TU Delft

Delft University of Technology

Document Version
Final published version

Licence
CC BY

Citation (APA)
Guo, L. L., van Driel, W. D., & Zhang, G. Q. (2026). Gaussian Process Regression Modeling Benchmark of Atomic Vacancy Concentrations in Solder Bumps. *electronics*, *15*(14), Article 3030. https://doi.org/10.3390/electronics15143030

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

electronics

MDPI

Article

Gaussian Process Regression Modeling Benchmark of Atomic Vacancy Concentrations in Solder Bumps

Leo Guo *, Willem D. van Driel and Guoqi Zhang *

Department of Electronic Components, Technology and Materials, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands; willem.vandriel@tudelft.nl

* Correspondence: l.l.guo@tudelft.nl (L.G.); g.q.zhang@tudelft.nl (G.Z.)

Abstract

Expensive and scarce design objective data is an increasingly relevant problem in modern engineering applications. Gaussian process regression is a popular method that tackles this challenge by efficiently performing response surface surrogate modeling across a global design space. However, practitioners of this modeling procedure often fail to adequately tailor the regression model to the data. This work aims to address this issue by considering a variety of covariance kernels and fixed or learnable noise values. We demonstrate the efficacy of this consideration by means of both a synthetic design problem and a real-life micro-electronic design problem. This problem requires a surrogate model of the atomic vacancy concentration within a solder bump structure. Calculating this concentration relies on expensive thermo-mechanical finite element simulations. Previously, the commercial data-driven pipeline package Optimus has been used to build a Gaussian process regression model. However, using GPyTorch, we can vary surrogate modeling choices to obtain a model that more effectively generalizes to the entire design space. This process is not much more time-consuming than cross-validated Gaussian process regression with Optimus, taking only 0.31% of the time needed to sample the design space. This shows that careful selection of response surface modeling hyperparameters is both non-trivial and effective when constructing an efficient and reliable surrogate model.

Keywords: Gaussian process regression; response surface surrogate modeling; hyperparameter selection; micro-electronic design; solder reliability

1. Introduction

Design problems in modern engineering rarely have a closed-form solution. Therefore, numerical simulation capabilities are used ubiquitously to solve these design problems. However, the growing complexity of engineering problems imposes costly or time-consuming data acquisition. This occurs for data obtained from experiments [1,2], computations [2,3] or analytically [4]. For example, designing an aircraft involves simulations with millions of degrees of freedom that require thousands of CPU-hours [5,6]. As a result, only limited amounts of data are available to do analysis or design in practice.

Given the black-box nature of the objective function of an engineering problem, it is of interest to obtain an approximate global response based on a small design of experiments (DoE), consisting of a matrix of design rows $\mathbf{X}$ and the corresponding vector of objective evaluations $\mathbf{y}$. This process, called response surface modeling (RSM) [7], is usually achieved by means of statistical regression and can be used for (much) cheaper objective approximations for follow-up steps in a design optimization process. In the literature, a wide

Check for updates

Academic Editor: Fabio Postiglione

Received: 1 May 2026
Revised: 28 June 2026
Accepted: 7 July 2026
Published: 9 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

variety of regression models exist to model the relationship between **X** and **y**. In terms of complexity, RSM can be achieved by models ranging from linear ones, such as ordinary least squares, ridge [8] or lasso [9] regression, to highly non-linear models such as artificial neural networks (ANNs), random forest regression [10], and Bayesian ANNs [11].

Performing RSM with expensive simulation-driven design objectives is often met with two major bottlenecks: data scarcity and aleatoric uncertainty, i.e., the inherent stochastic nature of objective output data. These bottlenecks are effectively addressed by Gaussian process (GP) regression [12]. This method was conceptualized in the field of geostatistics as kriging in the 1950s [13], with model parameter selection based on the analysis of observed variances, rather than the maximum likelihood estimate [14], which is more common in modern-day machine learning. A GP model is a surrogate model that can interpolate between scarce points of data, while also providing global uncertainty quantification. The predictive power of a GP model is encoded by its covariance function, and this useful property has made it a popular choice for doing RSM with sparse and uncertain data, e.g., in astronomy [15], commodity pricing [16], and electronic remaining-useful-lifetime predictions [17]. In particular, we recognize a big potential for performing efficient and reliable RSM in electrical reliability engineering. Accumulated stress and electromigration in the solder joints of electronic chip packages are often the cause of electronic component failures [18,19]. Modeling this behavior involves expensive thermo-mechanical finite-element simulations [20,21], and as such, several previous data-driven attempts to tackle solder design problems have been undertaken with RSMs based on GP regression (GPR) models [22–25].

Various toolsets exist to accommodate the computation of GPR models in the broader context of state-of-the-art data-driven design. One such software, called Optimus, is a commercial package developed by Noesis Solutions [26]. Users of Optimus regularly make use of the features provided by the software to automate design space sampling and objective evaluation in a design optimization process [24,27,28], reducing the repetitive workload by doing so. The software solution offers a wide range of regression methods for surrogate modeling, such as polynomial regression and ANNs.

Despite the availability of readily applicable RSMs, the proper selection of hyper-parameters, i.e., any parameter that is initialized and not learned from the data, such as a GPR model's covariance kernel, is a topic that is rarely thoroughly discussed. The lack of such a data-driven standard stands in stark contrast to the ever-increasing complexity of computer simulations. For example, improving the best design through Bayesian maximization of energy absorption of spiniodoid structures [29,30] required multi-fidelity GPR models. However, systematically assessing the available covariance kernels could similarly lead to material design improvement. Thus, given these observations, it is clear that a guideline of best practices for performing RSM with GPR models in expensive simulation-based design engineering is required. The existence of various tools to this end necessitates a proper benchmark and raises the question of whether Optimus can compete against open-source solutions in Python that facilitate GPR computations, such as GPyTorch (https://gpytorch.ai/) [31], based on PyTorch (https://pytorch.org/) [32], and GPFlow (https://www.gpflow.org/) [33], based on TensorFlow (https://www.tensorflow.org/) [34]. In particular, GPyTorch has previously been used to provide a range of possible GPR models, which in turn efficiently provide optimized solder joints of an electronic package using BO [35]. To the authors' knowledge, the possibility of performing GPR with Optimus has not previously been benchmarked against an open-source toolkit. This article, therefore, aims to shed light on this literature gap and present a novel benchmark of Optimus's capability to adequately establish GPR models by demonstrating this against a solder reliability engineering use case scenario.

The main novel contributions of the paper are as follows:

- We provide a novel, in-depth benchmark of the GPR performance provided by the open-source GPyTorch tool compared to the commercial Optimus software package. This is applied to synthetic data as well as a realistic and practical engineering use case of designing solder bumps in an electronic package.
- We provide awareness and solid justification for the practice of carefully selecting the hyperparameters of GPR models (kernel type, noise parameter fixture) as a guideline towards performing RSM. This is based on both synthetic results and the observations from the engineering problem.

The outline of the paper is as follows: Section 2 introduces the relevant mathematical background concerning Gaussian process regression modeling, and also quantitatively reports on the effect of kernel types and noise parameter fixtures on a synthetic objective. Section 3 introduces the context of the solder bump design problem and compares the GPR model obtained from Optimus against several GPR models built with GPyTorch (https://gpytorch.ai/). The findings are subsequently discussed in Section 4.

2. Gaussian Process Regression Modeling

2.1. Gaussian Processes

In the context of GP modeling, it is assumed that $f(\mathbf{x})$ is a random variable with $\mathbb{E}(f(\mathbf{x})) = 0$ and prior covariance function or kernel κ for any $\mathbf{x} \in [0,1]^D$ [36]. Given $\mathbf{u}, \mathbf{v} \in [0,1]^D$, the defining property is as such:

$$\kappa(\mathbf{u}, \mathbf{v}) := \mathrm{Cov}(f(\mathbf{u}), f(\mathbf{v})). \tag{1}$$

Assume that the covariance function $\kappa : [0,1]^D \times [0,1]^D \to \mathbb{R}$ gives rise to a GP model that attempts to explain $\mathbf{y}$ given $\mathbf{X}$. While some authors include an explicit prior mean function, the zero prior mean assumption in the present manuscript implies that selecting a GP model is equivalent to selecting a covariance kernel [12].

