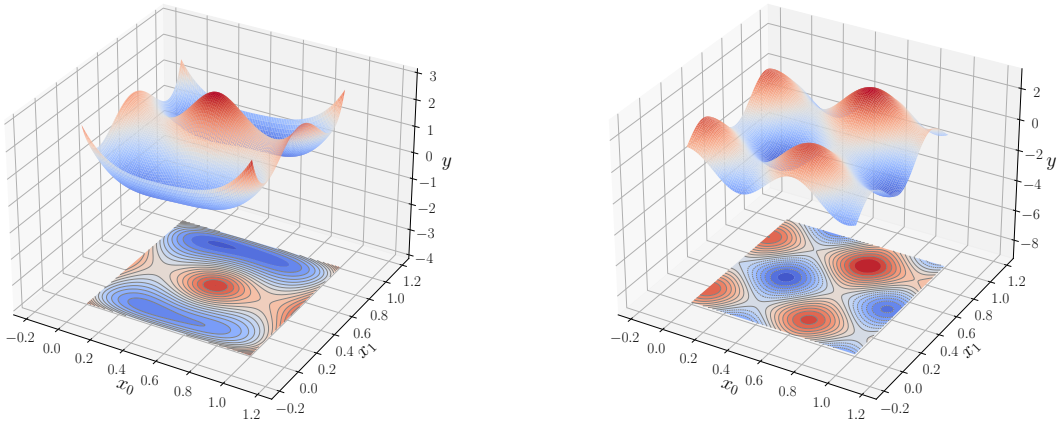

Figure 2.3 Left: $f_1(\mathbf{x})$ (test-case 1); right: $f_2(\mathbf{x})$ (test-case 2); $\mathbf{x} \in \mathcal{X} = [0,1]^2$.

Test-case 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: $m \in \{15, 30, 100\}$.

2.4.2 Results and analysis

The numerical results obtained in this section are presented in Figures 2.4, 2.5, and 2.6. 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). Assuming 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 ($n \in \{4, \ldots, 50\}$), the observed values of $\widehat{Q}^2_n$ (equation Eq. (2.2)) and Q^2_{n*} (equation Eq. (2.8)), 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 the complete model-fitting procedure is repeated 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 Dubrule (1983) cannot be used, making the computations rather intensive. The three figures show, and as expected, that the Q^2_{LOO} tends to under-estimate 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 2.4 where $Q^2_{LOO} < 0.3$ for $m = 5$ and 15.

Let us first concentrate on the non-weighted estimators (solid curves). 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 behavior, 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 2.1, and thus residuals at these points are themselves already better representatives of the local average errors.

Let us 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 2.4 and FSSF (green curve) in Figure 2.5. 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 2.1 – 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 behavior across test-cases. Setting aside the strikingly singular situation of test-case 2, for which kernel herding displays a pathological (bad) behavior for the "very good" model, and all methods present an overall good behavior, the details of the

tested function do not seem to play an important role concerning the relative merits of the estimators and validation designs.

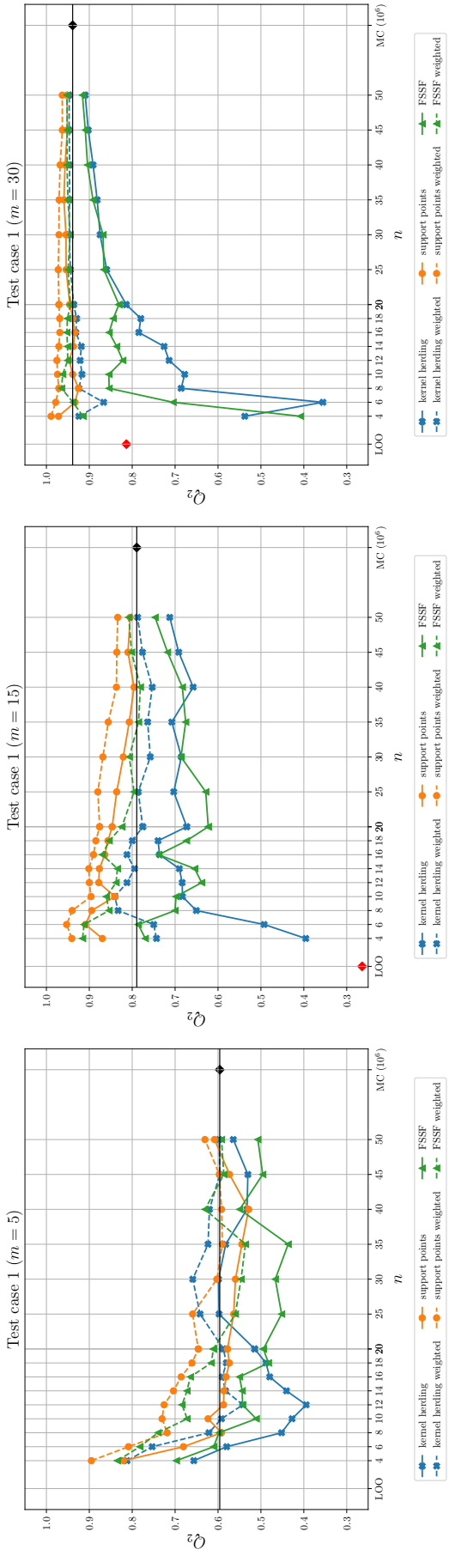

Figure 2.4 Test-case 1: predictivity assessment of a poor (left), good (center) and very good (right) model with kernel herding, support points and FSSF test sets.

