Title

Mathematical nuances of Gaussian process-driven autonomous experimentation

Journal

MRS Bulletin, 48(2)

ISSN

0883-7694

Authors

Noack, Marcus M

Reyes, Kristofer G

Publication Date

2023-02-01

DOI

10.1557/s43577-023-00478-8

Mathematical nuances of Gaussian process-driven autonomous experimentation

Marcus M. Noack* and Kristofer G. Reyes

The fields of machine learning (ML) and artificial intelligence (AI) have transformed almost every aspect of science and engineering. The excitement for AI/ML methods is in large part due to their perceived novelty, as compared to traditional methods of statistics, computation, and applied mathematics. But clearly, all methods in ML have their foundations in mathematical theories, such as function approximation, uncertainty quantification, and function optimization. Autonomous experimentation is no exception; it is often formulated as a chain of off-the-shelf tools, organized in a closed loop, without emphasis on the intricacies of each algorithm involved. The uncomfortable truth is that the success of any ML endeavor, and this includes autonomous experimentation, strongly depends on the sophistication of the underlying mathematical methods and software that have to allow for enough flexibility to consider functions that are in agreement with particular physical theories. We have observed that standard off-the-shelf tools, used by many in the applied ML community, often hide the underlying complexities and therefore perform poorly. In this paper, we want to give a perspective on the intricate connections between mathematics and ML, with a focus on Gaussian process-driven autonomous experimentation. Although the Gaussian process is a powerful mathematical concept, it has to be implemented and customized correctly for optimal performance. We present several simple toy problems to explore these nuances and highlight the importance of mathematical and statistical rigor in autonomous experimentation and ML. One key takeaway is that ML is not, as many had hoped, a set of agnostic plug-and-play solvers for everyday scientific problems, but instead needs expertise and mastery to be applied successfully.

Introduction

Machine learning (ML) and artificial intelligence (AI) have transformed how problems involving model creation and decision-making from data are approached in all areas of science and engineering. Examples are wide ranging and include weather forecasts,[1,2] protein folding,[3,4] natural language processing,[5] image recognition,[6,7] and autonomous experimentation.[8–12] Some successes, for instance, IBM's Watson and AlphaGo, reached international fame. When Watson famously won the popular game Jeopardy in 2011 against two of the best human players, the general opinion was that Watson would soon be able to answer any medical or scientific questions better than any human; the reality turned out to be very different and mathematics can explain why. Contrary to what was perceived outside IBM's offices, Watson's architecture was specifically customized to win in a game, such as Jeopardy, with minimal generalizations in place. A current example is large language models,[13] such as GPT-3[14] or Turing-NLG,[15] which are tailored for natural language processing and can deliver amazing results for some tasks, but are also easily tricked into wrong and overconfident answers.[16] Tuned for a specific task, they do very well. At the other end of the spectrum of generalizability, largely agnostic ML software tools are being developed and distributed (Scikit-learn, PyTorch, TensorFlow) in order to give more people access to the power of ML. Although this generalization is laudable and necessary, it can also lead to user errors and dissatisfaction with the results. There seems to be a natural tradeoff between the power of AI and ML and its generalization potential. Many of the successes of

Marcus M. Noack, Lawrence Berkeley National Laboratory, Applied Mathematics and Computational Research Division, Berkeley, USA; MarcusNoack@lbl.gov
Kristofer G. Reyes, Department of Materials Design and Innovation, University at Buffalo, The State University of New York, Buffalo, USA; kreyes3@buffalo.edu
*Corresponding author

doi:10.1557/s43577-023-00478-8

ML can be attributed to the wide availability of off-the-shelf software tools. However, this availability and the user-friendliness of those tools can lead to a one-concept-fits-all attitude, leading to poor performance of the algorithms in nonstandard scenarios. To understand this discrepancy, we have to dive a bit deeper into the mathematics of ML.

ML can broadly be divided into supervised and unsupervised methods. Supervised learning uses "labeled" data $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}$ to find some function $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ that can approximate unobserved pairs $(\mathbf{x}_i, \mathbf{y}_i)$. $\mathbb{R}^n$ is often called the input space or parameter space and here denoted by $\mathcal{X}$; $\mathbb{R}^m$ is the output space. Unsupervised learning does not use labels and attempts to find structural (geometric or topological) information about a data set $\mathcal{D} = \{\mathbf{x}_i\}$. Throughout this article, we focus on autonomous experimentation, which is often classified as "active learning" and is part of supervised ML; even so, many of our takeaways are valid for unsupervised learning as well. Supervised learning can be characterized by two main building blocks, the definition of a function space—sometimes called the hypothesis space—$\mathcal{F}$ containing all conceivable model functions, and the selection of an optimality condition, most often some misfit, that is maximized (or minimized) to find a candidate solution—a particular element of the function space—which is called the training in ML. Often, neither step is getting the attention it deserves. Common mistakes are to use ML tools that span a function space that does not contain functions with the desired behavior based on physics or practitioner intuition, or to combine solutions in an ensemble even though their function spaces are disjoint. A quick note on the diversity of ML. Neural networks (NNs) have been so prominent in the literature and media that one could equate them with ML; however, kernels,[17] especially in combination with Bayesian methods, provide a very powerful and flexible framework for learning which, in small data regimes, outperforms NNs. Even so, mathematically it can be shown that all different methods are just instances of a more fundamental framework.[18,19]

As an example, we want to have a look at kernel ridge regression (KRR),[20–22] where the underlying function space is a, so-called, reproducing kernel Hilbert space (RKHS)[23] $\mathcal{H} = \{f : f(\mathbf{x}) = \sum_i^N \alpha_i k(\mathbf{x}, \mathbf{x}_i; h) \forall \mathbf{x} \in \mathcal{X}\}$. For a more accessible notion of the RKHS, we can understand it as a set of functions that are all defined by a weighted linear combination of kernels. In simple terms, kernels are functions that get two points of the parameter space as input and return a measure of similarity of the function itself. Stationary kernels only depend on the distance between the two input points; most often the similarity of the function value decreases as the distance increases. Nonstationary kernels have no such restrictions and can encode complicated rules about how similarity between inputs behaves across the domain. Alternatively, but equivalently, we can view kernels as basis functions defined on the input space, centered at given locations $\mathbf{x}_i$. RKHSs have gained popularity in recent years due to their importance for many machine learning methods, such as Gaussian processes (GPs),[24] support vector machines (SVMs),[25] and kernel PCA (principal component analysis).[26] It is not uncommon to see practitioners using a GP posterior mean and the surrogate computed by KRR to create ensemble models; despite the fact that, for the same kernel and the quadratic loss function, these models will coincide, which instills unsupported confidence in the ensemble. Ensembles of ML solutions carry the risk of bias if the underlying function spaces are not compatible (e.g., disjoint). The most commonly used optimality condition for KRR is to place a measure on the difference between predicted and given values $\hat{y}_i \in \hat{\mathcal{D}}$, which is the test data set. In the standard literature, not much effort is spent on different kernels for KRR; instead the radial basis function (RBF) kernel $k(\mathbf{x}_1, \mathbf{x}_2) = \sigma_s \exp(-\frac{||\mathbf{x}_1 - \mathbf{x}_1||_2^2}{2l^2})$ or other kernels of the Matérn class are oftentimes used without justification;[17,27] the RBF kernel gives rise to a function space that only contains functions of infinite differentiability (very smooth functions)—a property often not supported by the data or the underlying physics (see **Figure 1**). Similarly, other Matérn kernels—and for that matter all other kernels—have well-defined differentiability properties that directly influence the model. l and σ_s are free parameters, examples of so-called hyperparameters that can be interpreted as a global length scale and a signal variance, respectively. Their global validity is often unsupported and, when enforced, can lead to poor performance of kernel methods. These challenges directly affect how well ML can control data acquisition without human supervision.

