Advanced Stationary and Non-Stationary Kernel Designs for Domain-Aware Gaussian Processes

Marcus M. Noack[1,*] and James A. Sethian[1,2]

[1]The Center for Advanced Mathematics for Energy Research Applications (CAMERA), Lawrence Berkeley National Laboratory, Berkeley, CA 94720
[2]Department of Mathematics, University of California, Berkeley
[*]MarcusNoack@lbl.gov

February 28, 2023

Abstract

Gaussian process regression is a widely-applied method for function approximation and uncertainty quantification. The technique has gained popularity recently in the machine learning community due to its robustness and interpretability. The mathematical methods we discuss in this paper are an extension of the Gaussian-process framework. We are proposing advanced kernel designs that only allow for functions with certain desirable characteristics to be elements of the reproducing kernel Hilbert space (RKHS) that underlies all kernel methods and serves as the sample space for Gaussian process regression. These desirable characteristics reflect the underlying physics; two obvious examples are symmetry and periodicity constraints. In addition, we want to draw attention to non-stationary kernel designs that can be defined in the same framework to yield flexible multi-task Gaussian processes. We will show the impact of advanced kernel designs on Gaussian processes using several synthetic and two scientific data sets. The results show that informing a Gaussian process of domain knowledge, combined with additional flexibility and communicated through advanced kernel designs, has a significant impact on the accuracy and relevance of the function approximation.

1 INTRODUCTION

Gaussian processes (GPs) [15] provide a powerful mathematical framework for function approximation from data. The associated technique is generally referred to as Gaussian process regression (GPR). GPs are flexible, robust, non-parametric and naturally include uncertainty quantification. Given some data $\mathcal{D} = \{\mathbf{x}_i, y_i\}$, the GP regression model assumes $y(\mathbf{x}) = f(\mathbf{x}) + \epsilon(\mathbf{x})$. Here, $\mathbf{x}$ is the position in some input or parameter space, y is the associated noisy function evaluation, and $\epsilon(\mathbf{x})$ represents the noise term. The covariance matrix $\mathbf{\Sigma}$ of the prior Gaussian probability distribution is defined via kernel functions $k(\mathbf{x}_i, \mathbf{x}_j; \phi)$, where ϕ is a set of hyperparameters that are commonly found by maximizing the marginal log-likelihood of the data. Kernels induce an inner product in a Hilbert space and therefore impose a metric, which can be interpreted as a similarity measure. The flexibility and prior-knowledge-adherence of kernel functions, and therefore, of the associated similarity measures is the main focus of this paper.

As a Bayesian method, Gaussian processes are capable of domain-aware approximations of model functions. By placing one or more prior Gaussian probability distributions over a carefully-defined function space, and using the posterior Gaussian distribution in a way that captures the desired features, we can take into account several data sets and a variety of domain knowledge bases. Generally speaking, the theory of Gaussian processes allows for three main possibilities to communicate domain knowledge:

1. We can extract subspaces of the function space in such a way that all elements have certain desired characteristics. The extracted function space is the, so-called, reproducing kernel Hilbert space (RKHS). This can be accomplished by developing advanced designs for stationary and non-stationary kernels;

2. The prior can be placed and shaped in accordance with domain knowledge; this can, for instance, be accomplished by a constrained log-likelihood optimization or by minimizing the Kullback-Leibler divergence between priors [11, 3, 13];

3. Flexible multi-task Gaussian processes can be defined using non-stationary kernels. Kernels can be seen as a similarity measure; the more flexibility they have, the more can be learned across the input space and the different tasks. The advantage here is that, if enough flexibility is provided to the kernel, no other changes have to made to a single-task GP.

In this work, we focus on points 1 and 3, which target advanced kernel designs for stationary and non-stationary kernels. Kernels dictate which functions are part of the RKHS, and are therefore optimally suited to impose hard constraints on the posterior mean. One interesting example is the restriction to symmetry or periodicity of the posterior mean. One goal of this paper is to show that taking advantage of advanced kernel designs makes GPR significantly more accurate. Designing these kernels can be done by taking advantage of permitted operators, such as adding kernels and applying linear operators to them. Advanced kernel designs also allow for a very natural way of dealing with multi-modal data sets. In the GP literature, this is often referred to as multi-task, multi-output or multivariate regression problems. Typically, there is no natural distance between tasks and covariances between them are often heterogeneous across the input space and across the task indices. Due to this complexity, many workarounds have been proposed [1, 16]. However, no workaround is necessary if the kernels are given enough flexibility to find the optimal, possibly non-constant distances between tasks. That way, most common problems of multi-output GPR, such as missing data or missing cross-task covariances, are addressed or avoided.

Contributions. The contributions in this paper can be summarized as follows: (1) We show how to tailor kernel designs to communicate domain-knowledge to a GP, using both known stationary kernels as well as introducing and deriving new stationary kernels; (2) We show how to build customized non-stationary kernels, using previously-published but not well-established non-stationary kernel designs; and (3) We draw attention to a natural way to implement multi-task GPs by formulating them in terms of non-stationary kernel designs. As we will see, while this idea is not new, it deserves reevaluation in the presence of advanced non-stationary kernel designs.

Organization. This paper is organized as follows. First, we will show the basic Gaussian process regression framework which takes advantage of the standard kernel classes. We will see that, while the standard kernels are very general, there are weaknesses associated with them, which lead to unnecessary inaccuracies of the function approximation. Second, we will show the mathematics needed to make a Gaussian process domain aware by defining advanced kernel designs. These designs are partly known to the Gaussian-process community but largely unknown to the practitioner. While presenting them, we will show their impact on GP function approximations directly. Here we will also discuss kernels for multi-task Gaussian processes. Third, we will show the impact of the presented methodologies on experimental data which will be simulated by using previously-acquired scientific data sets.

2 The Mathematics of Advanced Kernel Designs for Gaussian Processes

2.1 Preliminaries

We define a set $\mathcal{X}_i \subset \mathbb{R}^{n_1}$, which is often referred to as the parameter space or the input space, and elements $\mathbf{x}_i \in \mathcal{X}_i$. We also define a set $\mathcal{X}_o \subset \mathbb{R}^{n_2}$ with elements $\mathbf{x}_o \in \mathcal{X}_o$, which represent the arbitrary but fixed indices of all function values of a vector-valued function whose domain consists of values that are elements of $\mathcal{X}_i$. These indices often have no physical meaning or equivalent, and have to be chosen arbitrarily, which constitutes the main difficulty for multi-task regression. To tackle multi-output Gaussian process regression, we are defining the Cartesian product space $\mathcal{X} = \mathcal{X}_i \times \mathcal{X}_o$, $\mathcal{X} \subset \mathbb{R}^{n_1+n_2}$ with elements $\mathbf{x} = [\mathbf{x}_i, \mathbf{x}_o]^T$. We call this set the index set, because the functions defined on it are elements of a function space we will define later. Note however, that $\mathcal{X}_i$ as well as $\mathcal{X}_o$ are considered index sets, since their elements index function values of functions that are themselves elements of a set. The assumption that the input and output spaces are a subspace of the Euclidean space is not a strict requirement but used here for simplicity. We define a total of four functions on $\mathcal{X}$. First, the latent function $f = f(\mathbf{x})$ which can be interpreted as the inaccessible ground truth. Second, the noisy measurements $y = y(\mathbf{x})$. Third, the posterior-mean function $m(\mathbf{x})$. Fourth, the posterior-variance function $\sigma^2(\mathbf{x})$. We note that typically in multi-task GPR, those functions are vector-valued functions. Since we are introducing the Cartesian product space $\mathcal{X}$, this is not necessary; we have effectively reduced a multi-output Gaussian process to a single-output Gaussian process. Since we are not interpreting the tasks as a set of functions on $\mathcal{X}_i$ but

instead as a scalar function on $\mathcal{X}_o$ with its own metric, we refer to this as a function-valued GP. For instance, the output could be defined on $\mathbb{R}$, which leads to $\mathcal{X} = \mathcal{X}_i \times \mathbb{R}$; in this case the output is a function on $\mathbb{R}$. The general concept of transforming a multi-output GP to a single-output GP is not new, and is normally referred to as single-target method or output-as-input-view [12] and criticized for not taking into account cross-task covariances [1]; however, this criticism only applies to stationary kernels. One of the goals of this paper is to achieve cross-task covariances by defining flexible non-stationary kernels.