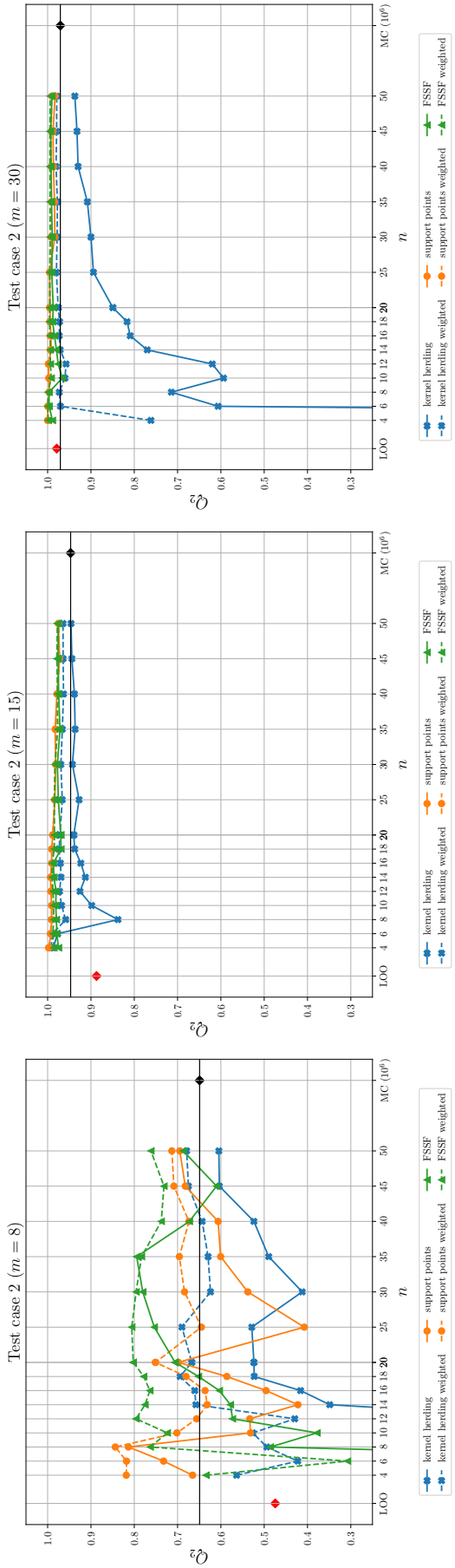

Figure 2.5 Test-case 2: predictivity assessment of a poor (left), good (center) and very good (right) model with kernel herding, support points and FSSF test sets.

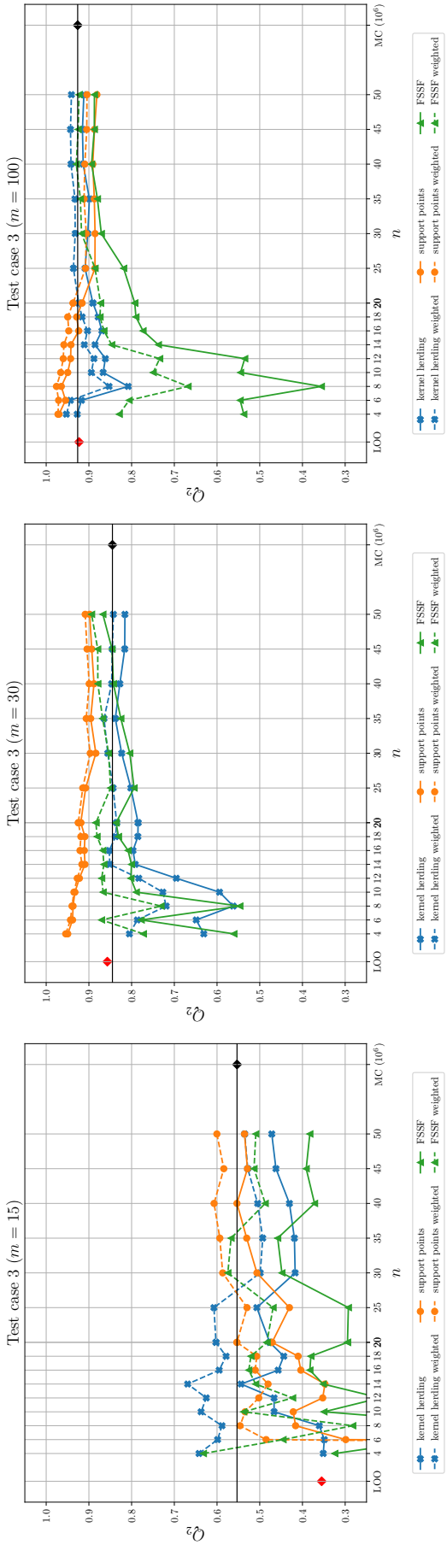

Figure 2.6 Test-case 3: predictivity assessment of a poor (left), good (center) and very good (right) model with kernel herding, support points and FSSF test sets.