In the literature, the most popular prior covariance functions include the radial basis function (RBF) kernel, the Matérn kernel, and the rational quadratic (RQ) kernel [37–40]. These are defined as follows:

$$\begin{aligned}
\kappa_{\mathrm{RBF},\boldsymbol{\theta}_{\mathrm{RBF}}}(\mathbf{u}, \mathbf{v}) &:= c \cdot \exp\left(-\frac{d^2(\mathbf{u}, \mathbf{v}; \boldsymbol{\lambda})}{2}\right) + s^2\delta(\mathbf{u} - \mathbf{v}), \\
\kappa_{\mathrm{Mat},\boldsymbol{\theta}_{\mathrm{Mat}}}(\mathbf{u}, \mathbf{v}) &:= c \cdot \left(1 + \sqrt{5}d(\mathbf{u}, \mathbf{v}; \boldsymbol{\lambda}) + \tfrac{5}{3}d^2(\mathbf{u}, \mathbf{v}; \boldsymbol{\lambda})\right) \exp\left(-\sqrt{5}d(\mathbf{u}, \mathbf{v}; \boldsymbol{\lambda})\right) + s^2\delta(\mathbf{u} - \mathbf{v}), \\
\kappa_{\mathrm{RQ},\boldsymbol{\theta}_{\mathrm{RQ}}}(\mathbf{u}, \mathbf{v}) &:= c \cdot \left(1 + \frac{d^2(\mathbf{u}, \mathbf{v}; \boldsymbol{\lambda})}{2\alpha}\right)^{-\alpha} + s^2\delta(\mathbf{u} - \mathbf{v}),
\end{aligned} \tag{2}$$

where $\alpha > 0$ is a mixture parameter in the case of the RQ kernel and

$$d(\mathbf{u}, \mathbf{v}; \boldsymbol{\lambda}) := \sqrt{(\mathbf{u} - \mathbf{v})^\top \mathrm{diag}(\boldsymbol{\lambda})^{-2}(\mathbf{u} - \mathbf{v})}. \tag{3}$$

The scalar value $c > 0$ is known as the amplitude parameter, while $\boldsymbol{\lambda} := (\lambda_1, \ldots, \lambda_D)^\top$ is the length scale parameter vector of the kernel. Additionally, s^2 is the prior noise variance parameter, which is only assumed to be active when κ is used to calculate the variance at a particular design. Hence, this parameter is multiplied by the appropriately translated Dirac-δ function in Equation (2). Finally, each kernel function's parameters are aggregated into a parameter vector $\boldsymbol{\theta}$, e.g., $\boldsymbol{\theta}_{\mathrm{RBF}} := (c, \boldsymbol{\lambda}, s^2)$.

As dependent variables in a Bayesian context, it is assumed that $\mathbf{y}$ is a realization of a random vector $\mathbf{Y}$, and that any model parameter vector $\boldsymbol{\theta}$ is a realization of a random

vector $\boldsymbol{\Theta}$. A GP model implies that $\mathbf{Y}$ follows a multivariate normal distribution when a model parameter vector $\boldsymbol{\theta}$ is established. The prior distribution of $\mathbf{Y}$ conditioned on $\boldsymbol{\Theta} = \boldsymbol{\theta}$ is given by

$$\mathbf{Y}|(\boldsymbol{\Theta} = \boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\boldsymbol{\theta}}), \text{ with covariance matrix } \mathbf{K}_{\boldsymbol{\theta}} = \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}) := (\kappa_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1,\ldots,N}. \quad (4)$$

For any given $\mathbf{x} \in [0,1]^D$, it can then be shown that the parameterized posterior is also multivariate normal [41]:

$$f(\mathbf{x})|(\mathbf{Y} = \mathbf{y}, \boldsymbol{\Theta} = \boldsymbol{\theta}) \sim \mathcal{N}(\mu_{\boldsymbol{\theta}}(\mathbf{x}), \sigma^2_{\boldsymbol{\theta}}(\mathbf{x})), \quad (5)$$

where

$$\mu_{\boldsymbol{\theta}}(\mathbf{x}) := \kappa_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{X})^\top \mathbf{K}_{\boldsymbol{\theta}}^{-1} \mathbf{y}, \quad (6)$$

$$\sigma^2_{\boldsymbol{\theta}}(\mathbf{x}) := \kappa_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}) - \kappa_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{X})^\top \mathbf{K}_{\boldsymbol{\theta}}^{-1} \kappa_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{X}), \quad (7)$$

with $\kappa_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{X}) := (\kappa_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_1), \ldots, \kappa_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_N))^\top$ as a column vector of length N. It is also convenient to denote $\mu_{\boldsymbol{\theta}}(\mathbf{Z}) := (\mu_{\boldsymbol{\theta}}(\mathbf{z}_1), \ldots, \mu_{\boldsymbol{\theta}}(\mathbf{z}_M))^\top$ as a column vector of length M whenever $\mathbf{Z} = (\mathbf{z}_1^\top, \ldots, \mathbf{z}_M^\top)^\top$ is a column of M design rows. The same notation is introduced for $\sigma^2_{\boldsymbol{\theta}}$.

While Equation (5) provides a closed form description of the (parameterized) posterior distribution of $f(\mathbf{x})|(\mathbf{Y} = \mathbf{y}, \boldsymbol{\Theta} = \boldsymbol{\theta})$, it is desirable to obtain the distribution of $f(\mathbf{x})|(\mathbf{Y} = \mathbf{y})$, also called the posterior predictive distribution (PPD), independent of the model parameter vector $\boldsymbol{\theta}$. Unfortunately, the PPD can often not be determined analytically, which means that it is necessary to approximate it. One choice for such a PPD estimate, in the context of GP modelling, is the maximum (log) likelihood estimate (MLE) [12]. The MLE is given by

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\text{MLE}} := \operatorname*{argmax}_{\boldsymbol{\theta} \in \mathcal{T}_\kappa} \ln(p(\mathbf{Y}|\boldsymbol{\Theta} = \boldsymbol{\theta})[\mathbf{y}]) &= \operatorname*{argmax}_{\boldsymbol{\theta} \in \mathcal{T}_\kappa} \ln(\phi_{\mathbf{0}, \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})}(\mathbf{y})) \\ &= \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathcal{T}_\kappa} \ln(\det(\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}))) + \mathbf{y}^\top \mathbf{K}_{\boldsymbol{\theta}}^{-1}(\mathbf{X}) \mathbf{y}, \quad (8)\end{aligned}$$

where

$$\phi_{\boldsymbol{\mu}, \boldsymbol{\Sigma}}(\mathbf{z}) := \frac{1}{(2\pi)^{N/2} \sqrt{\det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2} (\mathbf{z} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{z} - \boldsymbol{\mu}) \right)$$

denotes the N-dimensional multivariate normal probability density function with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

Using the MLE estimate $\hat{\boldsymbol{\theta}}_{\text{MLE}}$, the following approximation is obtained for the PPD density:

$$p(f(\mathbf{x})|\mathbf{Y} = \mathbf{y})[y] \approx p(f(\mathbf{x})|\mathbf{Y} = \mathbf{y}, \boldsymbol{\Theta} = \hat{\boldsymbol{\theta}}_{\text{MLE}})[y] = \phi_{\hat{\mu}(\mathbf{x}), \hat{\sigma}^2(\mathbf{x})}(y) =: \hat{\phi}(y; \mathbf{x}, \mathbf{y}, \kappa), \quad (9)$$

where the notation

$$\hat{\mu} := \mu_{\hat{\boldsymbol{\theta}}_{\text{MLE}}}, \quad \hat{\sigma}^2 := \sigma^2_{\hat{\boldsymbol{\theta}}_{\text{MLE}}} \quad (10)$$

is used for the optimized parameterized posterior parameters in Equations (6) and (7). In other words, the PPD density can be approximated by the (normal) probability density $\hat{\phi}$ of the parameterized posterior distribution in Equation (9) by utilizing the maximum likelihood estimate $\hat{\boldsymbol{\theta}}_{\text{MLE}}$ for the model parameter vector. As a naming convention, the normal distribution $\mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ will be called the regression-predictive distribution (RPD), fully characterized by its corresponding probability density $\hat{\phi}$, named the RPD density.