Next, we define a pre-Hilbert space

$$\mathcal{H} = \{f(\mathbf{x}) : f(\mathbf{x}) = \sum_{i}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}), \forall \boldsymbol{\alpha} \in \mathbb{R}^N, \mathbf{x} \in \mathbb{R}^n\}, \tag{1}$$

with covariance function $k(\mathbf{x}_i, \mathbf{x})$; N is the number of data-point locations, and $f(\mathbf{x})$ is the unknown latent function. Strictly speaking, Equation (1) is not a full infinite-dimensional pre-Hilbert space but a finite-dimensional sub-space spanned by the data. For it to qualify as a Hilbert space we have to equip the space with the norm $||f||_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$ and add all limit points of sequences that converge in that norm. As a reminder, note that scalar functions over $\mathcal{X}$, e.g. $f(\mathbf{x})$, are vectors (bold typeface) in $\mathcal{H}$. A kernel induces an inner product of two elements $\in \mathcal{H}$, i.e,
$\langle f(\mathbf{x}), g(\mathbf{x}) \rangle_{\mathcal{H}} = \sum_i \sum_j \alpha_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j)$, where α_i and β_j are coefficients, and $g(\mathbf{x}) = \sum_i^N \beta_i k(\mathbf{x}_i, \mathbf{x})$.

Definition 1. *A kernel is a symmetric and positive semi-definite (p.s.d.) function $k(\mathbf{x_1}, \mathbf{x_2})$, $\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, it therefore satisfies $\sum_i^N \sum_j^N c_i\, c_j\, k(\mathbf{x}_i, \mathbf{x}_j) \geq 0\, \forall N,\, \mathbf{x} \in \mathcal{X},\, \mathbf{c} \in \mathbb{R}^N$*

Given this definition, it is clear that the set of kernels is closed under addition, multiplication and linear transformation [2], which we will build upon later. Given the definition of the pre-Hilbert space (Equation 1), it can be shown that for elements of $\mathcal{H}$

$$\langle k(\mathbf{x}_0, \mathbf{x}), f(\mathbf{x}) \rangle = f(\mathbf{x}_0), \tag{2}$$

which is the reason the completion of the space $\mathcal{H}$ is called Reproducing Kernel Hilbert Space. Reproducing in that context refers to the fact that the inner product of a kernel, evaluated at a point, with a function produces the function at that point; the inner product "reproduces" the function value, which is the essence of Equation (2). Gaussian processes are based on defining a prior probability distribution over the RKHS. In this case the kernels are understood as covariance functions

$$k(\mathbf{x}_1, \mathbf{x}_2) = \int_{\mathcal{H}} f(\mathbf{x}_1) f(\mathbf{x}_2) q(f)\, df, \tag{3}$$

where q is some density function.

2.2 A Birds-Eye View on Gaussian Processes

Given data $\mathcal{D} = \{\mathbf{x}_i, y_i\}$, a prior probability distributions over functions $f(\mathbf{x})$ can be defined as

$$p(\mathbf{f}) = \frac{1}{\sqrt{(2\pi)^{\dim}|\mathbf{K}|}} \exp\left[-\frac{1}{2}(\mathbf{f} - \boldsymbol{\mu})^T \mathbf{K}^{-1} (\mathbf{f} - \boldsymbol{\mu})\right], \tag{4}$$

where $\mathbf{K}$ is the covariance matrix of the data, calculated by applying the kernel $k(\mathbf{x}_1, \mathbf{x}_2)$ (see Definition 1) to the data-point positions, and $\boldsymbol{\mu}$ is the prior mean vector. We can define the likelihood over functions $y(\mathbf{x})$ as

$$p(\mathbf{y}|\mathbf{f}) = \frac{1}{\sqrt{(2\pi)^{\dim}|\mathbf{V}|}} \exp\left[-\frac{1}{2}(\mathbf{y} - \mathbf{f})^T \mathbf{V}^{-1} (\mathbf{y} - \mathbf{f})\right], \tag{5}$$

where $\mathbf{V}$ is the matrix of the non-i.i.d. noise[5]. The noise is responsible for the difference between the unknown latent function $f(\mathbf{x})$ and the measurements $y(\mathbf{x})$. In the standard literature, often only i.i.d. noise is discussed, which is insufficient for many applications [5].

The vast majority of work published about Gaussian processes uses only a handful of standard kernels to compute covariances. By far the most frequently used kernel is the squared exponential kernel [9]

$$k(\mathbf{x}_1, \mathbf{x}_2) = \sigma_s^2 \exp\left[-\frac{||\mathbf{x}_1 - \mathbf{x}_2||^2}{2l^2}\right], \tag{6}$$

where σ_s^2 is the signal variance and l is the length scale which can be anisotropic, as we will see later. The signal variance and the length scales are examples of the, so-called, hyperparameters (ϕ) of the Gaussian process and are calculated by solving

$$\begin{aligned}\arg\max_{\phi} \Big(\log(L(D,\phi)\Big) \; = \\ -\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}(\phi))(\mathbf{K}(\phi)+\mathbf{V})^{-1}(\mathbf{y}-\boldsymbol{\mu}(\phi)) \\ -\frac{1}{2}\log(|\mathbf{K}(\phi)+\mathbf{V}|) - \frac{\dim(\mathbf{y})}{2}\log(2\pi)\Big). \end{aligned} \tag{7}$$

Finding the hyperparameters via deterministic function optimization can be seen as somewhat of a detour around a fully Bayesian approach and has to be done carefully to avoid over-fitting. However, many optimization algorithms offer some advantages compared to statistical techniques (Markov Chain Monte Carlo for instance) such as favorable scalability with dimensionality and higher probability of finding close-to-global solutions. A brief discussion with pointers to more information can be found in [15]. Given the hyperparameters, we can calculate and condition the joint prior

$$p(\mathbf{f},\mathbf{f}_0) = \frac{1}{\sqrt{(2\pi)^{\dim}|\boldsymbol{\Sigma}|}} \exp\left[-\frac{1}{2}\Big(\begin{bmatrix}\mathbf{f}-\boldsymbol{\mu}\\ \mathbf{f}_0-\boldsymbol{\mu}_0\end{bmatrix}^T \boldsymbol{\Sigma}^{-1}\begin{bmatrix}\mathbf{f}-\boldsymbol{\mu}\\ \mathbf{f}_0-\boldsymbol{\mu}_0\end{bmatrix}\Big)\right], \tag{8}$$

where

$$\Sigma \; = \; \begin{pmatrix}\mathbf{K} & \boldsymbol{\kappa}\\ \boldsymbol{\kappa}^T & \boldsymbol{\mathcal{K}}\end{pmatrix}, \tag{9}$$

to obtain the well-known posterior

$$\begin{aligned} p(\mathbf{f}_0|\mathbf{y}) \; &= \; \int_{\mathbb{R}^N} p(\mathbf{f}_0|\mathbf{f},\mathbf{y})\, p(\mathbf{f},\mathbf{y})\, d\mathbf{f} \\ &\propto \mathcal{N}(\boldsymbol{\mu}+\boldsymbol{\kappa}^T\,(\mathbf{K}+\mathbf{V})^{-1}\,(\mathbf{y}-\boldsymbol{\mu}), \boldsymbol{\mathcal{K}}-\boldsymbol{\kappa}^T\,(\mathbf{K}+\mathbf{V})^{-1}\,\boldsymbol{\kappa}), \end{aligned} \tag{10}$$

where $\boldsymbol{\kappa}_i \; = \; k(\mathbf{x}_0,\mathbf{x}_i,\phi)$, $\boldsymbol{\mathcal{K}} \; = \; k(\mathbf{x}_0,\mathbf{x}_0,\phi)$ and $\mathbf{K}_{ij} \; = \; k(\mathbf{x}_i,\mathbf{x}_j,\phi)$. $\mathbf{x}_0$ are the points at which the Gaussian posterior should be predicted. $\mathbf{f}_0$ are values of the latent function f at the points $\mathbf{x}_0$. The posterior contains the posterior mean $m(\mathbf{x})$ and the posterior variance $\sigma^2(\mathbf{x})$ (see also Section 2.1).