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

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

This section illustrates the performance of the different designs and estimators considered in this chapter 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 Joseph and Vakayil (2022), 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 favor 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. Random Cross-Validation (RCV) is also considered, 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 $R = 1\,000$ designs were randomly selected among them. A model is fitted on the m complementary points ($m = N-n$), which yields an empirical distribution of Q^2 values for each ratio n/N considered.

2.5.1 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 Geffraye et al. (2011). The studied scenario simulates a hypothetical large-break loss of primary coolant accident for which the output of interest is the peak cladding temperature de Crécy et al. (2008); Iooss et al. (2010). 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, cor-

responding 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 Fig. 2.7 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 Iooss et al. (2010) for 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 Da Veiga et al. (2021) 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 **??**. 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 **??**. 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 2.7, 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 **?**, 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.

2.5.2 Benchmark results and analysis

Figure 2.8 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 **??**. 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

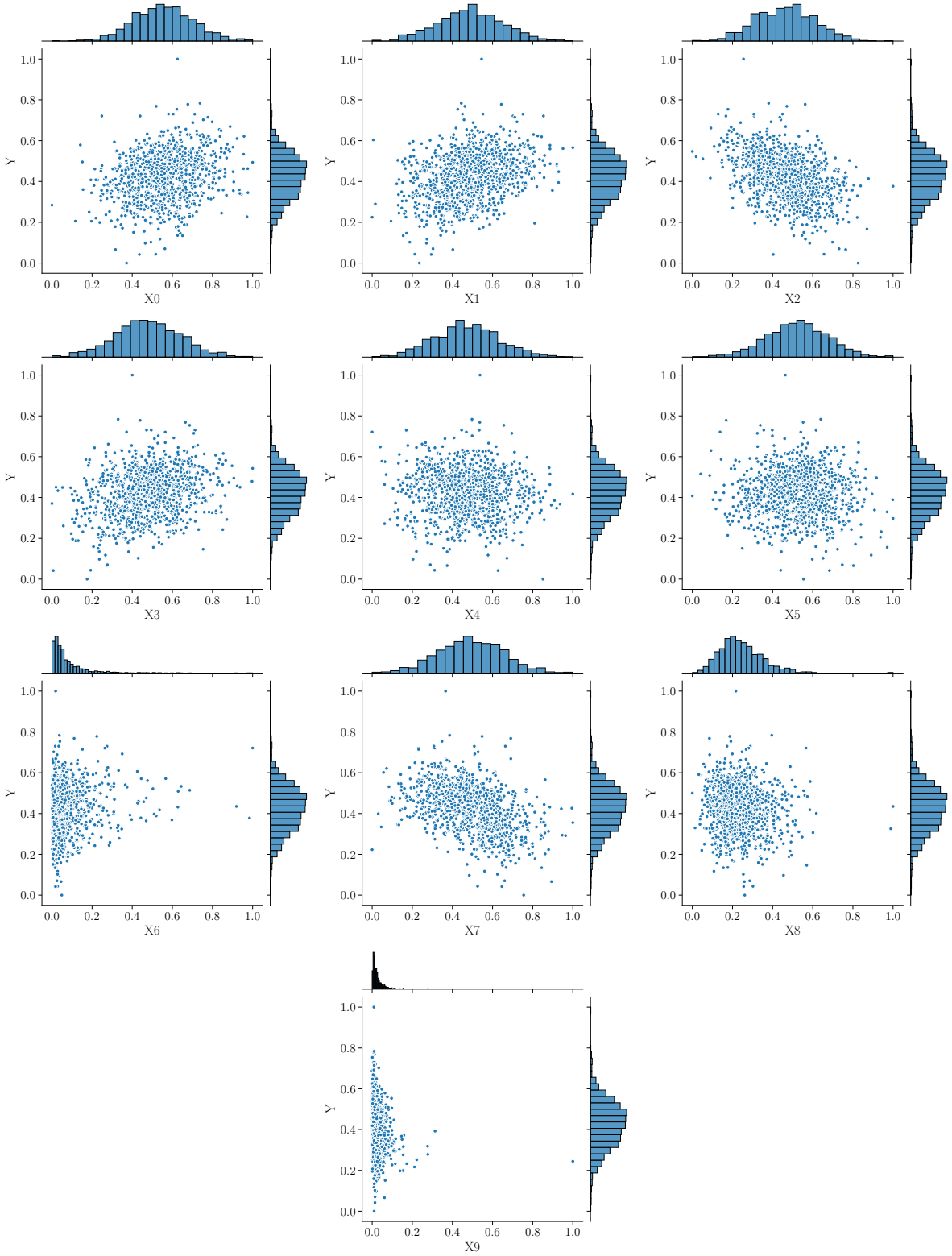

Figure 2.7 Test-case CATHARE: inputs output scatter plots ($N = 10^3$)