2.2. Covariance Kernels and GPR Quality

Training a GPR follows the standard train–test splitting procedure known throughout supervised learning practice. Concretely, this is undertaken by first splitting the DoE $\mathbf{D} = (\mathbf{X}, \mathbf{y})$ into train $\mathbf{D}_{\text{train}} = (\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}})$ and test sets $\mathbf{D}_{\text{test}} = (\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}})$. Subsequently, a GPR model is constructed over $\mathbf{D}_{\text{train}}$ by solving the likelihood optimization problem (Equation (8)), and the resulting RPD density $\hat{\phi}$ is validated over $\mathbf{D}_{\text{test}}$ by means of a performance metric. One such metric is the relative MSE (this quantity is sometimes known as the fraction of variance unexplained (FVU), equal to $1 - R^2$ where R^2 is the coefficient of determination) (RelMSE) of a GPR model. Given a non-constant ground truth vector $\mathbf{y} := (y_1, \ldots, y_M)$ and a prediction vector $\hat{\mathbf{y}} := (\hat{y}_1, \ldots, \hat{y}_M)$, the RelMSE is defined by

$$\text{RelMSE}(\mathbf{y}, \hat{\mathbf{y}}) := \frac{\text{MSE}(\mathbf{y}, \hat{\mathbf{y}})}{\text{Var}(\mathbf{y})} = \frac{\sum_{j=1}^{M} (y_j - \hat{y}_j)^2}{\sum_{i=1}^{M} (y_j - \bar{y})^2} \quad \text{with} \quad \bar{y} := \frac{1}{M} \sum_{j=1}^{M} y_j. \tag{11}$$

From the way that a GPR model is defined, it is clear that the covariance function has a significant impact on how the surrogate model behaves. As a consequence, building a high-quality regression surrogate model requires a fundamental level of understanding of the underlying data. As with many engineering use cases, this knowledge is often lacking. As a result, there exists a significant risk in selecting inaccurate model hyperparameters or an unsuitable surrogate model altogether, which has previously been shown with time series data [42]. To demonstrate this, we use GPyTorch to train a GPR model for each of the kernel types in Equation (2). As an objective function, the following 1D objective function f, also called the Schwefel function, is selected:

$$f(x) = 418.92829 - x \sin\left(\sqrt{|x|}\right) \tag{12}$$

The domain of this function is usually restricted to the interval $[-500, 500]$, on which f exhibits a highly non-linear and multimodal image or objective landscape. This makes the objective function an ideal candidate to pose as an easily visualizable GPR challenge.

Twenty Latin hypercube samples are selected on the objective domain, of which 15 are used for training and 5 for testing. More information about Latin hypercube sampling can be found in Appendix A. For each GPR model, 150 iterations with the gradient-based Adam optimizer [43] are used to numerically maximize the likelihood for the MLE (Equation (8)), which takes an average of 2.67 s on an AMD Ryzen 7 7700 CPU (8-core, 3.8 GHz) (this is the computing power for all computations used throughout the manuscript, which includes the finite element simulation results in the next section). The results can be seen in Figure 1.

The GPR model that assumes the Matérn covariance kernel can approximate the training samples much more confidently, as Figure 1 shows. More importantly, the Matérn model interpolates the training data more accurately than the other two models. This can be quantitatively confirmed by considering the overall error RelMSE$(\mathbf{y}, \hat{\mathbf{y}})$, i.e., the relative MSE between the prediction and ground truth at both the training and the test data combined, and can be viewed in Table 1.

Table 1. Results from the GPR models of the Schwefel function data. All values are given in %, relative to the objective data scale.

	κ_{Mat}	κ_{RBF}	κ_{RQ}
Test error RelMSE $(\mathbf{y}_{\text{test}}, \hat{\mathbf{y}}_{\text{test}})$	36.0	39.2	40.4
Overall error RelMSE $(\mathbf{y}, \hat{\mathbf{y}})$	9.26	83.4	88.4
regression-predictive noise $\hat{s}$	6.15	108	112

It should be noted that the test error remains similar across the three kernel types, between 35–45%. This means that no single covariance function can generalize the surrogate model to unseen data better than the others. However, it should also be noted that the output data obtained from the synthetic objective used in Figure 1 has not been distorted by any aleatoric uncertainty, which makes the regression-predictive noise value $\hat{s}$ interesting to observe. It can be seen that the predicted noise value is much lower in the case of the Matérn kernel. This shows that assuming a noiseless PPD is a (rightfully) worthwhile consideration to make in order to achieve a lower overall error.

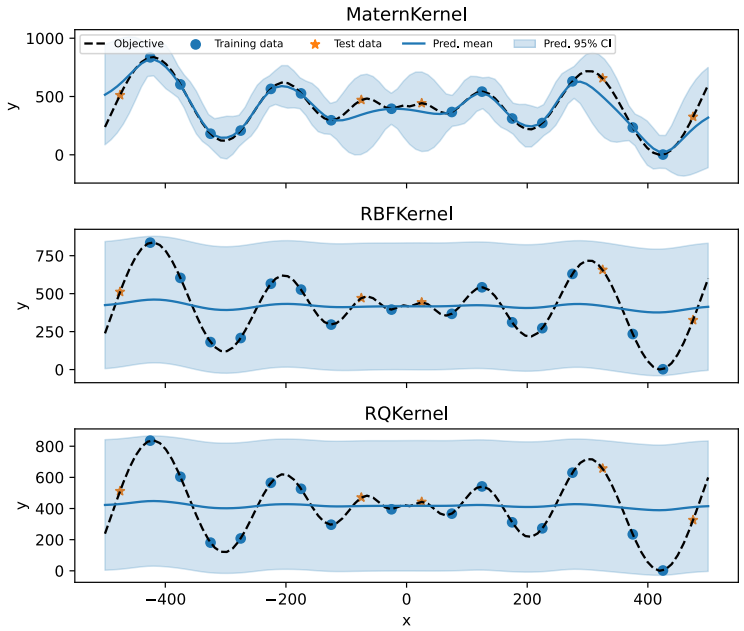


Figure 1. GPR models of the Schwefel function using three common covariance kernels.

By fixing the regression-predictive noise value of $\hat{s} \equiv 0.1\%$ and training an otherwise similar set of GPR models as in Figure 1, the results in Figure 2 are obtained.

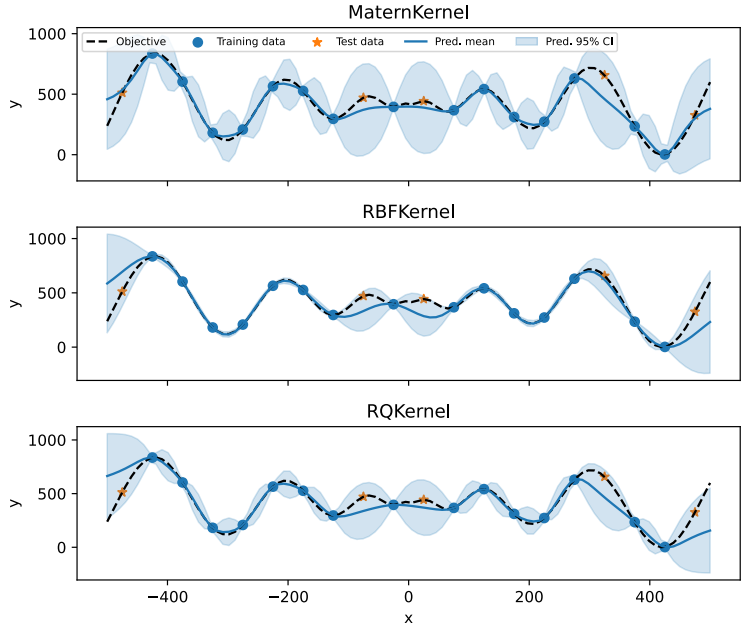


Figure 2. GPR models of the Schwefel function using three common covariance kernels, with fixed homoskedastic noise values.

Visually, it can be confirmed that the training data are being interpolated by the fixed-noise GPR model with the RBF kernel, as well as the one with the RQ kernel. The overall error has been reduced accordingly, as Table 2 shows.

Table 2. Results from the GPR models of the Schwefel function data after fixing the noise value. All values are given in %, relative to the objective data scale.

	$\kappa_{\mathbf{Mat}}$	$\kappa_{\mathbf{RBF}}$	$\kappa_{\mathbf{RQ}}$
Test error RelMSE $(\mathbf{y}_{\text{test}}, \hat{\mathbf{y}}_{\text{test}})$	28.9	65.2	82.3
Overall error RelMSE $(\mathbf{y}, \hat{\mathbf{y}})$	7.21	16.3	20.7

Interestingly, Table 2 demonstrates that the test error for the GPR models that use the RBF and RQ has increased when compared to the noisy counterpart models in Table 1. Since the test error of the Matérn model *decreases* when assuming a fixed noise level, the results suggest that the Matérn kernel is an inherently better choice for building a surrogate model of the Schwefel function when no noise is assumed.

3. Benchmark Results

The synthetic results from the previous section show that it is possible to obtain a surrogate model with lower test error by reducing the likelihood search space—in particular, by assuming little aleatoric noise. However, this comes at the (computational) cost of training and assessing multiple GPR models. Indeed, the six models trained for the comparisons in Figures 1 and 2 take 16 s in total. This added expense can be justified when the objective function of the design problem is (very) expensive, e.g., the output of a finite element analysis.

3.1. Solder Bump Design

One such design problem with an expensive objective concerns improving the wafer-level chip-scale packaging process. This process involves the placement of tin (Sn) ball-shaped solder bump structures that connect a chip package to a printed circuit board by means of a copper (Cu) substrate pad. Under high current density levels and temperature, the effect of electromigration can cause the creation of unwanted intermetallic compounds at the site of the bumps [19]. This eventually leads to the formation of interfacial voids between the bump and the substrate, causing an open circuit and failure of the package. This effect is shown (reprinted from [44], with the permission of AIP Publishing) in Figure 3.