2.3 Basic Kernel Design and its Weaknesses

In standard Gaussian process regression, we aim to approximate one function value (single task) across the index set. Distances within the index set are generally assumed to be isotropic and Euclidean. Moreover, we often require first and second order stationarity of the process. These assumptions translate into kernels of the form

$$k(\mathbf{x}_1,\mathbf{x}_2) \; = \; k(||\mathbf{x}_1-\mathbf{x}_2||,\sigma_s^2,l), \tag{11}$$

where l is the isotropic and constant length scale and σ_s^2 is the constant signal variance, and constant prior-mean functions. Therefore, independent of the dimensionality of the index set, we only have to solve Equation (7) for two hyperparameters, one signal variance and one length scale. In addition, we do not assume any particular characteristics of the model function, which translates to the use of standard kernels (e.g. Matérn, exponential, squared exponential). In the vast majority of published work, the squared exponential kernel is used [9]. Also, when several tasks are involved, they are often assumed to be independent in the standard GP framework. If the tasks are assumed to be correlated, the used methods are either based on significant augmentations of the basic GP theory or on stationary separable kernels [1].

While the standard approach yields an agnostic and widely applicable approach to regression, it also has some considerable drawbacks. In this paper, we focus on two of them:

- First, by defining any kernel, the user implicitly chooses which functions — carrying hidden restrictions or assumptions — are elements of the RKHS. Often, this is done by accident without knowing what is imposed. For instance, using the squared exponential kernel imposes an infinite order of differentiability onto the posterior-mean function, even if this assumption is actually not backed by

physics or other domain knowledge. Restricting the approximated function to certain properties that are not reasonable should be avoided. An alternative, which is one focus of this paper and discussed below, exploits the fact that the user often knows certain local and global characteristics of the posterior mean. Enforcing them can yield a vastly improved accuracy of the approximation. In the case of stationary kernel designs, this is due to extra information that is propagated to unexplored regions of the index set.

- Second, stationary kernels do not have the flexibility to encode varying similarities across the index set. That means, two function values in one corner of the domain will have the same covariance as two other function values in the other corner of the domain, as long as their respective point-distances are the same; the inner product and therefore the similarity will not depend on the respective location of the point pairs. This makes it difficult to learn similarities across tasks, since there is no natural distance between them. As long as this distance is assumed to be constant across the input and output domains, stationary separable kernels are sufficient to encode the covariances. If this assumption is dropped, we typically have to rely on workarounds to estimate covariances [1, 16]. One alternative, we want to draw the reader's attention to, is to addresses this issue by adding extra flexibility to non-stationary kernels which translates into a method that is able to learn more complicated patterns of the data set across the input and the output space (tasks). In doing so, and in contrast to standard methods, the basic theory of GPs remains unchanged, and the entire difference between single-task and multi-task GPs is contained within the kernel design, maintaining the inherent robustness of a GP and avoiding common problems such as missing data, assumed linear dependence of tasks, and reduced interpretability.

We want to note here that there are many methods to address the issues with multi-task Gaussian processes. See Borchani et al. [1] for a comprehensive overview.

To reiterate, the main goal in this work is to define symmetric positive semi-definite functions

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \tag{12}$$

that serve as stationary and non-stationary kernel functions and effectively extract a RKHS in such a way that it only contains functions with certain desirable characteristics and is able to encode and learn complicated cross-task covariances.

We next discuss stationary kernels, showing how to incorporate hard constraints on the posterior mean. This is followed by a generalization to non-stationary kernels, showing how they can provide a framework for multi-task GPs.

2.4 Advanced Stationary Kernel Designs for Hard Constraints on the Posterior Mean

Stationary kernels are positive definite functions of the form

$$k(\mathbf{x}_1, \mathbf{x}_2) \;=\; k(||\mathbf{x}_1 - \mathbf{x}_2||), \tag{13}$$

where $||\cdot||$ is some norm. The Euclidean norm is used in the overwhelming majority of studies.

The set of kernel functions is closed under addition, multiplication, and application of linear operators. Therefore, kernel functions can be combined in many ways to formulate powerful definitions of similarity between data points. We will see that this can be used to inform the process that similarity is recurrent in $\mathcal{X}$ or follows a certain structure.

2.4.1 Stationary Kernels Constraining Differentiability

The Matérn kernel class is defined as

$$k(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{2^{v-1}\Gamma(v)} \Big(\frac{\sqrt{2v}}{l} r \Big)^v B_v \Big(\frac{\sqrt{2v}}{l} r \Big), \tag{14}$$

where r is some metric in $\mathcal{X}$, B_v is the modified Bessel function and v is the parameter controlling the differentiability. Combined with anisotropic kernel definitions, a practitioner can control the level of differentiability in each direction of the input space. This is a rather well-known characteristic of kernels and is included here for completeness.

2.4.2 Kernels for Additive Functions

Approximating a function of the form $\sum_i g_i(x_i)$ can be accomplished by choosing a Gaussian process with defining kernel

$$k(\mathbf{x}_1, \mathbf{x}_2) \;=\; \sum_i k_i(\mathbf{x}_1^i, \mathbf{x}_2^i). \tag{15}$$

The resulting process can propagate information about the function into regions of the index set where information is given only in ($n-1$-dimensional) subspaces. Figure 1 shows an example of how additive kernels can be used. While points are only given in a sub-space of the index set, the information can be propagated into all of $\mathcal{X}$. The standard kernel used in the example shown in Figure 1 is defined as

$$k(\mathbf{x}_1, \mathbf{x}_2) \;=\; 2\, \exp[-\frac{|\mathbf{x}_1 - \mathbf{x}_2|}{0.5}], \tag{16}$$

where $|\cdot|$ denotes the Euclidean distance in $\mathcal{X}$. The additive kernel is defined as

$$k(\mathbf{x}_1, \mathbf{x}_2) \;=\; \exp[-\frac{|x_1^1 - x_2^1|}{0.5}] + \exp[-\frac{|x_1^2 - x_2^2|}{0.5}]. \tag{17}$$

Figure 1 shows how powerful the knowledge of additivity can be; information can be propagated far away from the available data along axes directions.

2.4.3 Anisotropy of Distance Measures on $\mathcal{X}$

In addition to summation, one can also combine kernels by a product. In both cases, every direction can have its own length scale, giving rise to two formulations of anisotropy

$$k(\mathbf{x}_1, \mathbf{x}_2) \;=\; \exp[-\frac{|x_1^1 - x_2^1|}{l_1}] + \exp[-\frac{|x_1^2 - x_2^2|}{l_2}] \quad [Additive] \tag{18}$$

$$k(\mathbf{x}_1, \mathbf{x}_2) \;=\; \exp[-\frac{|x_1^1 - x_2^1|}{l_1}]\, \exp[-\frac{|x_1^2 - x_2^2|}{l_2}] \quad [Multiplicative]. \tag{19}$$

However, the additive kernel comes with additional properties presented in the last section. If the model function is not additive, the use of kernel (18) will lead to wrong predictions. Another way of implementing anisotropy is by altering the Euclidean distance in $\mathcal{X}$ with a different metric such that

$$k(\mathbf{x}_1, \mathbf{x}_2) = k((\mathbf{x}_1 - \mathbf{x}_2)^T\, \mathbf{M}\, (\mathbf{x}_1 - \mathbf{x}_2)), \tag{20}$$

where M is any symmetric positive definite matrix. More on that can be found in [15, 5]. Anisotropy plays an important role in many data sets and its inclusion is vitally important [5].

2.4.4 Linear Operators Acting on Kernels

Ginsbourger et al. [2] pointed out that kernels can be passed through linear operators.