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 2.8 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 **??**, $\widehat{Q}^2_n$ (unweighted) 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 2.2.2, shown in dashed lines in Figure 2.8. 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 Eq. (2.7) is shown in Figure 2.9 (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)

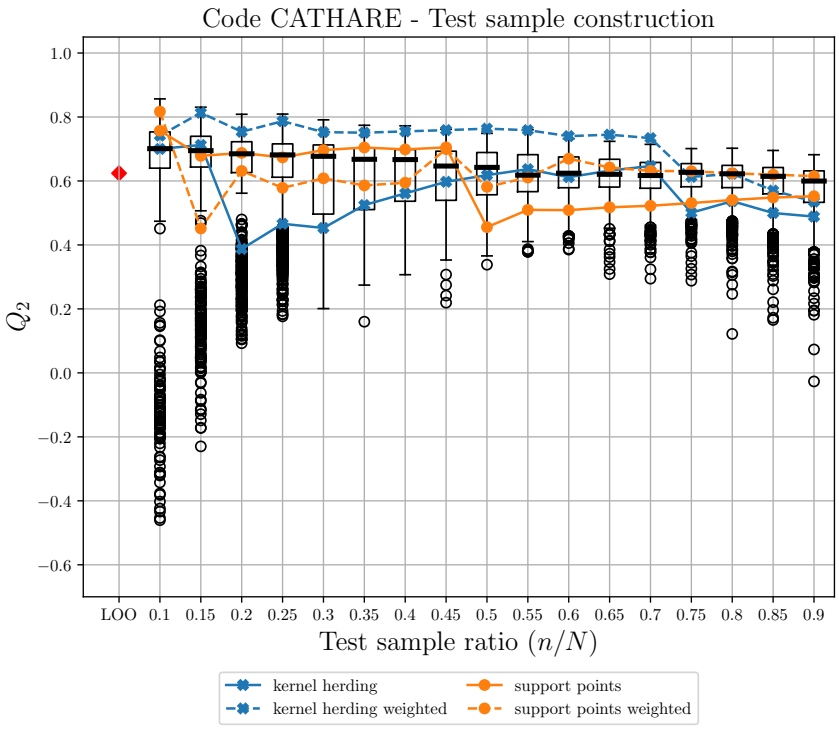

Figure 2.8 Test-case CATHARE: estimated Q^2. The box-plots are for random cross-validation, the red diamond (left) is for Q^2_{LOO}.

but is consistent with the behavior of Q^2_{n*} on Figure 2.8. Note that the energy-distance kernel Eq. (2.11) used for support points cannot be used for the weighting method of Section 2.2.2 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 behaviors 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.

2.6 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 chapter explored approaches based on support

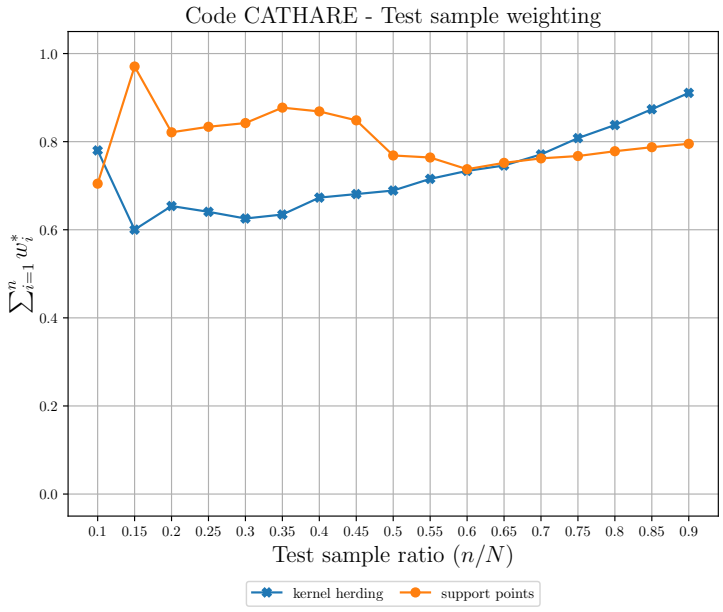

Figure 2.9 Test-case CATHARE: sum of the weights Eq. (2.7).

points, kernel herding and FSSF, considering the incremental construction of a test set (*i*) either as a particular space-filling design problem, where 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 Pronzato and Rendas (2021b)) 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 **??**), 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 **??**, 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 of 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_μ, 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 Mak and Joseph (2018), or MMD minimization in general **?**.

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 2.2.2 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 **??**, 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.

Part III:

Contributions to rare event estimation

La résignation est un suicide quotidien.