As instruments and detectors are accelerating their peak data acquisition rates and the increasing complexity of scientific questions give rise to larger and higher-dimensional parameter spaces, it becomes infeasible for the human brain to make optimal decisions about experimental design. Autonomous experimentation (AE) describes the ability of an instrument and algorithm to decide what measurements should be performed next, ideally without the need for human interference; it is a multifaceted field that needs expertise in instrument science, robotics, computer science, and ML. The role of ML is twofold: First, as raw data—images, films, spectra—leave the instrument, they have to be analyzed and dimensionality reduced. Although many classical methods are successfully being used, ML is increasingly considered a viable option. Second, intelligent autonomous decision-making is performed based on all collected and analyzed data. This decision-making is commonly categorized as active learning, which, as aforementioned, is a kind of supervised learning in which the algorithm can choose its own training data. If no offline training data are available, stochastic process-driven uncertainty quantification (UQ) is often used in the form of Gaussian (stochastic) processes (GPs).[28–31] The principle of a GP is simple; given a set of noisy function evaluations, we define a normal probability distribution that explains the data and can be conditioned on observations to yield a probabilistic view of the model function in unobserved regions (see the next

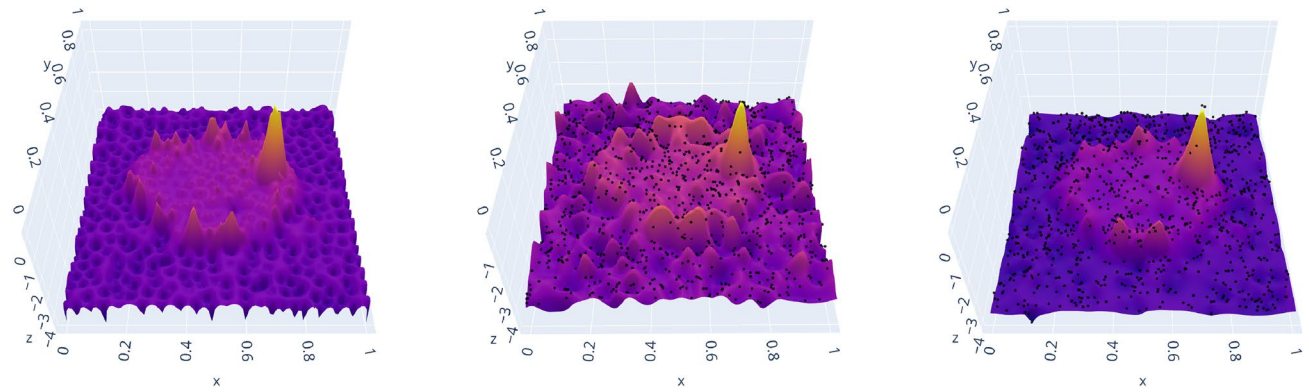


Figure 1. Learning a two-dimensional map from a set of 1000 observations with an emphasis on kernel designs. The model shown arose from the evaporation of a nanoparticle-containing solution.[9] The latent function is inherently non-smooth (non-differentiable). This has to be accounted for when defining a kernel for the Gaussian process (GP)-driven autonomous experiment. The ground truth is displayed on the left. In the center, we see the posterior mean of a GP using the squared exponential radial basis function kernel. In this case, the model has to be smooth, which leads to artifacts in the model. On the right-hand side, the posterior mean using an exponential kernel is shown. The exponential kernel is rarely an appropriate choice, except when the latent function is non-differentiable, which happens to be the case for many mapping experiments in the materials sciences. Here, the use of the exponential kernel leads to a more accurate model prediction.

section). A well-tuned GP can quantify the uncertainty of the model function across the domain, allowing for intelligent decision-making and therefore autonomous control. However, the control is only effective if the GP is set up correctly, meaning the right function space is considered and the optimization is sufficiently well posed. AE is a particularly challenging ML problem because for new experiments often no offline training data are available and decisions have to be made on the fly as data are collected. It, therefore, emphasizes the need for particular rigor of the underlying mathematics; black-box applications of off-the-shelf ML tools will often show poor performance, which manifests itself through the overestimation of uncertainties that severely limit the efficiency of the AE. Autonomous experimentation plays an important role in the materials sciences due to the fact that scientific questions are often posed as finding one or more materials properties as a function of some parameters. Examples are crystal sizes as a function of an annealing history, x-ray scattering mapping, or point-wise evaluation of spectra originating from neutron scattering.

The aforementioned emphasis in ML on the choice of a sensible function space and an appropriate optimality condition is the main reason why common off-the-shelf tools perform suboptimally; not always because they do not possess the ability for sufficiently flexible definitions, but they trade flexibility for user-friendliness, which is often preferred.

The main objective of this article is to show what gains can be made in Gaussian process-driven AE if we open up the black box that is ML and spend some time evaluating and customizing the underlying mathematics and statistics. After a short excursion into some theory, the remainder of this article discovers, by example, the shortcomings of some off-the-shelf applications of ML tools for AE and how they can be avoided by simple adjustments of the core algorithm.

Some theory of Gaussian processes and related autonomous experimentation

To maximize the value of the tests in the next section, we present some minimal but necessary theory in this section. We will start introducing Gaussian processes and then move to the way they affect AE through an acquisition functional.

Gaussian processes

Gaussian processes (GPs) are a type of stochastic process—sometimes called a random field—in which a set of random variables, often thought of as function evaluations $\{f(\mathbf{x}_1), f(\mathbf{x}_2), f(\mathbf{x}_3), f(\mathbf{x}_4), ...)\}$, are jointly normally distributed.[24,32,33] Imagine having information about a function in the form of probabilistic function evaluations and being interested in the best guess of those function evaluations in other places. A GP is based on the idea of defining a normal distribution over the known and unknown function evaluations. Given data $\mathcal{D} = \{\mathbf{x}_i, y_i\}$, a prior probability distribution over functions $f(\mathbf{x})$ can be defined as follows:

$$p(\mathbf{f}) = \frac{1}{\sqrt{(2\pi)^{\dim}|\mathbf{K}|}} \exp\left[-\frac{1}{2}(\mathbf{f}-\boldsymbol{\mu})^T\mathbf{K}^{-1}(\mathbf{f}-\boldsymbol{\mu})\right], \quad 1$$

where $\mathbf{K}$ is the covariance matrix of the data, whose entries are calculated by having the kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ act on the data positions. Kernels can be seen as basis functions that define the model but also compute how covariances behave as we move away from known data points. In this context, we can understand them as a similarity measure that allows us to calculate covariances purely based on data-point locations. $\boldsymbol{\mu}$ is the prior mean vector. We define the likelihood over observations $y(\mathbf{x})$ as

$$p(\mathbf{y}|\mathbf{f}) = \frac{1}{\sqrt{(2\pi)^{\dim}|\mathbf{V}|}} \exp\left[-\frac{1}{2}(\mathbf{y}-\mathbf{f})^T\mathbf{V}^{-1}(\mathbf{y}-\mathbf{f})\right], \quad 2$$

where $\mathbf{V}$ is the matrix of the noise.[11] Our first test ("The role of noise for autonomous experiments" section) will focus on different choices for the matrix $\mathbf{V}$; however, we assume uncorrelated noise that renders $\mathbf{V}$ diagonal. Most literature assumes identically and independently distributed (i.i.d., also homogeneous, homoscedastic, or simply constant) noise, which translates into $\mathbf{V} = \sigma_n^2\mathbf{I}$. Often, σ_n^2 is estimated by the experimenter ad hoc, while others absorb it into the kernel definition and optimize its value. As we will see, it is ideal to estimate the noise during the measurement process, especially for the purpose of AE.