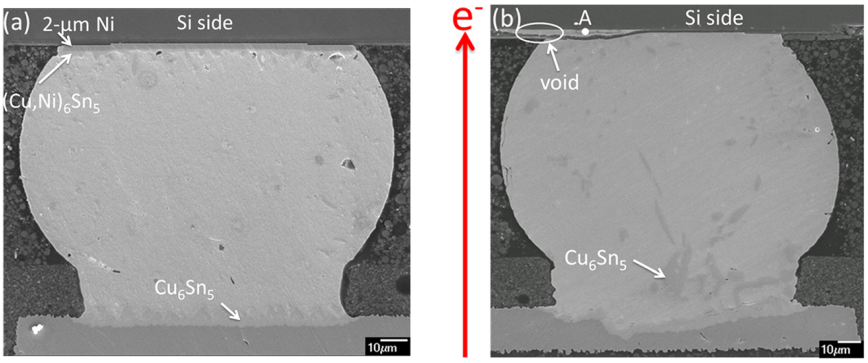


Figure 3. A cross–section image of a solder bump before (**a**) and after (**b**) being affected by voids (white ellipse) at the anode joint caused by electromigration. This is induced by a recurring electrical current—opposite to the electron flow, indicated by the red arrow—and the spreading formation of compounds such as a copper–tin alloy (Cu_6Sn_5).

Previously published reports have identified that geometric properties of the solder bumps can affect the rate at which electromigration occurs [45], which is especially the case in environments with a large difference between the operational and inoperable temperatures [46]. A study by Du et al. [24] recognizes that the greatest normalized concentration of atomic vacancies (dimensionless quantity) is an objective to minimize, which depends on the following geometric design parameters:

- d_{pad}: diameter of the Cu pad;
- t_{pad}: thickness of the Cu pad;
- d_{ubm}: diameter of the under-bump metallurgy (UBM);
- t_{ubm}: thickness of the UBM;
- d_{rep1}: diameter of the die repassivation opening;
- d_{del}: diameter of the solder bump;
- h_{ball}: height of the solder bump.

The schematic diagram in Figure 4 presents a geometric interpretation of these parameters.

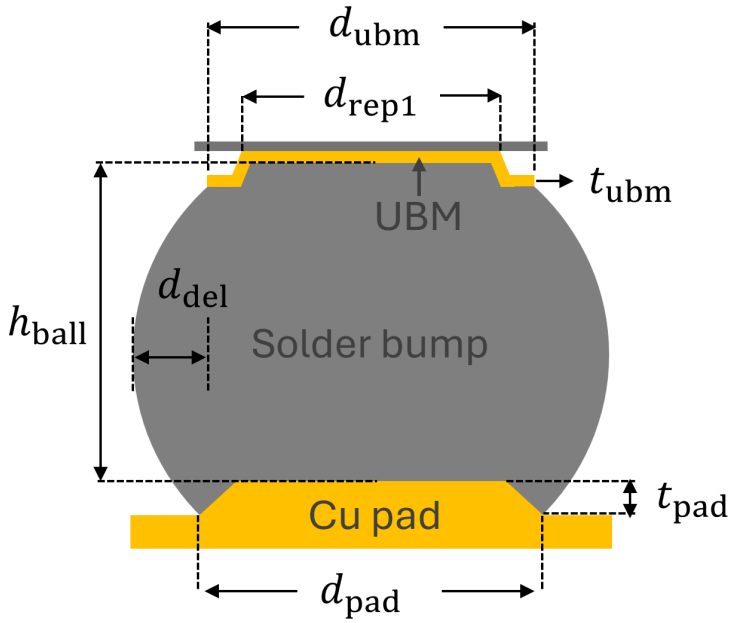


Figure 4. Geometry of a solder bump in contact with the UBM and the copper substrate.

The selected design parameters each have a feasible range, based on previously known nominal values. These intervals are given in Table 3.

Table 3. The solder ball parameters and their intervals.

Design Parameter	Lower Bound (mm)	Upper Bound (mm)
d_{pad}	150	180
t_{pad}	15	30
d_{ubm}	180	220
t_{ubm}	30	160
d_{rep1}	5	15
d_{del}	10	40
h_{ball}	130	165

Based on the design space Table 3, the authors have used Optimus 2019.1 to chart the relationship between the concentration and these geometric parameters in a data-driven manner. The study commences with 100 Latin hypercube samples drawn from the design

space. The maximum concentration for each sample was subsequently evaluated by means of a finite element analysis, facilitated by the commercial finite element solver Ansys Mechanical [24]. The thermomechanical simulation comprised a double solder bump setup, subjected to an operational electric current of 5A and a temperature of 150 °C, sustained for 1000 h. An example of the result of a finite element simulation is given in Figure 5.

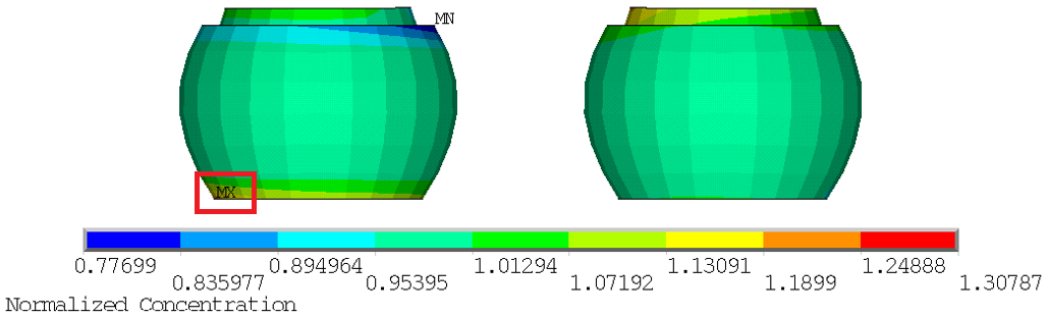


Figure 5. The normalized atomic vacancy concentration across two solder bumps after thermomechanical simulation testing. The value of the maximum concentration is of particular importance to minimize, the location of which is marked by MX and indicated by the red rectangle.

A projection plot of the maximum concentration (Max. conc.) of all the evaluated design samples is shown in Figure 6.

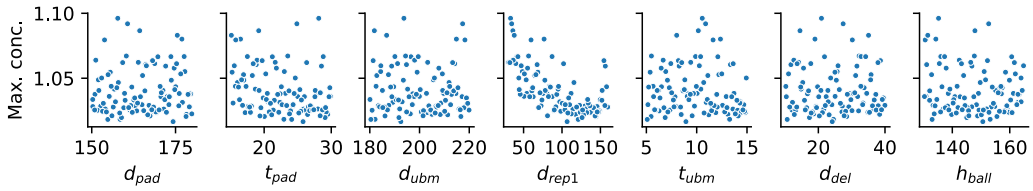


Figure 6. Each subfigure represents 100 Latin hypercube sample evaluations of the maximum concentration, projected to the corresponding design parameter of the sample.

Each evaluation of the concentration objective takes approximately 8 min to simulate. The full design of experiments, as displayed in Figure 6, therefore takes over 13 h to obtain. We observe that there is no significant (linear) correlation between the maximum concentration and any individual design parameter. This assessment relies on the fact that the design samples are distributed evenly across each individual parameter due to the properties of the Latin hypercube sampling. The strongest correlation is observed with the repassivation diameter d_{rep1}, attaining a linear correlation coefficient of -0.67. The remaining correlations do not exceed 0.3 in magnitude, which means that none of the discussed parameters behave trivially with respect to the objective and can therefore be ruled out as an integral part of the design space.

3.2. k-Fold Validated GPR Model Benchmark

Since the objective evaluations are not cheap (8 min)—compared to training a GPR model (2.67 s)—this makes GPR a suitable candidate technique to provide a surrogate model.

Optimus provides state-of-the-art *k*-fold cross-validation to select the regression model with the best generalizing performance, given a model type and its hyperparameters. This workflow has been successfully used in supervised machine learning-assisted engineering applications, such as medical tomography [47], energy consumption footprints in construction [48], and polymer modeling [49]. Concretely, this is undertaken by first splitting the DoE $\mathbf{D}$ into not only train ($\mathbf{D}_{\text{train}}$) and test sets ($\mathbf{D}_{\text{test}}$), but also a validation set

($\mathbf{D}_{\text{val}}$). The experiments in $\mathbf{D}_{\text{train}}$ and $\mathbf{D}_{\text{val}}$ are subsequently rearranged into k different train–validation splits. This results in k maximum likelihood estimates $\hat{\theta}_1, \ldots, \hat{\theta}_k$, out of which the best-performing model parameter vector is chosen based on the RelMSE metric. Lastly, the capability of the resulting regression model to generalize the prediction of the objective function is tested by calculating its performance on $\mathbf{D}_{\text{test}}$.