Theorem 1. *If $k(x_1, x_2)$ is a kernel, then $L_{x_1}(L_{x_2}(k))$ is also a valid kernel function.*

We can use this theorem to derive kernels for many different situations. Examples include enforcing axial or rotational symmetry, or periodicity upon the model function. To showcase the procedure, axial symmetry in two dimensions can be enforced by applying the operator

$$L(f(\mathbf{x})) \;=\; \frac{f([x^1, x^2]^T) + f([-x^1, x^2]^T) + f([x^1, -x^2]^T) + f([-x^1, -x^2]^T)}{4} \tag{21}$$

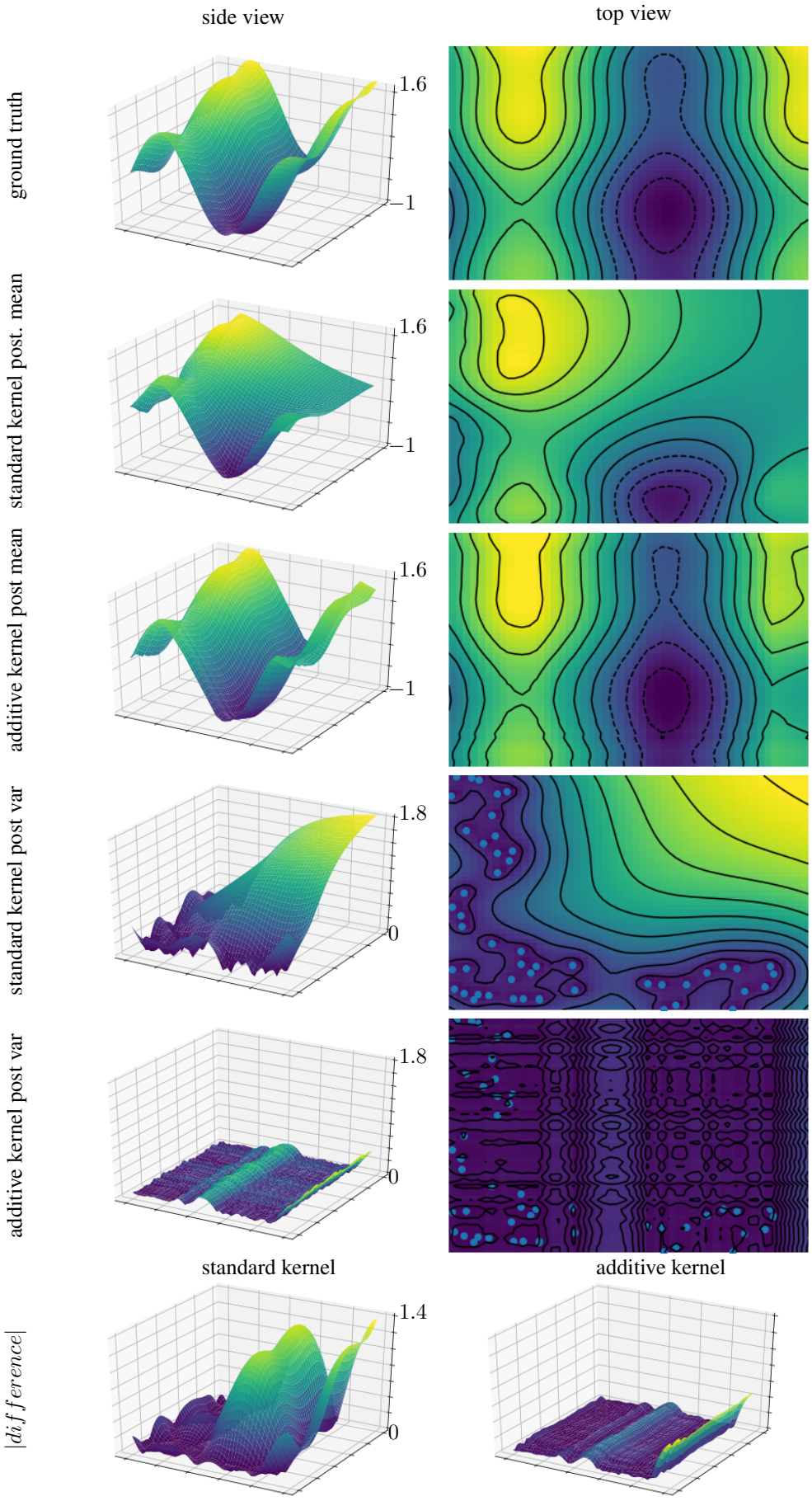


Figure 1: Standard kernel (Equation (16)) vs additive kernel (Equation (17)) function to approximate a function on $\mathcal{X} = [0,1] \times [0,1]$. Using the additive kernel means propagating information into regions where no data is available. The estimated variances are significantly smaller compared to the use of standard kernels.

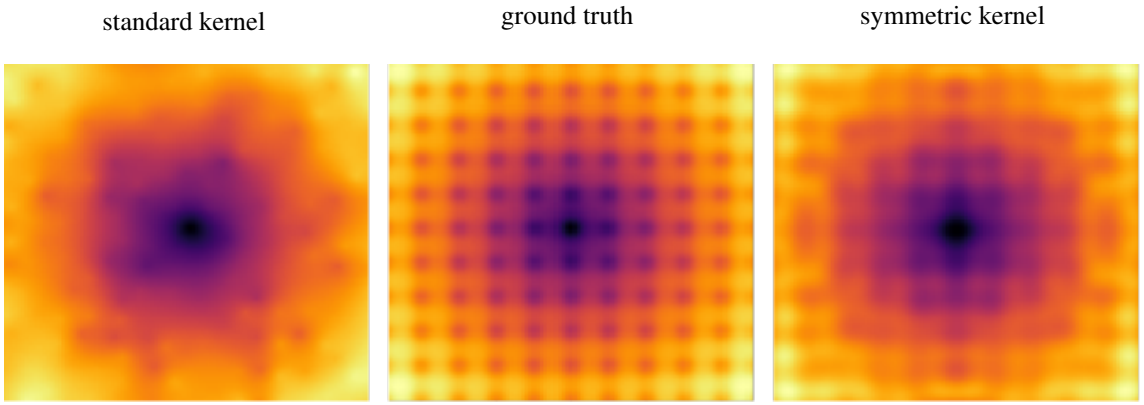


Figure 2: The powerful effect of kernel-based constraints on a GP regression. Displayed is Ackley's function (middle) and two GP posterior-mean functions. Left is the posterior mean calculated with an unconstrained Gaussian process. On the right is the posterior mean, calculated with imposed axial symmetry. In this example, the axial symmetry improves the uncertainty at a given number of measurements 4 fold and increases the computational speed 64 fold.

on a kernel function, which results in

$$\begin{aligned}
L_{\mathbf{x}_1}(k(\mathbf{x}_1,\mathbf{x}_2)) &= 1/4\,(k(\mathbf{x}_1,\mathbf{x}_2)+k([-x_1^1,x_1^2]^T,\mathbf{x}_2)\\
&\quad +k([x_1^1,-x_1^2]^T,\mathbf{x}_2)+k([-x_1^1,-x_1^2]^T,\mathbf{x}_2))\\
L_{\mathbf{x}_2}(k(\mathbf{x}_1,\mathbf{x}_2)) &= 1/4\,(k(\mathbf{x}_1,\mathbf{x}_2)+k(\mathbf{x}_1,[-x_2^1,x_2^2]^T)\\
&\quad +k(\mathbf{x}_1,[x_2^1,-x_2^2]^T)+k(\mathbf{x}_1,[-x_2^1,-x_2^2]^T))\\
&\quad \Rightarrow\\
L_{\mathbf{x}_2}(L_{\mathbf{x}_1}(k(\mathbf{x}_1,\mathbf{x}_2))) &= 1/16\,(k(\mathbf{x}_1,\mathbf{x}_2)+k([-x_1^1,x_1^2]^T,\mathbf{x}_2)\\
&\quad +k([x_1^1,-x_1^2]^T,\mathbf{x}_2)+k([-x_1^1,-x_1^2]^T,\mathbf{x}_2)\\
&\quad +k(\mathbf{x}_1,[-x_2^1,x_2^2]^T)\\
&\quad +k(\mathbf{x}_1,[x_2^1,-x_2^2]^T)+k(\mathbf{x}_1,[-x_2^1,-x_2^2]^T)\\
&\quad +k([-x_1^1,x_1^2]^T,[-x_2^1,x_2^2]^T)+k([-x_1^1,x_1^2]^T,[x_2^1,-x_2^2]^T)\\
&\quad +k([-x_1^1,x_1^2]^T,[-x_2^1,-x_2^2]^T)+k([x_1^1,-x_1^2]^T,[-x_2^1,x_2^2]^T)\\
&\quad +k([x_1^1,-x_1^2]^T,[x_2^1,-x_2^2]^T)+k([x_1^1,-x_1^2]^T,[-x_2^1,-x_2^2]^T)\\
&\quad +k([-x_1^1,-x_1^2]^T,[-x_2^1,x_2^2]^T)+k([-x_1^1,-x_1^2]^T,[x_2^1,-x_2^2]^T)\\
&\quad +k([-x_1^1,-x_1^2]^T,[-x_2^1,-x_2^2]^T)),
\end{aligned} \tag{22}$$

where $k(\mathbf{x}_1,\mathbf{x}_2)$ can be any kernel, for instance the anisotropic squared exponential kernel

$$k(\mathbf{x}_1,\mathbf{x}_2) = \exp\left[-\frac{\langle \mathbf{x}_1-\mathbf{x}_2,\mathbf{M},\mathbf{x}_1-\mathbf{x}_2\rangle}{l}\right]. \tag{23}$$