The vast majority of published work about Gaussian processes only utilizes a few well-known standard stationary kernels to compute covariances.[27] The most frequently used kernel is the RBF kernel

$$k(\mathbf{x}_1, \mathbf{x}_2) = \sigma_s^2 \exp\left[-\frac{||\mathbf{x}_1 - \mathbf{x}_2||_2^2}{2l^2}\right], \quad 3$$

where σ_s^2 is the constant signal variance and l is the isotropic length scale.[27] Both signal variance and length scale are hyperparameters (ϕ) of the Gaussian process and can be calculated by solving the optimization problem[24]

$$\begin{aligned}\arg\max_{\phi}\Big(&\log(L(D,\phi)) = \\ &-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}(\phi))(\mathbf{K}(\phi)+\mathbf{V})^{-1}(\mathbf{y}-\boldsymbol{\mu}(\phi)) \\ &-\frac{1}{2}\log(|\mathbf{K}(\phi)+\mathbf{V}|) - \frac{\dim(\mathbf{y})}{2}\log(2\pi)\Big),\end{aligned} \quad 4$$

which can be understood as maximizing the probability that the data would be observed, given a prior probability distribution. Hyperparameters can be seen as free parameters that are part of the kernel and control the quality of the model. Kernel functions can freely be defined to account for ever-increasing model complexity—as long as positive semi-definiteness is maintained. In fact, the real power of Gaussian processes is only revealed by utilizing nonstationary kernels. This is the focus of our second test ("Stationary kernels versus nonstationary kernels for Gaussian processes" section). As kernels become more complex, the number of needed hyperparameters rises and requires advanced optimization procedures; the optimization is often ill-posed and solutions are nonunique. This is the focus of the "Training as a constrained and ill-posed function optimization problem" section.

Given the hyperparameters, we calculate the posterior probability density function given by

$$\begin{aligned}p(f_0|\mathbf{y}) &= \int_{\mathbb{R}^N} p(f_0|\mathbf{f},\mathbf{y})p(\mathbf{f},\mathbf{y})d\mathbf{f} \\ &\propto \mathcal{N}(\boldsymbol{\mu}+\boldsymbol{\kappa}^T(\mathbf{K}+\mathbf{V})^{-1}(\mathbf{y}-\boldsymbol{\mu}), \boldsymbol{\mathcal{K}}-\boldsymbol{\kappa}^T(\mathbf{K}+\mathbf{V})^{-1}\boldsymbol{\kappa}),\end{aligned} \quad 5$$

where $\boldsymbol{\kappa}_i = k(\mathbf{x}_0, \mathbf{x}_i, \phi)$, $\boldsymbol{\mathcal{K}} = k(\mathbf{x}_0, \mathbf{x}_0, \phi)$, and $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j, \phi)$. $\mathbf{x}_0$ is the point at which the Gaussian posterior should be predicted. f_0 is the value of the latent function f at the point $\mathbf{x}_0$. The posterior contains the posterior mean $m(\mathbf{x}_0)$ and the posterior variance $\sigma^2(\mathbf{x}_0)$.

Autonomous experimentation

Having calculated the posterior probability density function (Equation 5), it can now be used to decide where future measurements should take place. For this, a function of the posterior, a so-called acquisition functional—sometimes simply called acquisition function—$f_a(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$, is defined to assign every measurement (point in the domain) a value. Commonly, regions of low uncertainty are assigned a low value and regions of high uncertainty or probability of finding certain desirable characteristics are assigned a high value. There is an overwhelming number of acquisition functionals in the literature. Often new acquisition functionals have to be defined to allow for optimal performance of the AE. Certain acquisition functionals will turn an autonomous experiment into Bayesian optimization.[32,34] Having defined the acquisition functional,[29] we solve

$$\arg\max_{\mathbf{x}\in\mathcal{X}} f_a(\mathbf{x}), \quad 6$$

$$s.t. g_i(\mathbf{x}) < / \leq / = / \geq / > 0, \forall i \in \{1,2,3,4,...\}, \quad 7$$

where the constraints $g_i(\mathbf{x})$ can be used to restrict the search to regions that are of special interest or to protect the instrument from navigating to regions that are inaccessible or unsafe. An additional modification is to estimate or learn a cost function $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and optimize $f_a(\mathbf{x})/c(\mathbf{x}, \mathbf{x}_0)$, where $\mathbf{x}_0$ is the last measurement location. In this case, new measurement suggestions are cost-sensitive. The "Acquisition functionals for optimal measurement suggestions" section focuses on different choices of acquisition functionals for a simple toy example to emphasize its importance for the successful execution of an autonomous experiment.

Case studies

In this section, we present four case studies that were carefully chosen to highlight specific characteristics of GPs and the associated autonomous experiments. Although many data sets do not stem from the materials sciences, the key takeaways always apply to data with similar properties and are agnostic to the field in which the data originated.

The role of noise for autonomous experiments

Reading through most of the available Gaussian process literature, one would be forgiven to assume that i.i.d. noise is fundamental to the GP framework. On the contrary, the GP framework—as one could expect from a Bayesian method—is without any adaptations able to handle non-i.i.d. noise. Non-i.i.d. noise plays an important role in x-ray scattering and neutron scattering applications, among others. In ML, it is common to ignore noise entirely and advanced noise models are virtually unheard of. In this example, we want to show that the consideration and inclusion of non-i.i.d. noise into the model is

indispensable for optimal autonomous experimentation.[11] This is because the sequence of measurements depends on the location of the maxima of the acquisition functional, which commonly depends on the posterior variance. However, the key takeaway here goes beyond AEs and has important implications for all of ML: accurate estimation and inclusion of noise are important for accurate model predictions. Before looking at the result, we would expect that the data acquisition is biased toward regions with high noise in order to reduce total uncertainty. Our experiment was set up using an anisotropic and stationary Matérn kernel with $\nu = 3/2$. The data set is taken from IR spectroscopy[29,35] and the material is organic matter. In this test, we are approximating a scalar intensity on $[0, 1] \times [0, 1]$.

Consider the ground-truth data and the noise model in **Figure 2**. The noise model is usually not known before the experiment, but we define it ad hoc to test how treating noise differently in the GP affects autonomous decision-making. For this example, we are performing standard maximum variance steering, where points are placed where the posterior variance is at its maximum. One would expect a higher point density in places of high measurement noise, but only non-i.i.d. noise, estimated and communicated for each new measurement delivers this behavior. While treating the noise as one hyperparameter, which can be found by solving Equation 4, delivers a satisfactory model, the overall approximation error is larger than for non-i.i.d. noise. Estimating one noise value ad hoc can render the model ineffective because it influences and misleads the hyperparameter optimization, which negatively affects the accuracy of uncertainty estimation.