H. Balzac

Bibliography

Abdallah, I., Lataniotis, C., and Sudret, B. (2019). Parametric hierarchical kriging for multi-fidelity aero-servo-elastic simulators – Application to extreme loads on wind turbines. *Probabilistic Engineering Mechanics*, 55:67 – 77.

Ajenjo, A. (2023). *Info-gap robustness assessment of reliability evaluations for the safety of critical industrialsystems.* PhD thesis, Université Bourgogne Franche-Comté.

Au, S.-K. and Beck, J. L. (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277.

Bai, H., Shi, L., Aoues, Y., Huang, C., and Lemosse, D. (2023). Estimation of probability distribution of long-term fatigue damage on wind turbine tower using residual neural network. *Mechanical Systems and Signal Processing*, 190:110101.

Baudin, M., Dutfoy, A., Iooss, B., and Popelin, A. (2017). Open TURNS: An industrial software for uncertainty quantification in simulation. In Ghanem, R., Higdon, D., and Owhadi, H., editors, *Springer Handbook on Uncertainty Quantification*, pages 2001–2038. Springer.

Beauregard, E., Bérille, E., Berrabah, N., Berthelot, M., Burrows, J., Capaldo, M., Cornet, S., Costan, V., Duchet, M., Dufossé, E., Dupont, E., Franchet, M., Gouze, E., Grau, A., Joly, A., Kell, N., de Laleu, V., Latraube, F., Lovera, A., de Bazelaire, A., Monnot, E., Nogaro, G., Pagot, J., Pérony, R., Peyrard, C., Piguet, C., Régnier, A., Santibanez, E., Senn, C., Smith, C., Soriano, F., Stephan, P., Terte, N., Veyan, P., Vizireanu, D., and Yeow, L. (2022). *L'éolien en mer : un défi pour la transition énergétique.* Lavoisier.

Borovicka, T., Jirina, M. J., Kordik, P., and Jirina, M. (2012). Selecting representative data sets. In Karahoca, A., editor, *Advances in data mining, knowledge discovery and applications*, pages 43–70. INTECH.

Briol, F., Oates, C., Girolami, M., Osborne, M., and Sejdinovic, D. (2019). Probabilistic Integration: A Role in Statistical Computation? *Statistical Science*, 34:1 – 22.

Briol, F.-X., Oates, C., Girolami, M., and Osborne, M. (2015). Frank-Wolfe Bayesian Quadrature: Probabilistic Integration with Theoretical Guarantees. In *Advances in Neural Information Processing Systems*.

Chabridon, V. (2018). *Reliability-oriented sensitivity analysis under probabilistic model uncertainty–Application to aerospace systems.* PhD thesis, Université Clermont Auvergne.

Chen, T., Wang, X., Yuan, G., and Liu, J. (2018). Fatigue bending test on grouted connections for monopile offshore wind turbines. *Marine Structures*, 60:52–71.

Chen, Y., Welling, M., and Smola, A. (2010). Super-samples from kernel herding. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pages 109 – 116. AUAI Press.

Cousin, A. (2021). *Optimisation sous contraintes probabilistes d'un système complexe : Application au dimensionnement d'une éolienne offshore flottante*. PhD thesis, Institut Polytechnique de Paris.

Crombecq, K., Laermans, E., and Dhaene, T. (2011). Efficient space-filling and non-collapsing sequential design strategies for simulation-based modelling. *European Journal of Operational Research*, 214:683–696.

Da Veiga, S., Gamboa, F., Iooss, B., and Prieur, C. (2021). *Basics and Trends in Sensitivity Analysis: Theory and Practice in R*. Society for Industrial and Applied Mathematics.

de Crécy, A., Bazin, P., Glaeser, H., Skorek, T., Joufcla, J., Probst, P., Fujioka, K., Chung, B., Oh, D., Kyncl, M., Pernica, R., Macek, J., Meca, R., Macian, R., D'Auria, F., Petruzzi, A., Batet, L., Perez, M., and Reventos, F. (2008). Uncertainty and sensitivity analysis of the LOFT L2-5 test: Results of the BEMUSE programme. *Nuclear Engineering and Design*, 12:3561–3578.

de Rocquigny, E., Devictor, N., and Tarantola, S. (2008). *Uncertainty in industrial practice: a guide to quantitative uncertainty management*. John Wiley & Sons.

Dimitrov, N. (2013). *Structural reliability of wind turbine blades: Design methods and evaluation*. PhD thesis, Technical University of Denmark.

Dimitrov, N., Kelly, M., Vignaroli, A., and Berg, J. (2018). From wind to loads: wind turbine site-specific load estimation with surrogate models trained on high-fidelity load databases. *Wind Energy Science*, 3:767 – 790.

DNV-RP-C203 (2016). DNV-RP-C203: Fatigue design of offshore steel structures. Technical report, Det Norske Veritas.

DNV-ST-0437 (2016). DNV-ST-0437: Loads and site conditions for wind turbines. Technical report, Det Norske Veritas.

Dowling, N. E. (1972). Fatigue Failure Predictions for Complicated Stress-Strain Histories. *Journal of Materials, JMLSA*, 7:71 – 87.