See Figure 2 for a presentation of the effects of such a kernel. To inform the GP about periodicity in x^2 direction, we can define the linear operator

$$L(f(\mathbf{x})) = \frac{f([x^1,x^2]^T)+f([x^1,x^2+p]^T)+f([x^1,x^2-p]^T)}{3}, \tag{24}$$

from which the following kernel can be derived

$$\begin{aligned}
&L_{\mathbf{x}_2}(L_{\mathbf{x}_1}(k(\mathbf{x}_1,\mathbf{x}_2))) =\\
&1/9\,(k(\mathbf{x}_1,\mathbf{x}_2)+k(\mathbf{x}_1,[x_2^1,x_2^2+p])^T+k(\mathbf{x}_1,[x_2^1,x_2^2-p])^T\\
&\quad +k([x_1^1,x_1^2+p]^T,\mathbf{x}_2)+k([x_1^1,x_1^2+p]^T,[x_2^1,x_2^2+p]^T)+k([x_1^1,x_1^2+p]^T,[x_2^1,x_2^2-p]^T)\\
&\quad +k([x_1^1,x_1^2-p]^T,\mathbf{x}_2)+k([x_1^1,x_1^2-p]^T,[x_2^1,x_2^2+p]^T)+k([x_1^1,x_1^2-p]^T,[x_2^1,x_2^2-p]^T)),
\end{aligned} \tag{25}$$

where p is the period. Figure 3 presents a posterior-mean function that results from such a kernel.

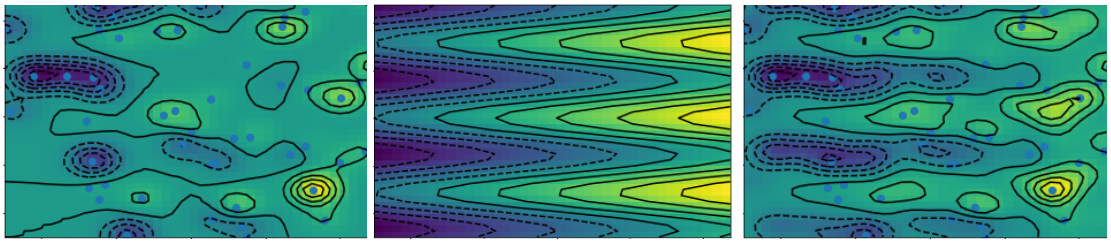


Figure 3: Posterior mean function given 50 data points for a standard GP on the left and a GP informing the posterior mean about periodicity on the right. The ground truth can be seen in the center. The blue points show the measurement locations. The periodicity from the kernel is enforced and used to inform the posterior mean in places without data.

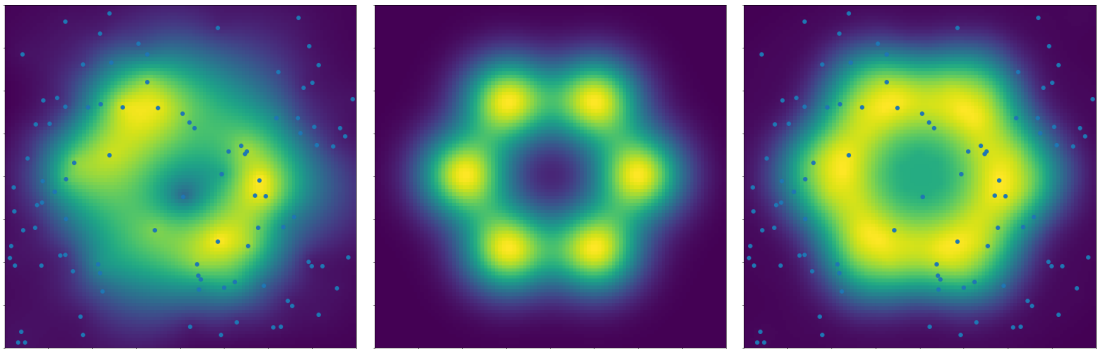


Figure 4: Six-fold symmetric test function (center) and GP approximations. The image on the left shows the standard GP posterior mean using the squared-exponential kernel. On the right, we see the GP approximation using kernel (26). All points are implicitly used six times increasing the amount of information used for the regression six-fold.

One could argue that axial symmetry is more of academic than of practical interest, since the knowledge about axial symmetry can simply be accounted for by limiting the domain for probing the function. However, the general principle can be extended to more complicated symmetries, such as rotational symmetry. For instance, the kernel for six-fold symmetry in two dimensions is defined as

$$k(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{36} \sum_{\phi \in p\pi/3} \sum_{\theta \in q\pi/3} \tilde{k}(\mathcal{R}_\phi \mathbf{x}_1, \mathcal{R}_\theta \mathbf{x}_2); \; p, q \in \{0, 1, 2, 3, 4, 5\}, \tag{26}$$

where $\tilde{k}$ is any valid stationary kernel, ϕ and θ are angles, and $\mathcal{R}$ is a rotation matrix rotating the vector $\mathbf{x}$ by the specified angle. The result of such a kernel can be seen in Figure 4.

2.5 The Essence of Stationary Kernels vs Non-Stationary Kernels

The word kernel stems from the theory of integral operators. See reference [15] for more explanation on the origin of kernels and the connection to integral operators. As mentioned before, stationary kernels are of the form

$$k = k(||\mathbf{x}_1 - \mathbf{x}_2||), \tag{27}$$

i.e. they are function of a norm placed on $\mathcal{X}$. GPs (and other stochastic processes) that use covariance functions (kernels) that only depend on the distance of data points and not on their respective locations are referred to as stationary.

In contrast, non-stationary kernels are more general symmetric p.s.d. functions of the form

$$k = k(\mathbf{x}_1, \mathbf{x}_2), \tag{28}$$

where the kernel function now contains as arguments the location of the data points, and hence relaxes the restriction so that $k = k(\mathbf{x}_1, \mathbf{x}_2) \neq k(||\mathbf{x}_1 - \mathbf{x}_2||)$.

Stationary and non-stationary kernels are both symmetric positive-definite functions, since they induce inner

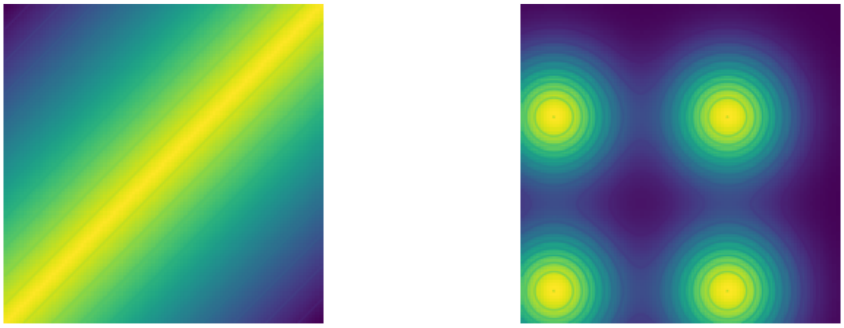


Figure 5: A top view onto a stationary (left) and non-stationary (right) kernel function over $\mathcal{X} \times \mathcal{X}$. Both functions are symmetric; however, while the stationary kernel function is constant along the diagonals, the non-stationary kernel function has no such restriction. Non-stationary kernels are therefore a much more flexible inner product in $\mathcal{H}$, which translates into a more flexible similarity measure. This added flexibility can be used to communicate information into remote corners of the index set, to constrain the posterior mean, and to enable expressive multi-task learning.

products in $\mathcal{H}$. The difference between stationary and non-stationary kernels can be illustrated visually via a one-dimensional example. If we consider $\mathcal{X} \subset \mathbb{R}^1$ and therefore $f = f(x)$, we can illustrate the kernel as a function over $\mathcal{X} \times \mathcal{X}$ (see Figure 5). Stationary kernel functions are constant along diagonals, unlike non-stationary kernels. This characteristic of non-stationary kernels translates into potentially highly flexible inner products, and therefore similarity measures, which can encode varying covariances within the input space, the output space and across the two spaces.