The study by Du et al. [24] has yielded a 10-fold cross-validated GPR model, obtained with Optimus, which makes use of the RBF kernel. Critically, Optimus presently only supports the RBF covariance kernel to perform GP modeling. This provides an opportunity to benchmark this previously obtained regression model by comparing it to an array of other GPR models that can be facilitated by GPyTorch (https://gpytorch.ai/). To match the 10-fold cross-validation feature provided by Optimus, a similar 10-fold cross-validated benchmark study is conducted using GPyTorch. Firstly, the full DoE is standardized to the unit cube. Then, the standardized design matrix $\mathbf{X}$ and normalized objective output vector $\mathbf{y}$ is split into train ($\mathbf{X}_{\text{train}}$,$\mathbf{y}_{\text{train}}$), validation ($\mathbf{X}_{\text{val}}$,$\mathbf{y}_{\text{val}}$), and test parts ($\mathbf{X}_{\text{test}}$,$\mathbf{y}_{\text{test}}$), each containing 80, 10, and 10 design rows respectively. $\mathbf{X}_{\text{train}}$ and $\mathbf{y}_{\text{train}}$ are then used to construct a variety of GPR models, each validated by $\mathbf{X}_{\text{val}}$ and $\mathbf{y}_{\text{val}}$. In particular, ten different train–validation configurations are considered. Apart from the RBF covariance kernel, which is represented by the Optimus GPR model, two covariance kernel types are used: the Matérn and RQ kernels. Finally, $\mathbf{X}_{\text{train}}$ and $\mathbf{y}_{\text{train}}$ are used to probe whether the regression model with the lowest validation error generalizes well to unseen data. This 10-fold cross-validation workflow takes a total of 25 s per covariance type for both Optimus and GPyTorch, which amounts to 75 s total. The results from this benchmark are presented in Table 4.

Table 4. Results from 10-fold validated GPR models of the 7-parameter solder bump data. The values corresponding to the RBF kernel derive from the Optimus model, while the Matérn and RQ values correspond to GPyTorch models. All values are given in %, relative to the objective data scale.

	κ_{Mat}	κ_{RBF} **(Optimus)**	κ_{RQ}
Lowest validation error RelMSE $(\mathbf{y}_{\text{val}}, \hat{\mathbf{y}}_{\text{val}})$	11.5	15.4	13.5
Test error RelMSE $(\mathbf{y}_{\text{test}}, \hat{\mathbf{y}}_{\text{test}})$	10.1	31.1	26.9
Overall error RelMSE $(\mathbf{y}, \hat{\mathbf{y}})$	2.30	5.90	2.51
Regression-predictive noise $\hat{s}$	1.80	5.41	4.31

From Table 4, it can be seen that the lowest overall error between $\mathbf{y}$ and $\hat{\mathbf{y}}$ was achieved with the Matérn kernel. Moreover, the Matérn kernel GPR model with the lowest validation error has a similar test error (1.4% difference), while the Optimus model (using the RBF covariance kernel) has more than double the test error compared to the validation error. In other words, the RBF covariance, which is the only GPR modeling covariance option provided by the Optimus toolkit, does not yield the most representative GPR model when generalized to the entire design space.

The atomic vacancy data obtained through the finite element simulations are affected by rounding errors, partially due to mesh discretization parameters. While these errors are deterministic in theory, they could be modeled as the realizations of a stochastic process. This is in line with the stochastic modeling of measurement errors that occur in practice. In other words, there is no distinction made between point-by-point reproducible deterministic errors or truly stochastic elements; both effects are modeled in the same way by means of Bayesian statistics. For this reason, it is useful to compare the regression-predictive noise values $\hat{s}$ in Table 4.

It can be observed that the $\hat{s}$ of the best-found Matérn kernel GPR model is relatively small compared to the unit variance of a normalized objective data set; in other words, this indicates a high signal-to-noise ratio across the objective. To this end, by fixing the noise value $\hat{s} \equiv 0.1\%$, the results in Table 5 are obtained. It should be explicitly noted that Optimus does not support the fixture of separate kernel parameters before the determination of the MLE. Hence, the noiseless GPR model with the RBF covariance kernel is trained using the GPyTorch implementation.

Table 5. Results from 10-fold validated GPyTorch GPR models of the 7-parameter solder bump data after fixing the noise value. All values are given in %, relative to the objective data scale.

	$\kappa_{\mathbf{Mat}}$	$\kappa_{\mathbf{RBF}}$	$\kappa_{\mathbf{RQ}}$
Lowest validation error RelMSE $(\mathbf{y}_{\text{val}}, \hat{\mathbf{y}}_{\text{val}})$	10.1	7.64	10.8
Test error RelMSE $(\mathbf{y}_{\text{test}}, \hat{\mathbf{y}}_{\text{test}})$	12.6	44.2	8.11

It should first be noted that the amount of time it takes to obtain the results in Table 5 is similar to that of Table 4 (75 s). It can be seen that the test error of a GPR model with an RQ kernel decreases significantly (26.9% $\searrow$ 8.11%), and even outperforms the model with the Matérn kernel, which was the previous best choice. On the other hand, fixing the noise value leads to worse generalization by the RBF covariance model, i.e., worse than the Optimus model in Table 4.

Judging by the lack of clear trends in the data from Figure 6, except for the d_{rep1} parameter, the objective seems (highly) non-trivially affected by the set of parameters. This is reflected by some sizable discrepancies between the lowest validation error and the test error in Tables 4 and 5, proving that some kernels struggle with design parameters with similar (low) sensitivities and/or multimodal objective behavior. This means that it could be a difficult task to minimize the maximum concentration across the design space. To illustrate the non-triviality of this global minimization effort, it is useful to examine the resulting GPR response surface with the best-performing GPyTorch model. By selecting the t_{ubm} and d_{rep1} parameters and fixing the remaining five parameters at 0.25 times the distance between the lower and upper bounds as nominal values, Figure 7 is obtained.

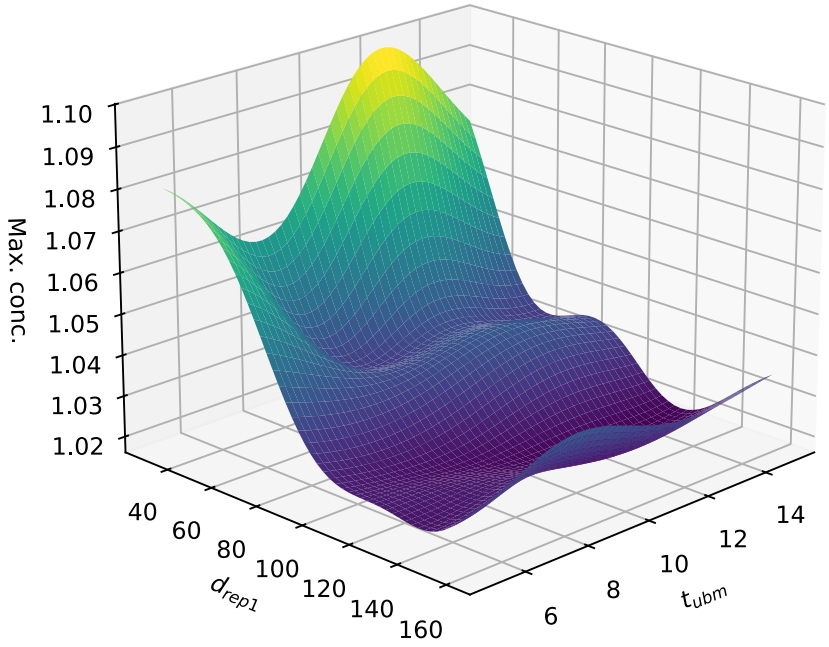


Figure 7. Part of the GPR response surface with the RQ covariance kernel and noiselessness assumption.

Figure 7 shows that even with most of the parameters fixed, the remaining two-dimensional slice of the objective space still presents several local minima, unlike the (less accurate) GPR-based RSM obtained by Optimus (see Figure 5 of [24]). Given the multimodality of the objective function, optimizing the vacancy concentration might therefore potentially require Bayesian optimization (BO) techniques [50,51]. BO often consists of a GPR model that is defined across the entire design space and an acquisition function that assigns newly sampled data, balancing design space exploration and exploitation. This optimization heuristic considers the entire design space at once, unlike gradient-based heuristics [52] such as the popular gradient descent (GD) method. Gradient methods are inherently greedy, i.e., they will converge to any local optimum, rather than attempting to optimize globally. Some modifications of GD, such as Adam [43], stochastic GD [53], and particle swarm optimization [54], are only able to partially mitigate this problem by considering various local optima.