Stationary kernels versus nonstationary kernels for Gaussian processes

Even more uncommon than non-i.i.d. noise in GPs is the use of nonstationary kernels. In a review by Pilario et al.,[27] we can dissect that around 90% of studies employing kernel methods use the RBF kernel. The number is much higher when considering a broader range of stationary kernels. Stationarity in the kernel means we are assuming that covariances between data points only depend on their distance, not on the points' respective locations (i.e., $k(\mathbf{x}_1, \mathbf{x}_2) = k(|\mathbf{x}_1 - \mathbf{x}_2|)$), a high standard to be met for most modern data sets; this is especially true for the materials sciences where changes of some parameters will have much more impact on the properties of a material in some regions in the parameter space than in other regions. Imagine inorganic crystal growth as a function of an annealing history; clearly, in some temperature regions, the crystal size will react much more strongly to temperature changes than in others. A prime example to see, experience, and understand the importance of nonstationarity in the kernel definition is to use the topography of the United States as a test data set. Although this data set did obviously not originate from an AE or in the materials sciences it has all characteristics we need to illustrate the importance of nonstationarity. Clearly, covariances should behave differently in the mountainous regions of the Rocky Mountains compared to the Great Plains. Although for accuracy of the function approximation, the difference between nonstationarity and stationarity could still be bearable, for autonomous experimentation where the accurate estimation of uncertainty is a deciding factor, nonstationary plays a vital role in the experiment design, as can be seen in **Figure 3**.

For this example, the stationary kernel is given by

$$k_{stat}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_s^2 \left(1 - \frac{\sqrt{3d}}{l}\right) \exp\left[-\frac{\sqrt{3d}}{l}\right], \quad 8$$

$d = (\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{M} (\mathbf{x}_1 - \mathbf{x}_2)$, which is the axially anisotropic Matérn kernel of first-order differentiability, with diagonal $\mathbf{M}$, length scale l, and signal variance σ_s^2. The nonstationary kernel[36] was defined as

$$k_{\text{non}}(\mathbf{x}_1, \mathbf{x}_2) = f(\mathbf{x}_1) f(\mathbf{x}_2) k_{\text{stat}}(\mathbf{x}_1, \mathbf{x}_2), \quad 9$$

where $f(x) = \sum_i^N \alpha_i \beta(\mathbf{x}_i, \mathbf{x}; w)$. α_i are the heights of some radial basis functions

$$\beta(\mathbf{x}_i, \mathbf{x}; w) = \exp\left[-||\mathbf{x}_i - \mathbf{x}|| w^2\right], \quad 10$$

with w being the parameter controlling the width. The term $f(\mathbf{x}_1) f(\mathbf{x}_2)$ can be interpreted as flexible signal variance, which impacts how uncertainties are estimated across the domain. This leads to uncertainties that are much more reflective of the true error compared with the use of the stationary kernel (see Figure 3).

Training as a constrained and ill-posed function optimization problem

One of the major drawbacks of using nonstationary kernels is that they need parametric representations of functions—signal variances, length scales, and so on—which gives rise to many more hyperparameters compared with standard stationary kernels. For the example in the last section, for instance, the kernel definition needed 286 hyperparameters to be found, compared to two for an isotropic RBF kernel. This dependency of the hyperparameter number on the kernel definition shifts the focus of kernel methods to the training (i.e., the optimization of the hyperparameters). In this case study, we want to analyze the characteristics of the solutions of the training for the kernels used in the last example.

The optimization of the hyperparameters is naturally constrained for many kernels due to the subset $(0, \infty]^n \subset \mathbb{R}$ on which the log likelihood is well defined (e.g., signal variances and length scales should never be negative or zero). n, in this case, is the number of hyperparameters. Other constraints are potentially introduced through domain knowledge. When using local optimizers for hyperparameter training, it is the author's recommendation to remove the bounds on the optimization by considering simple transformations (e.g., $\exp(-||\mathbf{x}_1 - \mathbf{x}_2|| l^2)$ instead of $\exp(-||\mathbf{x}_1 - \mathbf{x}_2||/l)$). Transformations such as this should be applied to make the optimization domain closed but unbounded. In **Figure 4**, we see top views of marginal log-likelihood functions. It is apparent that nonstationary kernels give rise to nonuniqueness of solutions