2.6 Advanced Non-Stationary Kernel Designs

Non-stationary kernels have the additional flexibility that they depend on the location of the input points, not only on the distance between them (see Equation (28)). This gives a learning algorithm powerful additional capabilities since the similarity measure between data can vary substantially across $\mathcal{X} \times \mathcal{X}$ (see Figure 5). Stationarity is an approximation that almost never holds in real data sets. Imagine a regression model of the topography of the United States. While correlation lengths in the Sierra Nevada and in the Rocky Mountains are in the order of miles, they will be hundreds of miles in the Great Plains. Using a stationary kernel would perform poorly in such a scenario. Non-stationary kernels, on the other hand, can capture the varying length scales and lead to accurate function approximations. This extra flexibility is also useful for multi-output Gaussian processes in which distances between tasks are arbitrary and any stationary choice would limit the method's ability to learn. In fact, we will see how flexible non-stationary kernels can replace tailored methods for multi-output GPR.

When designing advanced non-stationary kernels, we have to show that the resulting kernel functions are positive semi-definite, just like in the stationary case. However, it is often difficult to prove positive semi-definiteness in closed form for general functions. Instead, one can induce positive semi-definiteness by taking advantage of the fact that, as mentioned earlier, the set of kernels is closed under addition, multiplication, and linear transformations. In addition, we can show that:

Theorem 2. *Let $k(\mathbf{x}_1, \mathbf{x}_2)$ be a valid kernel, then $f(\mathbf{x}_1)f(\mathbf{x}_2)k(\mathbf{x}_1, \mathbf{x}_2)$ is also a valid kernel according to Definition 1. Here, $f(\mathbf{x})$ is an arbitrary function.*

Proof. $\sum_i^N \sum_j^N b_i\, b_j\, k(\mathbf{x}_i, \mathbf{x}_j) \geq 0\, \forall N,\, \mathbf{x} \in \mathbb{R}^N,\, \mathbf{b} \in \mathbb{R}^N$
$\Rightarrow \sum_i^N \sum_j^N f(\mathbf{x}_i)\, f(\mathbf{x}_j)\, k(\mathbf{x}_i, \mathbf{x}_j) \geq 0\, \forall N,\, \mathbf{x} \in \mathbb{R}^N$ □

Since a constant is a valid kernel, $f(\mathbf{x}_1)f(\mathbf{x}_2)$ has to be a valid kernel too.

The kernel $f(\mathbf{x}_1)f(\mathbf{x}_2)k(\mathbf{x}_1, \mathbf{x}_2)$ represents a trade-off between flexibility and simplicity and is used in our examples. Figure 6 shows this particular kernel for a linear function f in one dimension, and underscores how using a stationary kernel leads to under-estimated and over-estimated posterior variances when the length scale varies across $\mathcal{X}$. In the example seen in Figure 7, we are using the kernel presented in Theorem

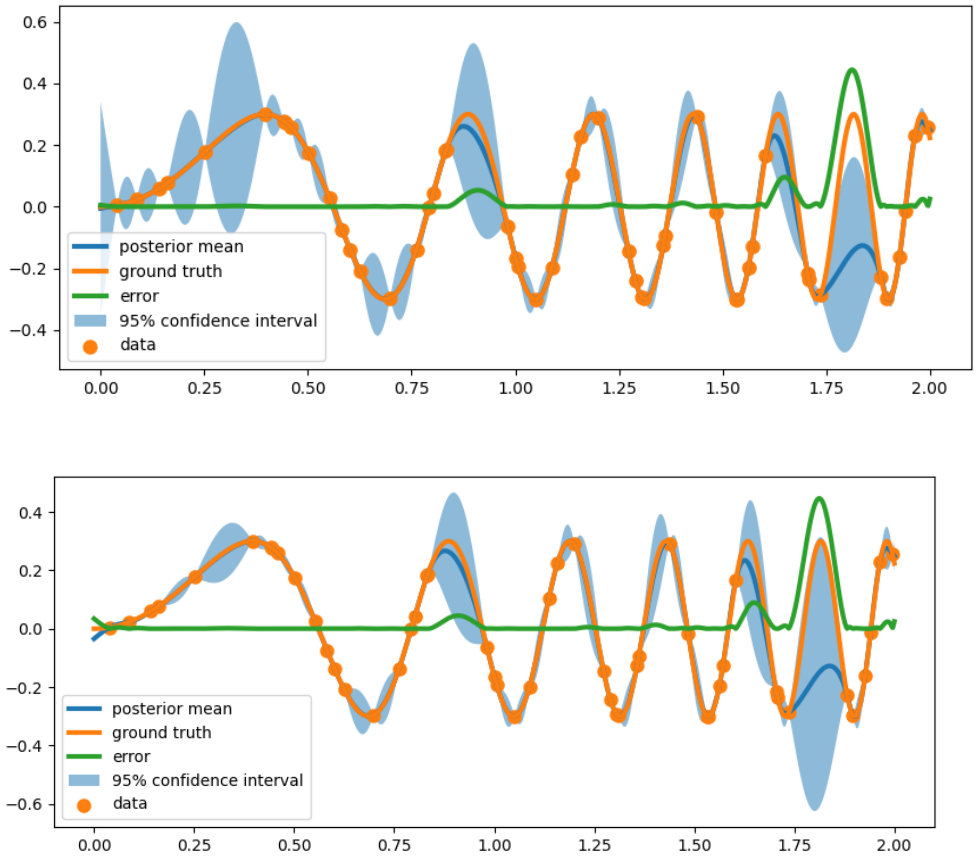


Figure 6: Comparison of a one-dimensional result of a Gaussian process using stationary (top) and non-stationary (bottom) kernels. In this figure, we use the kernel from Theorem 2 with a linear function, i.e. $k(x_1, x_2) = x_1 x_2 k_{matern}$. The stationary-kernel Gaussian process (top) significantly overestimates posterior variances on the left and underestimates them on the right. This is due to the fact that the similarity at a given distance is averaged across the domain. The non-stationary-kernel Gaussian process uses the location of points to compute the similarity and can therefore estimate the posterior variances more accurately.

2 with

$$f(\mathbf{x}) \;=\; (\phi_1\,(\sqrt{50} - ||\mathbf{x}||)) + \phi_2. \tag{29}$$

The result shows the incorrectly-estimated variances when the stationary kernel is used.

A particularly flexible kernel, introduced by [8] and reformulated and enhanced by [10], is defined as

$$k(\mathbf{x}_1, \mathbf{x}_2) \;=\; \frac{\sigma_s^2(\mathbf{x}_1)\,\sigma_s^2(\mathbf{x}_2)}{\sqrt{|\frac{\mathbf{\Sigma}(\mathbf{x}_1)+\mathbf{\Sigma}(\mathbf{x}_2)}{2}|}} \mathcal{M}(\sqrt{Q(\mathbf{x}_1, \mathbf{x}_2)}) \tag{30}$$

where

$$Q(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 - \mathbf{x}_2)^T \left(\frac{\mathbf{\Sigma}(\mathbf{x}_1) + \mathbf{\Sigma}(\mathbf{x}_2)}{2}\right)^{-1} (\mathbf{x}_1 - \mathbf{x}_2), \tag{31}$$

where $\mathcal{M}$ is the Matérn kernel. This kernel allows for non-constant signal variances, length scales and anisotropies. In addition, it takes only a small adjustment to vary the differentiability of the model within $\mathcal{X}$. The presented non-stationary kernel designs, together with an efficient way to find the hyperparameters, renders specialized techniques for multi-task GPs obsolete. The difference between single-task and multi-task Gaussian processes can be entirely contained within the kernel design as will be discussed in the next section.

2.7 Using Flexible Non-Stationary Kernels for Multi-Task GPR

The challenge of multi-task GPs is the missing natural distance between different tasks ($\in \mathcal{X}_o$). At the same time, the kernel function that defines covariances between data points depends on those distances.