The results exhibited in this section show that fixing the noise value is a non-trivial modeling choice, in line with the synthetic results from Section 2.2. These choices need to be taken into consideration when building an informative surrogate model that adequately covers the global design space. This is especially useful when the objective needs to be optimized, but the response surface is (highly) multimodal.

4. Discussion

Some potential research directions that were beyond the scope of this manuscript are highlighted here.

- **More data and design parameters.** The regression models that were evaluated during the benchmark are conditioned on the same data set of 100 designs. It would be valuable to observe the improvement that a hyperparameter study would provide when a larger data set is used. Moreover, the crucial importance of the processing conditions and material properties during electronic package manufacturing should be considered. Additional relevant design parameters related to fabrication processes in practice would be able to expand and refine any GPR-based response surface models.
- **More thorough sensitivity analysis.** The objective projection plot per design parameter in Figure 6 already yields some qualitative indications on the design parameter sensitivities. However, Sobol' sensitivity analysis [55] would be able to provide additional insight into the effect of the selected design parameters. Unfortunately, the provided simulation data was drawn from the design space by means of the Latin hypercube sampling technique. These data points are incompatible with the Sobol' sensitivity index calculation method that is most commonly used. Using a Sobol' sampling technique can be taken into account during future data collection.
- **Model parameter estimation benchmark.** GPR determines suitable covariance kernel parameters through the gradient-based optimization of the likelihood landscape. The determination of model parameters through variography is mentioned as an alternative option in the Introduction, giving rise to the traditional kriging method rather than the machine learning-based GPR. This method is not favorable for this design problem, however, due to the high dimensionality (7) compared to the low number of available design samples, increasing the uncertainty in the distribution derived from an empirical variogram. Nonetheless, it could be interesting to compare empirical variography with likelihood optimization as alternative methods of obtaining model parameters.
- **More considered hyperparameters.** The benchmark presented in this manuscript covered the results from a few popular covariance kernels for GPR models and fixing a model's noise parameter. A more expansive benchmark with additional toolbox

depth containing more kernel types and likelihood space restrictions, e.g., fixing the length scale of a covariance function.

- **Evaluating predictive variance.** Comparing the regression-predictive variance to the variance due to aleatoric uncertainty is a highly important and relevant topic. One possibility of comparing different RPDs against a ground truth, which takes the regression-predictive variance into account, is by considering the so-called test likelihood [56]. This method compares the (log) likelihood values—the function being maximized in Equation (8)—at test design vectors directly, favoring the RPD with the higher likelihood.
- **Physical interpretation.** The results from this work are data-driven. Therefore, the physical insights that can be concluded from this research are limited. While not within the scope of this manuscript, attempting to link the data-driven results to physical understanding is a highly interesting and important topic for future research. The findings that have been uncovered by our research will catalyze the observation of, e.g., the multimodal and covariational behavior of the atomic vacancy concentration in a real-world experimental setting.

5. Conclusions

Solder bump failure due to electromigration is a relevant and common problem in micro-electronic packaging. A high atomic vacancy concentration is one of the main culprits. This concentration is computationally expensive to simulate and highly physically complex in its design parameters, which demands a data-driven approach to create a cheap surrogate model. This paper presents GPR as a crucial tool in building RSMs for design problems with an expensive objective. It is within this context that we undertake a critical examination of the RSM capabilities of Optimus, which lacks the flexibility to let the user choose their own covariance function and model parameter fixtures. Results obtained from both a synthetic objective and the vacancy concentration objective from the solder bump design problem demonstrate that a robust open-source workflow for hyperparameter modeling is both necessary and feasible because this practice is cheap compared to obtaining atomic vacancy concentration data. Concretely, relying on a GPR that uses the RBF covariance kernel can severely affect the quality of the surrogate model given unseen data, as demonstrated with the 44.2% test error with a restricted noise parameter (31.1% in the case of the Optimus model with unrestricted noise parameter values) for the concentration objective. Considering other covariance classes will lead to better generalizability of the surrogate (8.11% test error).

It takes Optimus only 25 s to provide a 10-fold cross-validated GPR model with the RBF covariance. In contrast, it takes 150 s total to perform two 10-fold cross-validations on the various GPR models with covariance type and inferred noise level considerations. However, both computational expenditures pale in comparison to the 13 h to collect the atomic vacancy objective data. Our workflow constitutes a computational expense ratio of only 0.31%, and will therefore take a negligible amount of time compared to obtaining the expensive DoE. As shown from the benchmarking results, validating the hyperparameters of a GPR model is a key step in the procedure towards responsible and robust surrogate modeling. This fact is only exacerbated regarding Bayesian data-driven design optimization at large, and the results shown in this paper lead to the justification of previous work on this topic [35].

Author Contributions: Conceptualization, L.G.; methodology, L.G.; software, L.G.; validation, L.G. and W.D.v.D.; formal analysis, L.G.; investigation, L.G.; resources, W.D.v.D. and G.Z.; data curation, W.D.v.D.; writing—original draft preparation, L.G.; writing—review and editing, L.G., W.D.v.D. and G.Z.; visualization, L.G. and W.D.v.D.; supervision, W.D.v.D. and G.Z.; project administration, W.D.v.D. and G.Z.; funding acquisition, G.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the ECSEL Joint Undertaking (JU) under grant number 826417. The JU receives support from the European Union's Horizon 2020 research and innovation program and Germany, Austria, Spain, Finland, Hungary, Slovakia, The Netherlands, and Switzerland.

Data Availability Statement: The data and the results related to the GPR models originating from the GPyTorch can be found on GitHub: https://github.com/llguo95/solder_bump_design (accessed on 6 July 2026).

Acknowledgments: The authors would like to thank Leiming Du for providing the finite element results and the related visualization resources. Furthermore, the authors would like to thank Jiaxiang Yi for the valuable discussions leading up to the finalization of this work.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

DoE	Design of experiments
RSM	Response surface (surrogate) modeling
ANN	Artificial neural network
GP	Gaussian process
BO	Bayesian optimization
GPR	Gaussian process regression
RBF	Radial basis function
RQ	Rational quadratic
PPD	Posterior predictive distribution
MLE	Maximum likelihood estimate
RPD	Regression-predictive distribution
RelMSE	Relative mean squared error
CPU	Central processing unit
LHS	Latin hypercube sampling
ARD	Automatic relevance determination

Appendix A. Latin Hypercube Sampling

Let D be the number of design parameters, denoted $x_1, \ldots, x_D$. For any $j \in \{1, \ldots, D\}$, assume that x_j is associated with a lower bound l_j and an upper bound u_j. The corresponding design space is then defined as $\mathcal{X} := [l_1, u_1] \times \cdots \times [l_D, u_D] \subset \mathbb{R}^D$, where the positive integer D is also referred to as the dimensionality of the design space. The single-output objective function in question is denoted as $f : \mathcal{X} \to \mathbb{R}$.

When keeping computational expense in mind, it is essential to know how f responds to the various designs that populate $\mathcal{X}$. As an immediate consequence, the potentially limited number of samples on which to evaluate f should be as spread out and unclustered as possible. Exploration of the design space by means of sampling $\{\mathbf{x}_1, \mathbf{x}_2, \ldots\} \subset \mathcal{X}$ and evaluating $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots\} \subset \mathbb{R}$ therefore constitutes a key technique within the data-driven methodology.

One experimental design approach involves randomly permuting a stratified sampling method across $\mathcal{X}$, called Latin hypercube sampling (LHS) [57]. This works as follows:

1. Given N desired samples on $\mathcal{X}$, partition $[l_j, u_j]$ into N intervals of equal length $I_{j,1}, I_{j,2}, \ldots, I_{j,N}$ for every $j \in \{1, \ldots, D\}$.
2. Select D permutations $\rho_1, \rho_2, \ldots, \rho_D$ on N elements. This is usually done by drawing from a uniform random distribution.
3. For any $i \in \{1, \ldots, N\}$, select one sample $\mathbf{x}_i$ from $I_{1,\rho_1(i)} \times I_{2,\rho_2(i)} \times \cdots \times I_{D,\rho_D(i)}$. This is usually also done at uniformly random.
4. The LHS samples consist of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$.

The main benefit compared to a uniform random sampling approach lies in a more accurate approximation of statistical estimators. For computationally expensive experimental design, however, the main reason for considering LHS is to circumvent the periodicity of the standard grid sampling method. For a demonstration of LHS samples, please refer to Figure A1.