Dubrule, O. (1983). Cross validation of kriging in a unique neighborhood. *Journal of the International Association for Mathematical Geology*, 15(6):687–699.

Durante, F. and Sempi, C. (2015). *Principles of copula theory*. CRC press.

Echard, B., Gayton, N., and Lemaire, M. (2011). AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation. *Structural Safety*, 33:145–154.

Fang, K., Liu, M.-Q., Qin, H., and Zhou, Y.-D. (2018). *Theory and application of uniform experimental designs*, volume 221. Springer.

Fang, K.-T., Li, R., and Sudjianto, A. (2006). *Design and Modeling for Computer Experiments*. Chapman & Hall/CRC.

Fatemi, A. and Yang, L. (1998). Cumulative fatigue damage and life prediction theories: a survey of the state of the art for homogeneous materials. *International Journal of Fatigue*, 20(1):9–34.

Fekhari, E., Iooss, B., Chabridon, V., and Muré, J. (2022). Efficient techniques for fast uncertainty propagation in an offshore wind turbine multi-physics simulation tool. In *Proceedings of the 5th International Conference on Renewable Energies Offshore*, pages 837–846.

Fekhari, E., Iooss, B., Muré, J., Pronzato, L., and Rendas, J. (2023). Model predictivity assessment: incremental test-set selection and accuracy evaluation. In *Studies in Theoretical and Applied Statistics*, pages 315–347. Springer.

Forrester, A., Sobester, A., and Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons.

Fuhg., J., Fau, A., and Nackenhorst, U. (2021). State-of-the-Art and Comparative Review of Adaptive Sampling Methods for Kriging. *Archives of Computational Methods in Engineering*, 28:2689–2747.

Geffraye, G., Antoni, O., Farvacque, M., Kadri, D., Lavialle, G., Rameau, B., and Ruby, A. (2011). CATHARE2 V2.5_2: A single version for various applications. *Nuclear Engineering and Design*, 241:4456–4463.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. The MIT Press.

Graf, P., Stewart, G., Lackner, M., Dykes, K., and Veers, P. (2016). High-throughput computation and the applicability of Monte Carlo integration in fatigue load estimation of floating offshore wind turbines. *Wind Energy*, 19(5):861–872.

Hansen, M. and Henriksen, L. (2013). *Basic DTU Wind Energy controller*. Number 0028 in DTU Wind Energy E. DTU Wind Energy.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Hickernell, F. (1998). A generalized discrepancy and quadrature error bound. *Mathematics of computation*, 67(221):299–322.

Hirvoas, A. (2021). *Development of a data assimilation method for the calibration and continuous update of wind turbines digital twins*. PhD thesis, Université Grenoble Alpes.

Huchet, Q. (2019). *Kriging based methods for the structural damage assessment of offshore wind turbines*. PhD thesis, Université Blaise Pascal.

Huchet, Q., Mattrand, C., Beaurepaire, P., Relun, N., and Gayton, N. (2019). AK-DA: An efficient method for the fatigue assessment of wind turbine structures. *Wind Energy*, 22(5):638–652.

Huszár, F. and Duvenaud, D. (2012). Optimally-Weighted Herding is Bayesian Quadrature. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 377 – 386.

IEC-61400-1 (2019). IEC 61400-1: Wind energy generation systems - Part 1: Design requirements. Technical report, International Electrotechnical Commission (IEC).

Iooss, B. (2021). 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.

Iooss, B., Boussouf, L., Feuillard, V., and Marrel, A. (2010). Numerical studies of the meta-model fitting and validation processes. *International Journal of Advances in Systems and Measurements*, 3:11–21.

Jones, D., Schonlau, M., and Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13:455–492.

Jonkman, B. (2009). Turbsim User's Guide: Version 1.50. Technical report, NREL.

Joseph, V. R. and Vakayil, A. (2022). SPlit: An optimal method for data splitting. *Technometrics*, 64(2):166–176.

Kaimal, J., Wyngaard, J., Izumi, Y., and Coté, O. (1972). Spectral characteristics of surface-layer turbulence. *Quarterly Journal of the Royal Meteorological Society*, 98(417):563–589.

Kanagawa, M. and Hennig, P. (2019). Convergence Guarantees for Adaptive Bayesian Quadrature Methods. In *Advances in Neural Information Processing Systems*, volume 32.

Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. (2018). Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences. arXiv:1807.02582.

Kanner, S., Aubault, A., Peiffer, A., and Yu, B. (2018). Maximum dissimilarity-based algorithm for discretization of metocean data into clusters of arbitrary size and dimension. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 51319.

Kaplan, Z., Li, Y., Nakayama, M., and Tuffin, B. (2019). Randomized quasi-Monte Carlo for quantile estimation. In *2019 Winter Simulation Conference (WSC)*, pages 428–439.

Katsikogiannis, G., Sørum, S., Bachynski, E., and Amdahl, J. (2021). Environmental lumping for efficient fatigue assessment of large-diameter monopile wind turbines. *Marine Structures*, 77:102939.