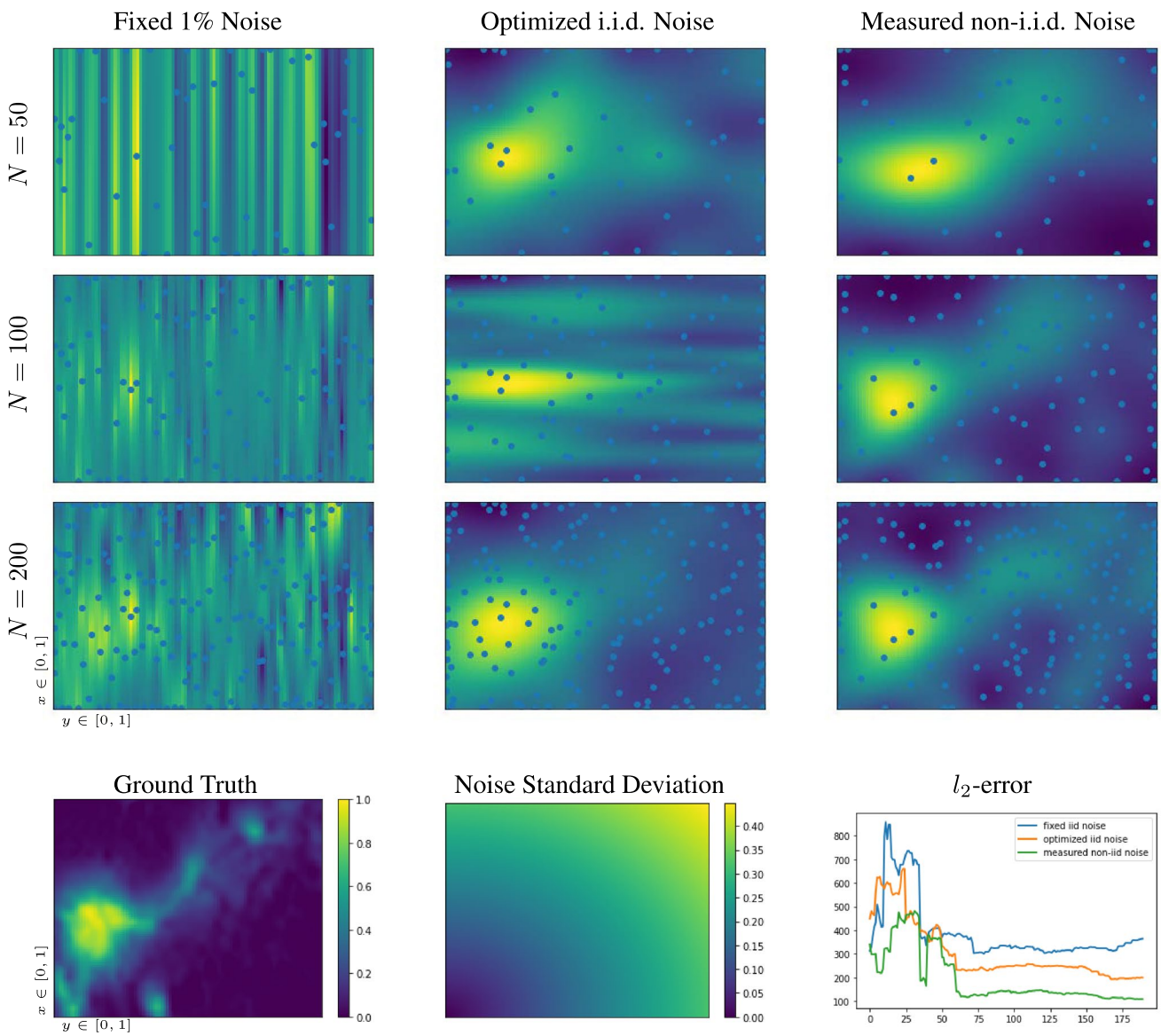


Figure 2. The importance of accurate treatment of noise for the success of an autonomous experiment approximating a scalar spectral intensity on $[0, 1] \times [0, 1]$. The top nine images show the progression of the model for three different scenarios (from left to right): The assumption of a constant 1% noise of the mean of the data values, optimized identically and independently distributed (i.i.d.) noise as part of the kernel, and measured noise. The bottom images show the ground-truth model, the noise model, and the accuracy of the model function approximation as a function of the iteration number. We can see that fixed noise, which is underestimated in this case, has the potential to render the autonomous experiment entirely ineffective. The hyperparameter optimization is misled, which leads to overfitting. Although the optimized noise model performs better, no extra emphasis is put on regions with high noise. The measured noise delivers the best model by strategically increasing measurement point density in high-noise regions.

and regions in which the ratio of eigenvalues of the Hessian is very large, indicating flat regions or ridges.

It is a frequently discussed topic whether to use optimization to find the hyperparameters or to use Markov Chain Monte Carlo (MCMC), which is the fully Bayesian approach. The argument against optimization is that it can lead to overfitting and the argument against MCMC is that it can be slow to converge, especially in situations when many hyperparameters have to be found. The advantage of optimization is that we do not need to specify a prior, which can be difficult for nonstandard kernels. A potential middle ground is to find optima using optimization and the Laplace approximations to account for the uncertainty in the hyperparameters.

Acquisition functionals for optimal measurement suggestions

As described in the "Some theory of Gaussian processes and related autonomous experimentation" section, the acquisition functional f_a has a major impact on the performance

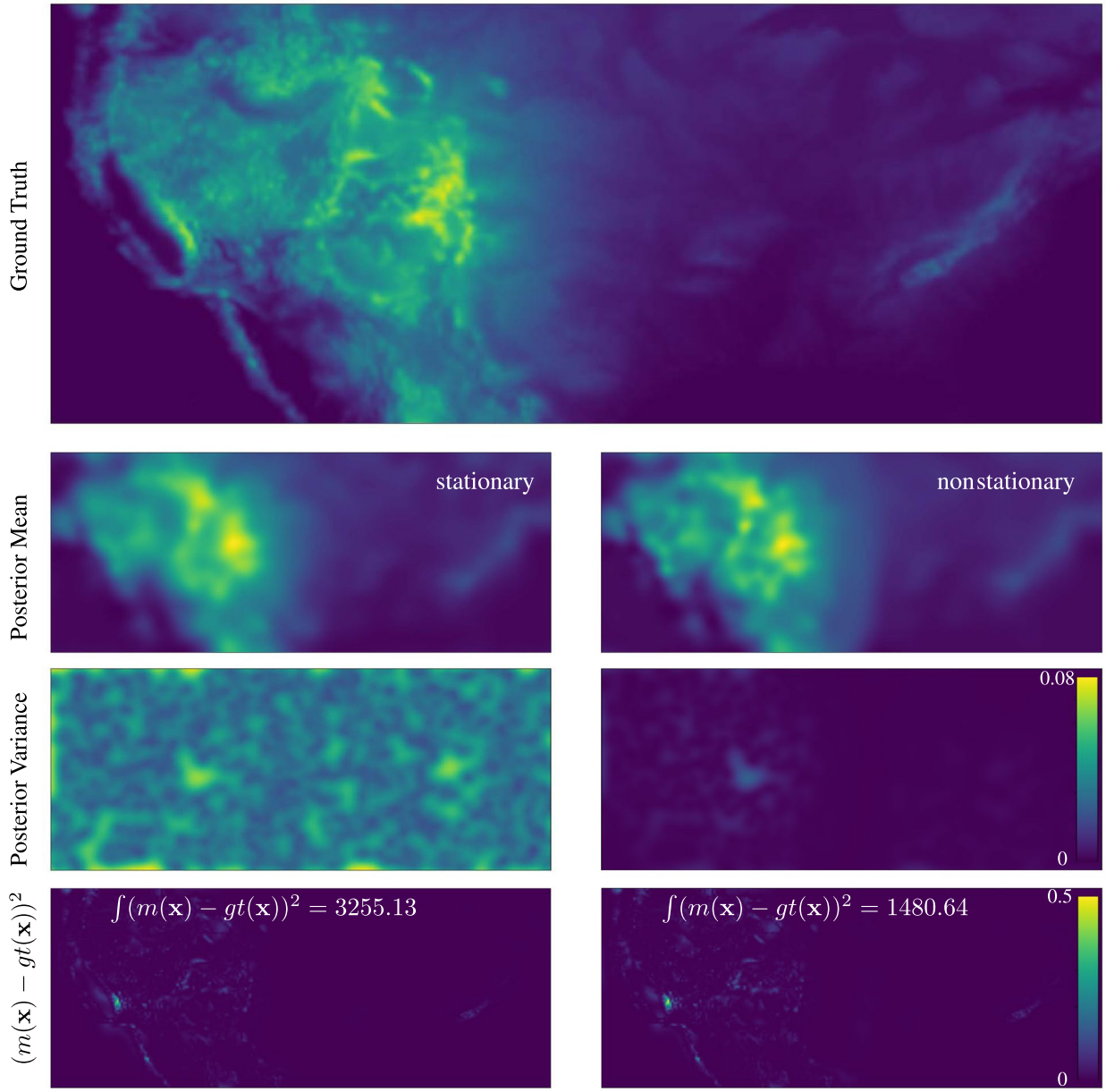


Figure 3. Stationary (left) and nonstationary (right) Gaussian processes applied to the topography of the United States. From the top: Ground-truth model using the entire data set, posterior means, variances, and errors for stationary (Equation 8) and nonstationary (Equation 9) kernels. The overall approximation error is lower for the nonstationary kernel. The key takeaway, however, is that uncertainties are much more realistically represented for the nonstationary kernel. The posterior variance for the stationary kernel does not respond to local characteristics of the underlying latent function. In contrast, the posterior variance calculated using the nonstationary kernel seems sensitive to the latent function, which results in very low uncertainties just east of the Rocky Mountains where the topography is nearly flat.

of an autonomous experiment. Imagine a situation in which a function should be explored but with a focus on regions $\{\tilde{\mathbf{x}} \subset \mathcal{X} : f(\tilde{\mathbf{x}}) < b, b \in \mathbb{R}\}$. In simpler terms, "valleys" with a certain "depth" should be given priority in the exploration. For materials scientists, the valley could be understood as a region of small crystal or grain sizes. Because the problem is exploratory, a practitioner could utilize $f_a(\mathbf{x}) = \sigma^2(\mathbf{x})$. However, the focus on valleys could motivate a lower-confidence-bound style acquisition functional $f_a(\mathbf{x}) = -(m(\mathbf{x}) - 3\sigma(\mathbf{x}))$. Ideally, the functional would allow the exploration of the regions of interest. Recalling that an acquisition functional is a function of the posterior probability density function—therefore the name "functional"—we could focus on the entire region of interest by defining

$$f_a(\mathbf{x}) = p(-\infty \leq f(\mathbf{x}) \leq b) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{b} e^{\frac{f(\mathbf{x})-m(\mathbf{x})^2}{2\sigma^2}} df$$
$$= 0.5\left(1 + \text{erf}\left(\frac{b - m(\mathbf{x})}{\sigma\sqrt{2}}\right)\right), \quad (11)$$