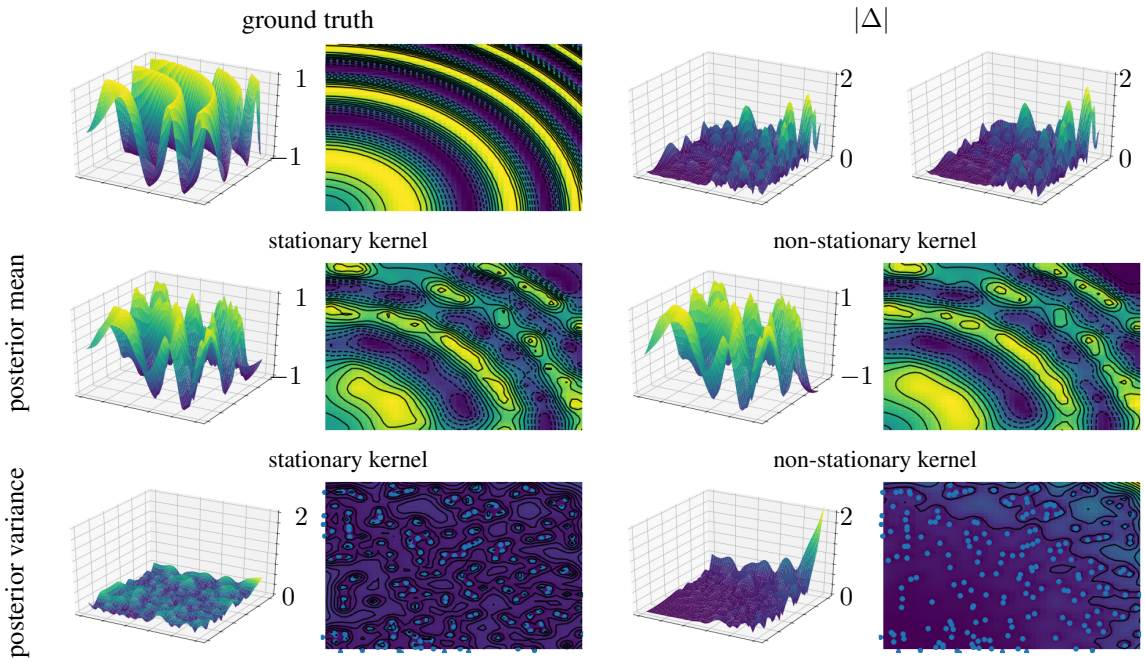


Figure 7: Comparison of a two-dimensional GP approximation of a function over $\mathcal{X} = [0, 5] \times [0, 5]$ using stationary and non-stationary kernels. Similar to what we have seen in Figure 6, the posterior variance is significantly overestimated close to the origin where the frequency of the function is low. The farther we move away from the origin, the more does the stationary-kernel GP underestimate the posterior variance. This is of significant impact, for instance, for optimal data-acquisition strategies where the posterior variance plays a large role in the choice of the next measurement.

Therefore, the point positions to calculate distances are often chosen arbitrarily and equidistant, for instance as indices $\{0, 1, 2, 3, 4\}$. A stationary kernel in task-direction can then be defined that assigns a constant covariance contribution for each task pair. However, covariances between tasks might depend on $\mathbf{x}_i \in \mathcal{X}_i$ and $\mathbf{x}_o \in \mathcal{X}_o$, i.e., the positions in the input and output space, which induces a dependence on positions and therefore non-stationarity. Stationary kernels are unable to encode these non-constant covariance contributions. To avoid this issue, early on, several tasks would just be seen as independent Gaussian processes. Clearly, in this approach, cross-task covariances are ignored and learning can only happen within a task. To circumvent this problem, several techniques have been proposed. In the following overview, we are following the survey in [1], which broadly divides the multi-output regression methods into "problem transformation methods" and "algorithm adaption methods". Algorithm adaption methods are based on additional techniques and methodologies, such as support vector regression, to encode the correlation between tasks. It is able to capture cross-task correlations but, as the name suggests, needs the adaption of the basic algorithm and the theory, which often leads to new issues. Problem transformation methods, on the other hand, are based on transforming the problem into several single-task problems, creating separate models for them, and concatenating all the models. It is often criticized for not capturing the intricate correlations between the tasks. Problem transformation methods leave the basic theory of GPs intact and are therefore more general, widely applicable, and avoid common problems like missing data in one or more tasks.

We argue that the GP framework is, without alteration, able to account for the correlation between multiple tasks. For that, we draw attention to the fact that problem transformation methods in combination with flexible non-stationary kernels do not suffer from the limited ability to encode cross-task covariances. Problem transformation methods commonly make use of separable, stationary kernels that fail to encode non-constant cross-task covariances due to arbitrarily-fixed distances between tasks. We can address this issue by using flexible kernel definitions, we present in this paper. Using advanced non-stationary kernels liberates us from the problems of multi-output Gaussian processes; we can assume a constant, arbitrary distance between the tasks and the kernel will learn how these distances translate into similarities as a function on the index set $\mathcal{X}$. This leaves the basic theory of GPs untouched, and is therefore robust against common multi-task GP problems such as missing data in a subset of tasks or poor interpretability. The advanced kernels needed for the extra flexibility come at the cost of many hyperparameters we have to find, which

however, can be countered with clever optimization procedures.

As alluded to earlier, instead of interpolating a vector-valued function over the input space, we approximate a scalar function $f(\mathbf{x})$ on $\mathcal{X} = \mathcal{X}_i \times \mathcal{X}_o$. The norm on the RKHS induces a metric which is entirely defined by the kernel and therefore extends also into the output space. Therefore, a flexible kernel overcomes the challenge of arbitrarily defined distances between tasks. In this framework, we assume $\mathcal{X}_o$ to be a subset of the index set which leads us to refer to this special kind of multi-task Gaussian processes as function-valued Gaussian processes (fvGP) — one could imagine the output of an evaluation to be itself a function over $\mathbb{R}^n$. To reiterate the key takeaway of this section, the main difference of a multi-output, or function-valued, GP and a single-task GP is the choice of the kernel. The main problem with this approach is the vastly increased number of hyperparameters that have to be found. This issue will briefly be discussed in the next section.

In Figure 8, we define two different tasks that show a particularly high correlation between the circled areas. This correlation is reflected in the covariance matrix. The kernel for this example is defined in Equations (30) and (31) with

$$\begin{aligned}
\boldsymbol{\Sigma}(\mathbf{x}) &= \begin{pmatrix} l(\mathbf{x}_1, \mathbf{x}_2) & 0 & 0 \\ 0 & l(\mathbf{x}_1, \mathbf{x}_2) & 0 \\ 0 & 0 & l(\mathbf{x}_1, \mathbf{x}_2) \end{pmatrix} \\
l(\mathbf{x}_1, \mathbf{x}_2) &= h_1 + (h_2 \left(\exp\left[(\mathbf{x}_1 - \mathbf{h}_1)^T \boldsymbol{M} (\mathbf{x}_1 - \mathbf{h}_1)\right] + \exp\left[(\mathbf{x}_2 - \mathbf{h}_2)^T \boldsymbol{M} (\mathbf{x}_2 - \mathbf{h}_2)\right]\right)) \\
\boldsymbol{M} &= \begin{pmatrix} h_3 & 0 & 0 \\ 0 & h_3 & 0 \\ 0 & 0 & h_3 \end{pmatrix},
\end{aligned} \tag{32}$$

where all h_i and $\mathbf{h}_i$ are found by the training process. The results in Figure 8 show that a flexible non-stationary kernel can encode complicated non-local covariances that will be used for the approximation and uncertainty quantification.

3 A Note on Optimizing the Marginal Log-Likelihood when using Advanced Kernel Designs

As mentioned throughout this paper, the main issue that accompanies advanced kernel designs is the number of hyperparameters we have to find. We can think of the hyperparameters as a vector $\boldsymbol{\phi} \in \mathbb{R}^n$. When using standard kernel definitions, n is often two or three. This number is significantly larger for advanced stationary and especially non-stationary kernels. We have shown that we can invoke functions over $\mathcal{X}$ into the kernel definitions. These functions can be defined as the sum of arbitrary basis functions. Their coefficients and possibly locations are also hyperparameters and have to be found. This example makes clear that the number of hyperparameters n can quickly rise to numbers that make the marginal-log-likelihood optimization a lengthy procedure. To find the global or a high-quality local optimum, an optimization of this scale commonly needs many function evaluations to succeed. In addition, each function evaluation of the marginal log-likelihood is potentially costly since it involves an inversion or a system solve, and a log-determinant computation. In a fully Bayesian approach, finding the posterior distribution of the hyperparameters faces the same challenges. One possible solution to the problem is the use of hybrid optimization algorithms that can run in parallel to the GP prediction and can provide best-estimate optima whenever queried. The log-likelihood function, its gradient and the Hessian evaluations can be accelerated using GPU computer architectures. In addition, we can start many local searches in parallel — and remove the found optima by deflation — to take full advantage of HPC computer architecture [4, 7].