Kennard, R. and Stone, L. (1969). Computer aided design of experiments. *Technometrics*, 11:137–148.

Kim, T., Natarajan, A., Lovera, A., Julan, E., Peyrard, E., Capaldo, M., Huwart, G., Bozonnet, P., and Guiton, M. (2022). A comprehensive code-to-code comparison study with the modified IEA15MW-UMaine Floating Wind Turbine for H2020 HIPERWIND project. *Journal of Physics: Conference Series*, 2265(4):042006.

Klebanov, I., Schuster, I., and Sullivan, T. (2020). A rigorous theory of conditional mean embeddings. *SIAM Journal on Mathematics of Data Science*, 2(3):583–606.

Kleijnen, J. and Sargent, R. (2000). A methodology for fitting and validating metamodels in simulation. *European Journal of Operational Research*, 120:14–29.

Kucherenko, S., Feil, B., Shah, N., and Mauntz, W. (2011). The identification of model effective dimensions using global sensitivity analysis. *Reliability Engineering & System Safety*, 96:440–449.

Lacoste-Julien, S., Lindsten, F., and Bach, F. (2015). Sequential Kernel Herding: Frank-Wolfe Optimization for Particle Filtering. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38, pages 544–552.

Larsen, G. C., Madsen, H., Thomsen, K., and Larsen, T. (2008). Wake meandering: a pragmatic approach. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 11(4):377–395.

Lasserre, M. (2022). *Apprentissages dans les réseaux bayésiens à base de copules non-paramétriques*. PhD thesis, Sorbonne Université.

Lataniotis, C. (2019). *Data-driven uncertainty quantification for high-dimensional engineering problems*. PhD thesis, ETH Zürich.

Le Maître, O. and Knio, O. (2010). *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*. Springer Science & Business Media.

Lebrun, R. and Dutfoy, A. (2009). A generalization of the Nataf transformation to distributions with elliptical copula. *Probabilistic Engineering Mechanics*, 24(2):172–178.

Lemaire, M., Chateauneuf, A., and Mitteau, J.-C. (2009). *Structural reliability*. John Wiley & Sons.

Leobacher, G. and Pillichshammer, F. (2014). *Introduction to quasi-Monte Carlo integration and applications*. Springer.

Li, W., Lu, L., Xie, X., and Yang, M. (2017). A novel extension algorithm for optimized Latin hypercube sampling. *Journal of Statistical Computation and Simulation*, 87:2549–2559.

Li, X. and Zhang, W. (2020). Long-term fatigue damage assessment for a floating offshore wind turbine under realistic environmental conditions. *Renewable Energy*, 159:570–584.

Li, Y., Kang, L., and Hickernell, F. (2020). Is a transformed low discrepancy design also low discrepancy? *Contemporary Experimental Design, Multivariate Analysis and Data Mining: Festschrift in Honour of Professor Kai-Tai Fang*, pages 69–92.

Mak, S. and Joseph, V. (2018). Support points. *The Annals of Statistics*, 46:2562 – 2592.

Marrel, A. and Chabridon, V. (2021). Statistical developments for target and conditional sensitivity analysis: Application on safety studies for nuclear reactor. *Reliability Engineering & System Safety*, 214:107711.

Morio, J. and Balesdent, M. (2015). *Estimation of Rare Event Probabilities in Complex Aerospace and Other Systems: A Practical Approach*. Woodhead Publishing, Elsevier.

Morokoff, W. J. and Caflisch, R. E. (1995). Quasi-Monte Carlo Integration. *Journal of Computational Physics*, 122(2):218–230.

Müller, W. G. (2007). *Collecting Spatial Data*. Springer, 3rd edition.

Murcia, J., Réthoré, P., Dimitrov, N., Natarajan, A., Sørensen, J., Graf, P., and Kim, T. (2018). Uncertainty propagation through an aeroelastic wind turbine model using polynomial surrogates. *Renewable Energy*, 119:910–922.

Müller, K. and Cheng, P. (2018). Application of a Monte Carlo procedure for probabilistic fatigue design of floating offshore wind turbines. *Wind Energy Science*, 3:149 – 162.

Nagler, T., Schellhase, C., and Czado, C. (2017). Nonparametric estimation of simplified vine copula models: comparison of methods. *Dependence Modeling*, 5:99–120.

Nash, J. and Sutcliffe, J. (1970). River flow forecasting through conceptual models part I–A discussion of principles. *Journal of Hydrology*, 10(3):282–290.

Nogales Gómez, A., Pronzato, L., and Rendas, M.-J. (2021). Incremental space-filling design based on coverings and spacings: improving upon low discrepancy sequences. *Journal of Statistical Theory and Practice*, 15(4):77.

Oates, C. J. (2021). Minimum Discrepancy Methods in Uncertainty Quantification. Lecture Notes at École Thématique sur les Incertitudes en Calcul Scientifique (ETICS21), https://www.gdr-mascotnum.fr/etics.html.