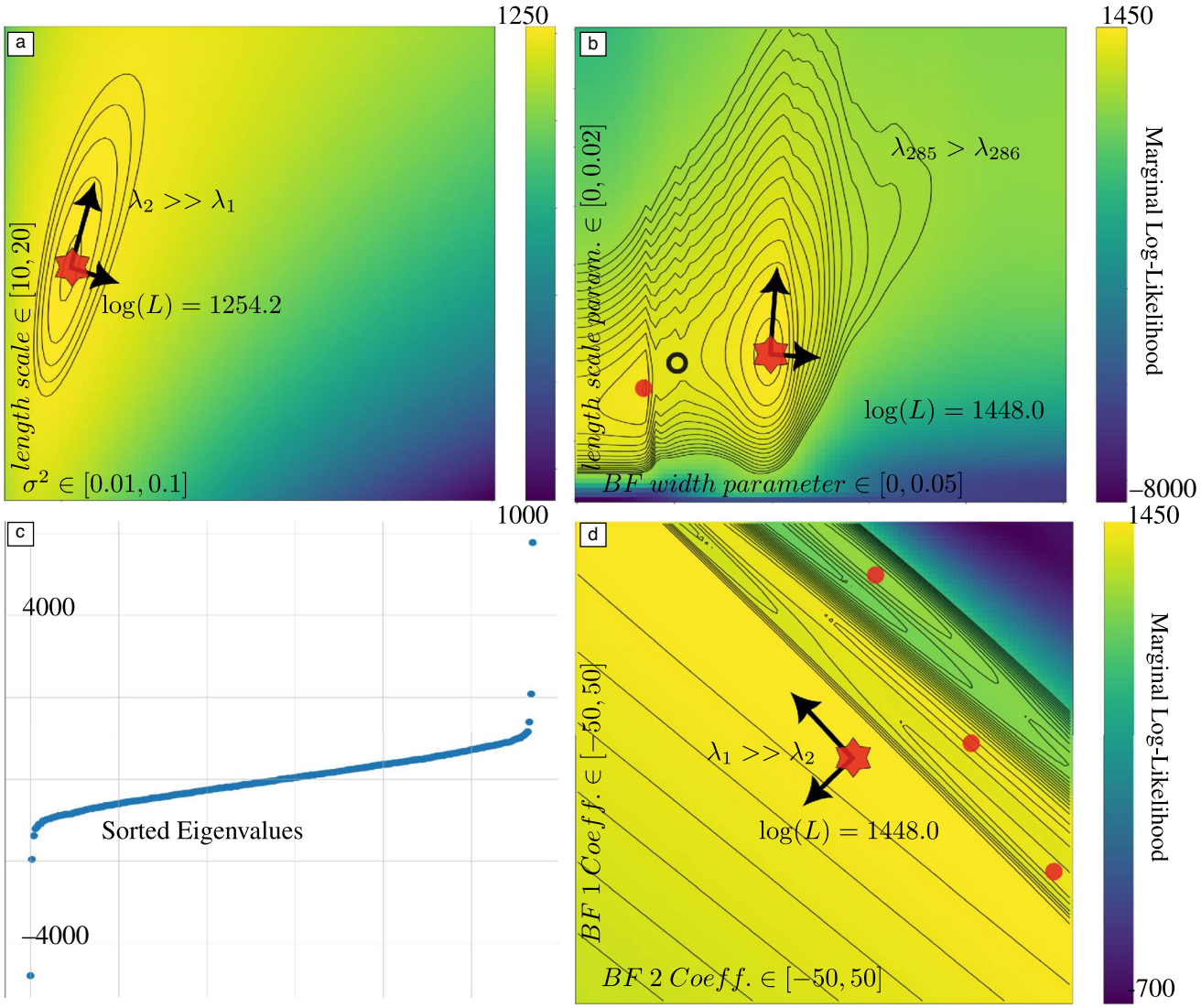


Figure 4. Figure emphasizing the difficulties of hyperparameter optimization (training) of highly flexible nonstationary kernels. This issue is inherent in most machine learning methods. (a) The marginal log-likelihood function for the stationary kernel (Equation 8). There is a global optimum that can easily and inexpensively be found with local or global optimizers. We can see from the eigenvalues of the Hessian that there is a direction (long arrow) in which we could move and the function value would only slightly change, which suggests some ambiguity in the model. We can see in (b–d) that this phenomenon becomes increasingly challenging in higher dimensions. (b) The marginal log-likelihood function over a 2D slice through the 286-dimensional space spanned by the hyperparameters. The two presented hyperparameters are the constant width of the basis functions of the kernel and the length scale. Just in this slice, we see three optima—two maxima and one saddle point—a local optimizer could identify as the final solution. (c) All found eigenvalues of the Hessian in the nonstationary case. The plot reveals that the final solution is actually a saddle point since the eigenvalues change sign. The key takeaway from this figure is that finding the hyperparameters of our model robustly becomes more challenging as we use more flexible nonstationary kernel designs. Although this is not a reason to avoid those more powerful kernels, it motivates the employment of powerful optimization algorithms that can handle nonuniqueness well. (d) The same function in a 2D slice spanned by the first two basis function coefficients. Here, many optima can be identified, and some are marked in the image. Every time, the ratio of the eigenvalues tells a story about the stability and robustness of the solution.

i.e., the probability that the latent function $f(\mathbf{x}) \leq b$. For a visual comparison of the mentioned acquisition functionals, see **Figure 5**. We can see in the figure that all acquisition functions find similar results. However, while pure exploration wastes measurements on exploring the function outside of the region of interest, the lower-confidence bound focuses too much on finding the minima. The acquisition functional (Equation 11) balances the two to allow exploration of the region of interest.

Conclusion

In this article, we looked at some mathematical idiosyncrasies and nuances of Gaussian process-driven autonomous data acquisition, which can, if not identified and countered, lead to undesired behavior and even error-prone model identification. We selected four examples: non-i.i.d. measurement noise, nonstationary kernels, the issue of ill-posed optimization problems that have to be solved for training a GP, and