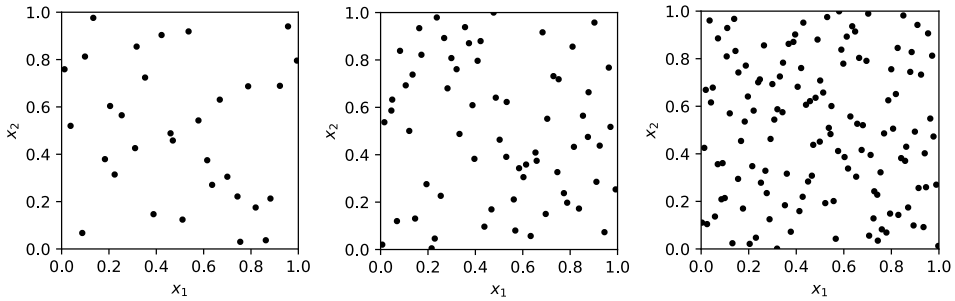


Figure A1. Latin hypercube sampling on a unit square with 32, 64, and 128 samples.

Appendix B. Covariance Kernel Functions

Throughout this manuscript, covariance functions κ, also called (covariance) kernels, play a pivotal role in the behavior of GPs. This section aims to provide additional context concerning covariance functions for a GP regression practitioner to consider when model selection is of importance.

Appendix B.1. Relevance Determination

Let $\mathbf{u}, \mathbf{v} \in [0,1]^D$. A simplified version of the covariance kernels defined in Equation (2) can be provided by assuming $\lambda := \lambda_1 = \ldots = \lambda_D$, so that

$$\begin{aligned}
\kappa_{\mathrm{RBF},\boldsymbol{\theta}_{\mathrm{RBF}}}(\mathbf{u},\mathbf{v}) &:= c \cdot \exp\left(-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\lambda^2}\right) + s^2\delta(\mathbf{u}-\mathbf{v}),\\
\kappa_{\mathrm{Mat},\boldsymbol{\theta}_{\mathrm{Mat}}}(\mathbf{u},\mathbf{v}) &:= c \cdot \left(1 + \frac{\sqrt{5}\|\mathbf{u}-\mathbf{v}\|}{\lambda} + \frac{5\|\mathbf{u}-\mathbf{v}\|^2}{3\lambda^2}\right)\exp\left(-\frac{\sqrt{5}\|\mathbf{u}-\mathbf{v}\|}{\lambda}\right) + s^2\delta(\mathbf{u}-\mathbf{v}),\\
\kappa_{\mathrm{RQ},\boldsymbol{\theta}_{\mathrm{RQ}}}(\mathbf{u},\mathbf{v}) &:= c \cdot \left(1 + \frac{\|\mathbf{u}-\mathbf{v}\|^2)}{2\alpha\lambda^2}\right)^{-\alpha} + s^2\delta(\mathbf{u}-\mathbf{v}).
\end{aligned} \tag{A1}$$

Accordingly, the length scale vector reduces to, e.g., $\boldsymbol{\theta}_{\mathrm{RBF}} := (c, \lambda, s^2)^\top$.

In other words, the kernels from Equation (A1) accommodate each design variable with one uniform length scale λ, rather than a unique parameter for each dimension, which is called *automatic relevance determination* (ARD). The covariance kernels introduced in this manuscript in Equation (2) assume ARD. Considering a uniform length scale can significantly reduce the number of model parameters for consideration when solving

the maximum likelihood problem (Equation (8)), especially if the design dimensionality D is high.

Appendix B.2. Matérn Covariance Family

The ARD (Equation (2)) and uniform Matérn covariance kernels (Equation (A1)) are part of a larger family of Matérn-ν kernels $\kappa^{(\nu)}_{\mathrm{Mat},\boldsymbol{\theta}_{\mathrm{Mat}}}$, which introduce an additional parameter $\nu > 0$. Depending on different values of ν, the kernel has varying smoothness properties [12], where larger values of ν indicate smoother posterior trajectories. The value of $\nu = 5/2$ is consistently used throughout the manuscript, which is the standard value used in GPyTorch and given by the earlier formulas. Other values for ν are $1/2$ and $3/2$, of which the uniform length scale versions are given by:

$$\begin{aligned}\kappa^{(1/2)}_{\mathrm{Mat},\boldsymbol{\theta}_{\mathrm{Mat}}}(\mathbf{u},\mathbf{v}) &:= c \cdot \exp\left(-\frac{\|\mathbf{u}-\mathbf{v}\|}{\lambda}\right) + s^2\delta(\mathbf{u}-\mathbf{v}),\\ \kappa^{(3/2)}_{\mathrm{Mat},\boldsymbol{\theta}_{\mathrm{Mat}}}(\mathbf{u},\mathbf{v}) &:= c \cdot \left(1+\frac{\sqrt{3}\|\mathbf{u}-\mathbf{v}\|}{\lambda}\right)\exp\left(-\frac{\sqrt{3}\|\mathbf{u}-\mathbf{v}\|}{\lambda}\right) + s^2\delta(\mathbf{u}-\mathbf{v}).\end{aligned} \tag{A2}$$

Appendix B.3. Non-Stationary Kernels

For any $\mathbf{u},\mathbf{v} \in [0,1]^D$, the kernel functions mentioned throughout the manuscript depend on $\mathbf{u}-\mathbf{v}$, rather than any other combination of $\mathbf{u}$ and $\mathbf{v}$. These kernels are called *stationary*. Depending on the use case, it is worthwhile to broaden the scope to include non-stationary kernels, such as the linear (or dot product) kernel:

$$\kappa_{\mathrm{Lin},\boldsymbol{\theta}_{\mathrm{Lin}}}(\mathbf{u},\mathbf{v}) = c \cdot \mathbf{u}^\top \boldsymbol{\Sigma}\mathbf{v} + s^2\delta(\mathbf{u}-\mathbf{v}), \tag{A3}$$

which is—as the name suggests—used to model linear relationships, similar to a simple least-squares linear regression model. Here, $\boldsymbol{\Sigma} = (\mathrm{Cov}(\mathbf{u}_i,\mathbf{v}_j))_{i,j=1,\ldots,N}$ is any (prior) covariance matrix, populated by the prior covariances between any two components of $\mathbf{u}$ and $\mathbf{v}$.

References

1. Marco, A.; Berkenkamp, F.; Hennig, P.; Schoellig, A.P.; Krause, A.; Schaal, S.; Trimpe, S. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization. In *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2017; pp. 1557–1563.
2. Babaee, H.; Perdikaris, P.; Chryssostomidis, C.; Karniadakis, G. Multi-fidelity modelling of mixed convection based on experimental correlations and numerical simulations. *J. Fluid Mech.* **2016**, *809*, 895–917. [CrossRef]
3. Babaee, H.; Acharya, S.; Wan, X. Optimization of forcing parameters of film cooling effectiveness. *J. Turbomach.* **2014**, *136*, 061016.
4. Udrescu, S.M.; Tegmark, M. AI Feynman: A physics-inspired method for symbolic regression. *Sci. Adv.* **2020**, *6*, eaay2631. [CrossRef] [PubMed]
5. Min, A.T.W.; Sagarna, R.; Gupta, A.; Ong, Y.S.; Goh, C.K. Knowledge transfer through machine learning in aircraft design. *IEEE Comput. Intell. Mag.* **2017**, *12*, 48–60. [CrossRef]
6. Aage, N.; Andreassen, E.; Lazarov, B.S.; Sigmund, O. Giga-voxel computational morphogenesis for structural design. *Nature* **2017**, *550*, 84–86. [CrossRef] [PubMed]
7. Box, G.E.; Draper, N.R. A basis for the selection of a response surface design. *J. Am. Stat. Assoc.* **1959**, *54*, 622–654. [CrossRef]
8. Hoerl, A.E.; Kennard, R.W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **1970**, *12*, 55–67. [CrossRef]
9. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1996**, *58*, 267–288. [CrossRef]
10. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
11. Jospin, L.V.; Laga, H.; Boussaid, F.; Buntine, W.; Bennamoun, M. Hands-on Bayesian neural networks—A tutorial for deep learning users. *IEEE Comput. Intell. Mag.* **2022**, *17*, 29–48. [CrossRef]
12. Williams, C.K.; Rasmussen, C.E. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006; Volume 2.