Oberkampf, W. and Roy, C. (2010). *Verification and validation in scientific computing*. Cambridge university press.

O'Hagan, A. (1991). Bayes-Hermite quadrature. *Journal of Statistical Planning and Inference*, 29:245–260.

Owen, A. (2003). The dimension distribution and quadrature test functions. *Statistica Sinica*, 13:1–17.

Petrovska, E. (2022). *Fatigue life reassessment of monopile-supported offshore wind turbine structures*. PhD thesis, University of Edinburgh.

Pronzato, L. (2022). Performance analysis of greedy algorithms for minimising a Maximum Mean Discrepancy. preprint, https://hal.inria.fr/hal-03114891/.

Pronzato, L. and Müller, W. (2012). Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22:681–701.

Pronzato, L. and Rendas, M. (2021a). Validation design I: construction of validation designs via kernel herding. preprint, https://arxiv.org/abs/2112.05583.

Pronzato, L. and Rendas, M.-J. (2021b). Validation design I: construction of validation designs via kernel herding. Preprint.

Pronzato, L. and Zhigljavsky, A. (2020). Bayesian quadrature and energy minimization for space-filling design. *SIAM/ASA Journal on Uncertainty Quantification*, 8:959 – 1011.

Qian, P., Ai, M., and Wu, C. (2009). Construction of Nested Space-Filling Designs. *Annals of Statistics*, 37:3616–3643.

Qian, P. and Wu, C. (2009). Sliced space filling designs. *Biometrika*, 96:945–956.

Rasmussen, C. and Williams, C. (2006). *Gaussian processes for machine learning*, volume 1. Springer.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S. (2008). *Global sensitivity analysis: the primer*. John Wiley & Sons.

Sancetta, A. and Satchell, S. (2004). The Bernstein copula and its applications to modeling and approximations of multivariate distributions. *Econometric Theory*, 20(3):535–562.

Santner, T., Williams, B., and Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer.

Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. (2013). Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41:2263–2291.

Shang, B. and Apley, D. (2020). Fully-sequential space-filling design algorithms for computer experiments. *Journal of Quality Technology*, 53:1 – 24.

Sheikholeslami, R. and Razavi, S. (2017). Progressive Latin hypercube sampling: An efficient approach for robust sampling-based analysis of environmental models. *Environmental Modelling & Software*, 93:109–126.

Slot, R. M., Sørensen, J. D., Sudret, B., Svenningsen, L., and Thøgersen, M. L. (2020). Surrogate model uncertainty in wind turbine reliability assessment. *Renewable Energy*, 151:1150 – 1162.

Snee, R. (1977). Validation of regression models: Methods and examples. *Technometrics*, 19:415–428.

Sriperumbudur, B., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. (2010). Hilbert Space Embeddings and Metrics on Probability Measures. *Journal of Machine Learning Research*, 11:1517–1561.

Sullivan, T. (2015). *Introduction to uncertainty quantification*, volume 63. Springer.

Székely, G. J. and Rizzo, M. L. (2013). Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143:1249 – 1272.

Teixeira, R., Nogal, M., O'Connor, A., Nichols, J., and Dumas, A. (2019a). Stress-cycle fatigue design with Kriging applied to offshore wind turbines. *International Journal of Fatigue*, 125:454–467.

Teixeira, R., O'Connor, N., and Nogal, M. (2019b). Probabilistic sensitivity analysis of offshore wind turbines using a transformed Kullback-Leibler divergence. *Structural Safety*, 81:101860.

Teymur, O., Gorham, J., Riabiz, M., and Oates, C. (2021). Optimal Quantisation of Probability Measures Using Maximum Mean Discrepancy. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pages 1027 – 1035.

Van den Bos, L. (2020). *Quadrature Methods for Wind Turbine Load Calculations*. PhD thesis, Delft University of Technology.

Van Kuik, G., Peinke, J., Nijssen, R., Lekou, D., Mann, J., Sørensen, J., Ferreira, C., van Wingerden, J., Schlipf, D., Gebraad, P., et al. (2016). Long-term research challenges in wind energy–a research agenda by the European Academy of Wind Energy. *Wind energy science*, 1(1):1–39.

Vanem, E., Fekhari, E., Dimitrov, N., Kelly, M., Cousin, A., and Guiton, M. (2023). A joint probability distribution model for multivariate wind and wave conditions. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 86847, page V002T02A013. American Society of Mechanical Engineers.

Velarde, J., Kramhøft, C., and Sørensen, J. (2019). Global sensitivity analysis of offshore wind turbine foundation fatigue loads. *Renewable Energy*, 140:177 – 189.

Wilkie, D. and Galasso, C. (2021). Gaussian process regression for fatigue reliability analysis of offshore wind turbines. *Structural Safety*, 88:102020.

Xu, Y. and Goodacre, R. (2018). 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.

Zwick, D. and Muskulus, M. (2015). The simulation error caused by input loading variability in offshore wind turbine structural analysis. *Wind Energy*, 18:1421 – 1432.