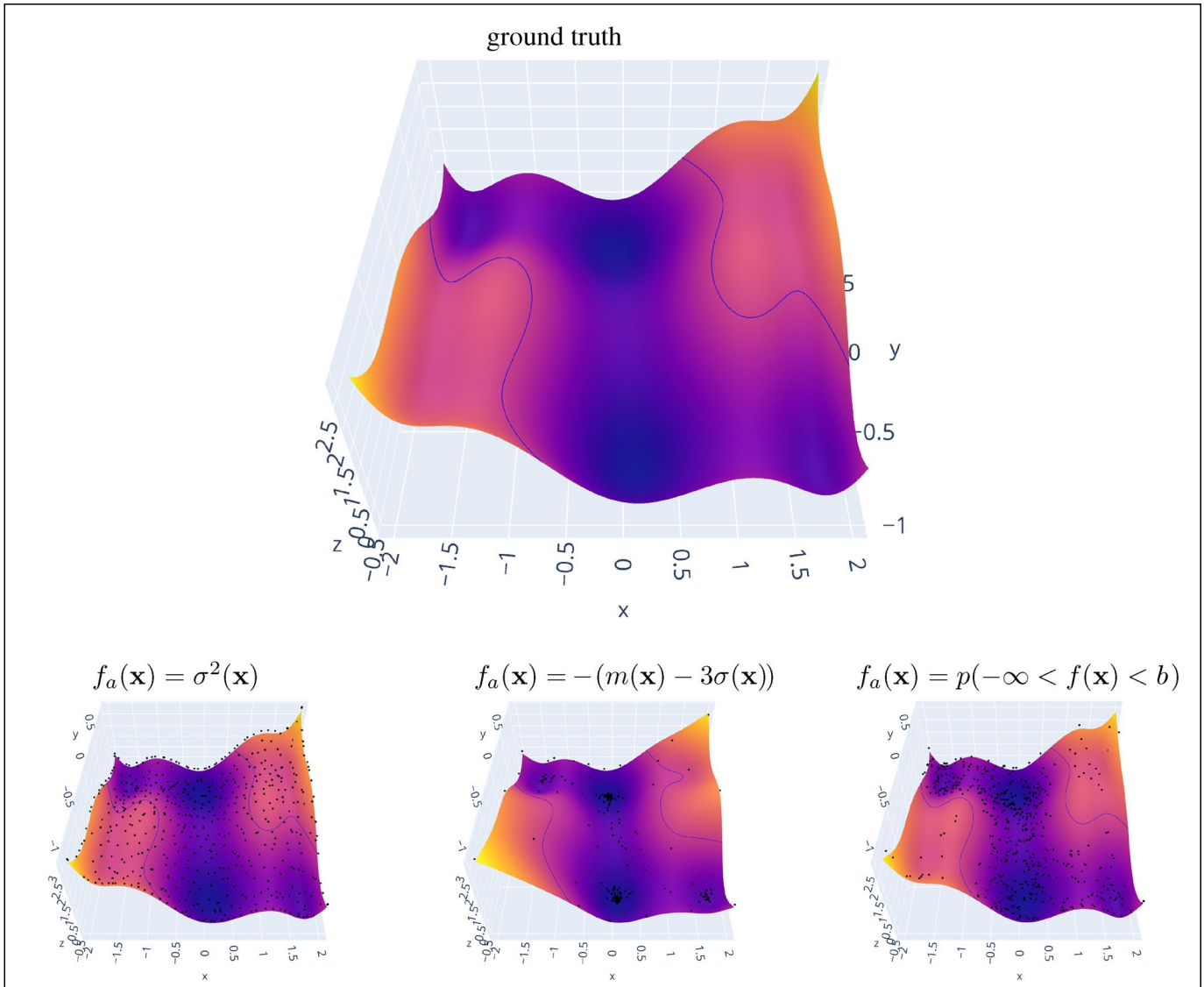


Figure 5. Three different acquisition functionals and their effect on exploring the latent function (top). We assume that the interest of the practitioner is in the valley region bordered by the contour line on both sides. An acquisition functional focusing on minimizing the uncertainty (pure exploration, left) will explore all regions of the input space. A lower-confidence-bound acquisition functional (middle) will focus on finding minima but runs the risk of not exploring the region of interest fully. An acquisition functional using the posterior distribution to value the probability of the latent function value being in the right range accomplishes exactly what was intended. Although neither of the shown acquisition functionals is particularly sophisticated, they show the importance of customization in order to maximize the benefit during an autonomous experiment.

tailored acquisition functionals. The key takeaway for all four examples is the same: To maximize the success of a GP function approximation or associated autonomous experiment, we have to go beyond standard setups and customize every aspect and property of the method. This takeaway is not exclusive to the GP framework, but applies to all ML methods.

The first test ("The role of noise for autonomous experiments" section) showed that the performance of an autonomous experiment strongly depends on the accurate estimation of the measurement noise. The dangers go much beyond inefficiencies; insufficiently accurate noise can render the resulting function approximation entirely useless. In addition, the estimation of uncertainties across the domain becomes so poor that data acquisition control becomes random due to very small length scales, which leads to overfitting (Figure 2) and high uncertainties almost everywhere. Our next test illustrated the benefits of allowing nonstationarity in the kernel design ("Stationary kernels versus nonstationary kernels for Gaussian processes" section). Function approximation and uncertainty quantification are significantly more accurate for nonstationary kernels. Both affect the performance of the autonomous experiment. More flexible kernels, however, lead to many more hyperparameters that have to be found. In the "Training as a constrained and ill-posed function optimization problem" section, we investigated the

nonuniqueness properties of the optimization problem and found that more hyperparameters and nonstationary kernels can lead to many solutions. This is an important lesson for novel automated kernel selection methods, such as deep kernel learning; unnecessary hyperparameters will lead to ill-posed optimization problems that could be impossible to solve robustly. The practitioner will often not be informed about the problem but has a faulty model to work with. Our last test ("Acquisition functionals for optimal measurement suggestions" section) drew attention to the acquisition functional of an autonomous experiment. The simple example illustrated (Figure 5) that a well-customized acquisition functional leads to a much targeted data acquisition.

Of course, there are many more pitfalls to look out for when performing ML and AE. Some of those are very practical, such as trying to visualize the model at least in slices to validate its validity before using it for decision-making and making sure the input data are cleaned up. Simple sanity checks can make sure the data are in order before applying any ML method: Were data points recorded twice? Is noise correctly communicated? Are there "NaN"s in the data set? What about outliers? And so on. For GPs, some of the sanity checks are naturally included because the trained hyperparameters often have a physical meaning, which can be checked against the intuition of the practitioner. For this, however, it is indispensable to know about the mathematics of the underlying method.

In summary, the authors hope that this collection of examples will help practitioners in the materials sciences avoid some of the common mathematical and statistical pitfalls when using Gaussian process-driven autonomous experimentation methods.

Acknowledgments

The investigation of non-i.i.d noise and the effect of the acquisition function on the data acquisition was funded through the Center for Advanced Mathematics for Energy Research Applications (CAMERA), which is jointly funded by the Advanced Scientific Computing Research (ASCR) and Basic Energy Sciences (BES) within the US Department of Energy Office of Science, under Contract No. DE-AC02-05CH11231. The work on nonstationary kernels and their impact on the training of a GP were supported by the Laboratory Directed Research and Development Program of Lawrence Berkeley National Laboratory under US Department of Energy Contract No. DE-AC02-05CH11231. K.R.'s work was funded by the University at Buffalo. We want to thank K. Yager (Brookhaven National Laboratory) for reviewing the manuscript before submission.

Data availability

The IR spectroscopy data ("The role of noise for autonomous experiments" section) were first presented in Reference 35 but is, to the best of the authors' knowledge, not publicly available. The topography data set is publicly available at *https://drive.google.com/file/d/1tjpTHmAGQvMa-f8OSRY67ppAY4AZKH3z/view?usp=sharing*.