13. Krige, D.G. A Statistical Approach to Some Mine Valuation and Allied Problems on the Witwatersrand: By DG Krige. Ph.D. Thesis, University of the Witwatersrand, Johannesburg, South Africa, 1951.
14. Christianson, R.B.; Pollyea, R.M.; Gramacy, R.B. Traditional kriging versus modern Gaussian processes for large-scale mining data. *Stat. Anal. Data Min. ASA Data Sci. J.* **2023**, *16*, 488–506. [CrossRef]
15. Aigrain, S.; Foreman-Mackey, D. Gaussian process regression for astronomical time series. *Annu. Rev. Astron. Astrophys.* **2023**, *61*, 329–371. [CrossRef]
16. Jin, B.; Xu, X. Gaussian process regression based silver price forecasts. *J. Uncertain Syst.* **2024**, *17*, 2450013. [CrossRef]
17. Dang, W.; Liao, S.; Yang, B.; Yin, Z.; Liu, M.; Yin, L.; Zheng, W. An encoder-decoder fusion battery life prediction method based on Gaussian process regression and improvement. *J. Energy Storage* **2023**, *59*, 106469. [CrossRef]
18. Tilgner, R. Physics of failure for interconnect structures: An essay. *Microsyst. Technol.* **2009**, *15*, 129–138.
19. Jen, M.H.R.; Liu, L.C.; Lai, Y.S. Electromigration test on void formation and failure mechanism of FCBGA lead-free solder joints. *IEEE Trans. Components Packag. Technol.* **2009**, *32*, 79–88. [CrossRef]
20. Sofiane, S.; Sampathkumar, G.; Renan, L. Simulation assessment of solder joint reliability for fully assembled printed circuit boards. *Microelectron. Reliab.* **2024**, *153*, 115320. [CrossRef]
21. Ma, H.; Gou, M.; Tian, X.; Tan, W.; Liang, H. Reliability simulation of IGBT module with different solders based on the finite element method. *Metals* **2024**, *14*, 1141. [CrossRef]
22. Van Driel, W.; Zhang, G.; Janssen, J.; Ernst, L. Response surface modeling for nonlinear packaging stresses. *J. Electron. Packag.* **2003**, *125*, 490–497. [CrossRef]
23. Wymysłowski, A.; Van Driel, W.; Van De Peer, J.; Tzannetakis, N.; Zhang, G. Advanced numerical prototyping methods in modern engineering applications–Optimisation for micro-electronic package reliability. *Microelectron. Reliab.* **2007**, *47*, 280–289. [CrossRef]
24. Du, L.; Deng, S.; Cui, Z.; Poelma, R.; Beelen-Hendrikx, C.; Zhang, K. Multi-parameters optimization for electromigration in WLCSP solder bumps. In *Proceedings of the 2024 25th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE)*; IEEE: New York, NY, USA, 2024; pp. 1–4.
25. de Jong, S.; Ghezeljehmeidan, A.; van Driel, W. Solder joint reliability predictions using physics-informed machine learning. *Microelectron. Reliab.* **2025**, *172*, 115797. [CrossRef]
26. About our Company | Noesis Solutions | Noesis Solutions. Available online: https://www.noesissolutions.com/about-noesis-solutions (accessed on 17 January 2021).
27. Bogaerts, L.; Faes, M.; Bergen, J.; Cloots, J.; Vasiliauskaite, E.; Vogeler, F.; Moens, D. Influence of thermo-mechanical loads on the lifetime of plastic inserts for injection moulds produced via additive manufacturing. *Procedia CIRP* **2021**, *96*, 109–114. [CrossRef]
28. Kadziela, B.; Manka, M.; Uhl, T.; Toso, A. Validation and optimization of the leaf spring multibody numerical model. *Arch. Appl. Mech.* **2015**, *85*, 1899–1914. [CrossRef]
29. Guo, L.; Kansara, H.; Khosroshahi, S.F.; Zhang, G.; Tan, W. Multi-fidelity Bayesian data-driven design of energy absorbing spinodoid cellular structures. *Comput. Struct.* **2026**, *323*, 108155. [CrossRef]
30. Kansara, H.; Khosroshahi, S.F.; Guo, L.; Bessa, M.A.; Tan, W. Multi-objective Bayesian optimisation of spinodoid cellular structures for crush energy absorption. *Comput. Methods Appl. Mech. Eng.* **2025**, *440*, 117890. [CrossRef]
31. Gardner, J.; Pleiss, G.; Weinberger, K.Q.; Bindel, D.; Wilson, A.G. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation: San Diego, CA, USA, 2018; Volume 31.
32. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation: San Diego, CA, USA, 2019; Volume 32.
33. Matthews, A.G.d.G.; Van Der Wilk, M.; Nickson, T.; Fujii, K.; Boukouvalas, A.; León-Villagrá, P.; Ghahramani, Z.; Hensman, J. GPflow: A Gaussian process library using TensorFlow. *J. Mach. Learn. Res.* **2017**, *18*, 1–6.
34. TensorFlow Developers. *TensorFlow*; Zenodo: Geneva, Switzerland, 2022.
35. Guo, L.; Inamdar, A.; Van Driel, W.D.; Zhang, G. Efficient and Adaptive Bayesian Data-Driven Design of Reliable Solder Joints for Micro-electronic Devices. *Appl. Math. Model.* **2025**, *154*, 116645.
36. Genton, M.G. Classes of kernels for machine learning: A statistics perspective. *J. Mach. Learn. Res.* **2001**, *2*, 299–312.
37. Schulz, E.; Speekenbrink, M.; Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **2018**, *85*, 1–16. [CrossRef]
38. Liu, M.; Chowdhary, G.; Da Silva, B.C.; Liu, S.Y.; How, J.P. Gaussian processes for learning and control: A tutorial with examples. *IEEE Control Syst. Mag.* **2018**, *38*, 53–86. [CrossRef]
39. Minasny, B.; McBratney, A.B. The Matérn function as a general model for soil variograms. *Geoderma* **2005**, *128*, 192–207. [CrossRef]
40. Kocijan, J. *Modelling and Control of Dynamic Systems Using Gaussian Process Models*; Springer: Berlin/Heidelberg, Germany, 2016.

41. Holt, W.; Nguyen, D. Essential Aspects to Bayesian Data Imputation. In *An Invitation to Undergraduate Research in Risk Management. Foundations for Undergraduate Research in Mathematics;* Birkhäuser: Cham, Switzerland, 2023.
42. Duvenaud, D. Automatic Model Construction with Gaussian Processes. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 2014.
43. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980
44. Liang, Y.C.; Tsao, W.; Chen, C.; Yao, D.J.; Huang, A.T.; Lai, Y.S. Influence of Cu column under-bump-metallizations on current crowding and Joule heating effects of electromigration in flip-chip solder joints. *J. Appl. Phys.* **2012**, *111*, 043705. [CrossRef]
45. Hau-Riege, C.; Keser, B.; Alvarado, R.; Syed, A.; Yau, Y.; Bezuk, S.; Caffey, K. The impact of different under bump metallurgies and redistribution layers on the electromigration of solder balls for wafer-level packaging. In *Proceedings of the 2014 IEEE 64th Electronic Components and Technology Conference (ECTC)*; IEEE: New York, NY, USA, 2014; pp. 1173–1178.
46. Niu, L.; Tian, X.; Jia, F. The study on electromigration of solder joints under thermal cycling load. In *Proceedings of the 2021 22nd International Conference on Electronic Packaging Technology (ICEPT)*; IEEE: New York, NY, USA, 2021; pp. 1–5.
47. Ueda, D.; Yamamoto, A.; Takashima, T.; Onoda, N.; Noda, S.; Kashiwagi, S.; Morisaki, T.; Honjo, T.; Shimazaki, A.; Miki, Y. Training, validation, and test of deep learning models for classification of receptor expressions in breast cancers from mammograms. *JCO Precis. Oncol.* **2021**, *5*, 543–551. [CrossRef] [PubMed]
48. Khalil, M.; McGough, A.S.; Pourmirza, Z.; Pazhoohesh, M.; Walker, S. Machine Learning, Deep Learning and Statistical Analysis for forecasting building energy consumption—A systematic review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105287. [CrossRef]
49. Sharma, A.; Mukhopadhyay, T.; Rangappa, S.M.; Siengchin, S.; Kushvaha, V. Advances in computational intelligence of polymer composite materials: Machine learning assisted modeling, analysis and design. *Arch. Comput. Methods Eng.* **2022**, *29*, 3341–3385. [CrossRef]
50. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* **2015**, *104*, 148–175. [CrossRef]
51. Garnett, R. *Bayesian Optimization*; Cambridge University Press: Cambridge, UK, 2023.
52. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
53. Shamir, O.; Zhang, T. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2013; pp. 71–79.
54. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of the ICNN'95-International Conference on Neural Networks*; IEEE: New York, NY, USA, 1995; Volume 4, pp. 1942–1948.
55. Saltelli, A.; Annoni, P.; Azzini, I.; Campolongo, F.; Ratto, M.; Tarantola, S. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Comput. Phys. Commun.* **2010**, *181*, 259–270. [CrossRef]
56. Quinonero-Candela, J.; Rasmussen, C.E.; Sinz, F.; Bousquet, O.; Schölkopf, B. Evaluating predictive uncertainty challenge. In *Proceedings of the Machine Learning Challenges Workshop*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 1–27.
57. McKay, M.D.; Beckman, R.J.; Conover, W.J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **2000**, *42*, 55–61. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.