Conflict of interest

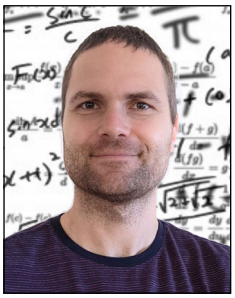
On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open access

References

1. M.C.A. Clare, O. Jamil, C.J. Morcrette, *Q. J. R. Meteorol. Soc.* **147**(741), 4337 (2021)
2. J.A. Weyn, D.R. Durran, R. Caruana, N. Cresswell-Clay, *J. Adv. Model. Earth Syst.* **13**(7), e2021002502 (2021)
3. M. AlQuraishi, *Cell Syst.* **8**(4), 292 (2019)
4. E. Callaway, *Nature* **588**(7837), 203 (2020)
5. K.R. Chowdhary, "Natural Language Processing," in *Fundamentals of Artificial Intelligence*, 1st edn. (Springer, New Delhi, 2020), p. 603
6. D. Keysers, T. Deselaers, C. Gollan, H. Ney, *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(8), 1422 (2007)
7. B.B. Traore, B. Kamsu-Foguem, F. Tangara, *Ecol. Inform.* **48**, 257 (2018)
8. R.W. Epps, M.S. Bowen, A.A. Volk, K. Abdel-Latif, S. Han, K.G. Reyes, A. Amassian, M. Abolhasani, *Adv. Mater.* **32**(30), 2001626 (2020)
9. M.M. Noack, K.G. Yager, M. Fukuto, G.S. Doerk, R. Li, J.A. Sethian, *Sci. Rep.* **9**, 11809 (2019)
10. M.M. Noack, G.S. Doerk, R. Li, M. Fukuto, K.G. Yager, *Sci. Rep.* **10**, 1325 (2020)
11. M.M. Noack, G.S. Doerk, R. Li, J.K. Streit, R.A. Vaia, K.G. Yager, M. Fukuto, *Sci. Rep.* **10**, 17663 (2020)
12. D.P. Tabor, L.M. Roch, S.K. Saikin, C. Kreisbeck, D. Sheberla, J.H. Montoya, S. Dwaraknath, M. Aykol, C. Ortiz, H. Tribukait, C. Amador-Bedolla, C.J. Brabec, B. Maruyama, K.A. Persson, A. Aspuru-Guzik, *Nat. Rev. Mater.* **3**(5), 5 (2018)
13. T. Brants, A.C. Popat, P. Xu, F.J. Och, J. Dean, "Large Language Models in Machine Translation," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, ed. by J. Eisner (Association for Computational Linguistics, Prague, 2007), p. 858
14. R. Dale, *Nat. Lang. Eng.* **27**(1), 113 (2021)
15. S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti, E. Zhang, R. Child, R. Yazdani Aminabadi, J. Bernauer, X. Song, M. Shoeybi, Y. He, M. Houston, S. Tiwary, B. Catanzaro, arXiv:2201.11990 (2022)
16. K. Valmeekam, A. Olmo, S. Sreedharan, S. Kambhampati, arXiv:2206.10498 (2022)
17. Y. Cho, L. Saul, *Adv. Neural Inf. Process. Syst.* **22**, 1 (2009)
18. J. Lee, Y. Bahri, R. Novak, S.S. Schoenholz, J. Pennington, J. Sohl-Dickstein, arXiv:1711.00165 (2017)
19. M. Unser, arXiv:2206.14625 (2022)
20. G.C. Cawley, N.L.C. Talbot, *Neural Process. Lett.* **16**(3), 293 (2002)
21. V. Vovk, "Kernel Ridge Regression," in *Empirical Inference* (Springer, New York, 2013), pp. 105–116
22. M. Welling, in *Max Welling's Classnotes in Machine Learning* (2013), p. 1
23. T. Hofmann, B. Scholkopf, A.J. Smola, *Ann. Stat.* **36**(3), 1171 (2008)
24. C.E. Rasmussen, C.K.I. Williams, *Gaussian Processes for Machine Learning*, vol. 2 (MIT Press, Cambridge, 2006)
25. W.S. Noble, *Nat. Biotechnol.* **24**(12), 1565 (2006)

26. R. Rosipal, M. Girolami, L.J. Trejo, A. Cichocki, *Neural Comput. Appl.* **10**(3), 231 (2001)
27. K.E. Pilario, M. Shafiee, Y. Cao, L. Lao, S.-H. Yang, *Processes* **8**(1), 24 (2020)
28. A.E. Gongora, B. Xu, W. Perry, C. Okoye, P. Riley, K.G. Reyes, E.F. Morgan, K.A. Brown, *Sci. Adv.* **6**(15), eaaz1708 (2020)
29. M.M. Noack, P.H. Zwart, D.M. Ushizima, M. Fukuto, K.G. Yager, K.C. Elbert, C.B. Murray, A. Stein, G.S. Doerk, E.H.R. Tsai, R. Li, G. Freychet, M. Zhernenkov, H.-Y.N. Holman, S. Lee, L. Chen, E. Rotenberg, T. Weber, Y. Le Goc, M. Boehm, P. Steffens, P. Mutti, J.A. Sethian, *Nat. Rev. Phys.* **3**(10), 685 (2021)
30. M. Seifrid, R. Pollice, A. Aguilar-Granda, Z.M. Chan, K. Hotta, C.T. Ser, J. Vestfrid, T.C. Wu, A. Aspuru-Guzik, *Acc. Chem. Res.* **55**(17), 2454 (2022)
31. E. Stach, B. DeCost, A.G. Kusne, J. Hattrick-Simpers, K.A. Brown, K.G. Reyes, J. Schrier, S. Billinge, T. Buonassisi, I. Foster, C.P. Gomes, J.M. Gregoire, A. Mehta, J. Montoya, E. Olivetti, C. Park, E. Rotenberg, S.K. Saikin, S. Smullin, V. Stanev, B. Maruyama, *Matter* **4**(9), 2702 (2021)
32. J. Snoek, H. Larochelle, R.P. Adams, *Adv. Neural Inf. Process. Syst.* **25**, 1 (2012)
33. Y. Yu, "Several New Advances for Gaussian Process Models," PhD thesis, (Northwestern University, Evanston, IL, 2020)
34. P.I. Frazier, arXiv:1807.02811 (2018)
35. P.M. Valdespino-Castillo, P. Hu, M. Merino-Ibarra, L.M. López-Gómez, D. Cerqueda-García, R. González-De Zayas, T. Pi-Puig, J.A. Lestayo, H.-Y. Holman, L.I. Falcón, *Front. Microbiol.* **9**, 510 (2018)
36. M.M. Noack, J.A. Sethian, arXiv:2102.03432 (2021) □



Marcus M. Noack is a research scientist at Lawrence Berkeley National Laboratory (LBNL). He completed postdoctoral research at LBNL on stochastic function approximation and autonomous experimentations. He received his master's degree in geophysics from Friedrich-Schiller University, Germany. Working as a doctoral candidate at Simula Research Laboratory, Norway, he was able to pursue his interests in the theory of wave propagation and mathematical function optimization. There, he leveraged his knowledge in theoretical and numerical physics and applied mathematics, and connected it with high-performance computing to create efficient methods to model wave propagation and solve nonlinear inverse problems. Noack received his PhD degree in theoretical and mathematical physics from the University of Oslo, Norway. His current research focuses on stochastic processes for function approximation and dimensionality reduction, function optimization, and high-performance computing while serving the autonomous experimentation community by providing support and practical software. Noack can be reached by email at MarcusNoack@lbl.gov.

Kristofer G. Reyes is an assistant professor in the Department of Materials Design and Innovation at the University at Buffalo, The State University of New York. He received his PhD degree in applied mathematics from the University of Michigan. He completed postdoctoral research in the Department of Operations Research and Financial Engineering at Princeton University. His research interests include decision-making algorithms and models for autonomous and guided experiments, knowledge elicitation and representation, and high-performance computational materials. Reyes can be reached by email at kreyes3@buffalo.edu.