4 Experiments

In this section, we want to show the potential impact of the proposed kernel designs on two scientific experiments, namely neutron scattering and IR spectroscopy. The shown data has been collected at the Thales instrument at the Institute Laue-Lagevin (ILL) in France, and at the Berkeley Synchrotron Infrared Structural Biology (BSISB) beamline at the Advanced Light Source (ALS) at Lawrence Berkeley National Laboratory (LBNL) in Berkeley, California. At these instruments, our work on Gaussian Processes is used for autonomous data acquisition and general analysis and interpretation purposes [6]. We show how the

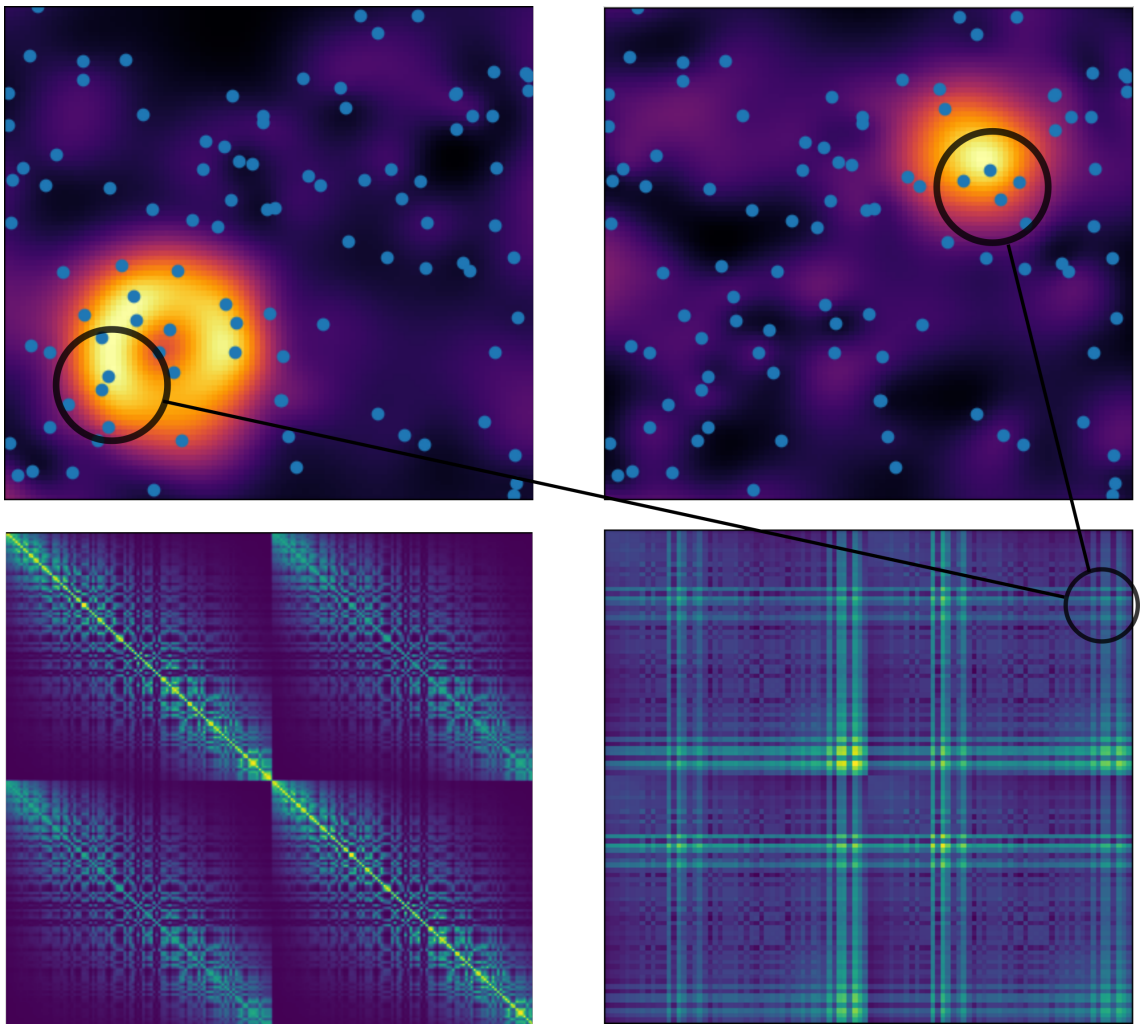


Figure 8: Representation of two tasks (top) and the associated covariance matrices (bottom) of a multi-task Gaussian process using 100 random data points. In the bottom left, the covariance is computed using a stationary kernel; while cross-covariances are not ignored (quadrant 1 and 3), they are just offset by a constant. The covariance in the bottom right is defined by a non-stationary kernel. This allows the Gaussian process to learn that the circled regions are correlated. The non-stationary kernel used in this example is shown in Equations (30) and (31), using the terms defined in Equation (32). The central point here is that the fundamental difference between single-task and multi-task Gaussian processes lies in the kernel design. Note the checkerboard pattern; the length scale l is the sum of two Gaussian functions. If a point is close to one of them, the covariance with all other data points will be comparatively large. However, the covariance reaches its maximum when both points are located at the center of one or the other Gaussian function.

presented improvements of kernel designs can advance the use of Gaussian processes in these experiments, and influence the experimental design and the resulting model approximation.

4.1 IR Spectroscopy

Infrared (IR) imaging spectroscopy employs full infrared spectra in order to study materials and biological samples. This is done by directing an infrared beam onto the sample at a point (x_1, x_2). Therefore, we can define the input space $\mathcal{X}_i \subset \mathbb{R}^2$ with outputs composed of entire spectra at selected points in the input set, i.e., $\mathcal{X}_o \subset \mathbb{R}^1_+$. As before, the final index set is defined by $\mathcal{X} = \mathcal{X}_i \times \mathcal{X}_o \subset \mathbb{R}^3$. We will assume that a spectrum is represented by 87 intensity values at a set of wave numbers. This example illustrates the duality of a multi-task GP over $\mathcal{X}_i$ and a single-task GP over $\mathcal{X}_i \times \mathcal{X}_o$, and that the difference can be contained within the used kernel. Here, the "tasks" have a natural distance between them, with the unit cm^{-1} of a wave number. This is not always the case. Instead of spectra, we could approximate the PCA components of spectra, which do not have a natural distance. The stationary kernel, we are using for this example, is defined as

$$\begin{aligned} k(\mathbf{x}_1, \mathbf{x}_2) &= k(|\mathbf{x}_1 - \mathbf{x}_2|) = \\ &\sigma^2 \, k_{exp}\Big(|\begin{bmatrix} x_1^1 \\ x_1^2 \end{bmatrix} - \begin{bmatrix} x_2^1 \\ x_2^2 \end{bmatrix}|\Big) \, M(|x_1^3 - x_2^3|), \end{aligned} \tag{33}$$

where k_{exp} is the exponential kernel and M is a Matérn kernel. The Euclidean distance in the exponential kernel is anisotropic. The non-stationary kernel is defined by

$$\begin{aligned} &k(\mathbf{x}_1, \mathbf{x}_2) = \\ &\sigma^2 \, (k_{exp}\Big(|\begin{bmatrix} x_1^1 \\ x_1^2 \end{bmatrix} - \begin{bmatrix} x_2^1 \\ x_2^2 \end{bmatrix}|, \phi_7, \phi_8\Big) \, M(|x_1^3 - x_2^3|, \phi_9) + A_1 \, A_2), \\ &A_1 = \exp\left[(x_1^3 - p_1(\mathbf{x}))^2\right]/\phi_6 + \exp\left[(x_1^3 - p_2(\mathbf{x}))^2\right]/\phi_6 \\ &A_2 = \exp\left[(x_2^3 - p_1(\mathbf{x}))^2\right]/\phi_6 + \exp\left[(x_2^3 - p_2(\mathbf{x}))^2\right]/\phi_6 \\ &p_1(\mathbf{x}) = \phi_0 \, (\phi_1 \, x^1) + (\phi_2 \, x^2) \\ &p_2(\mathbf{x}) = \phi_3 \, (\phi_4 \, x^1) + (\phi_5 \, x^2), \end{aligned} \tag{34}$$

where ϕ is a set of hyperparameters. The focus in this expression is on A, which allows the covariance to depend on two Gaussian functions that can change position in $\mathcal{X}_o$ as a linear function in $\mathcal{X}_i$. Figure 9 shows that the Gaussian process takes advantage of the given additional flexibility provided by the non-stationary kernel and thereby lowers the overall approximation error.

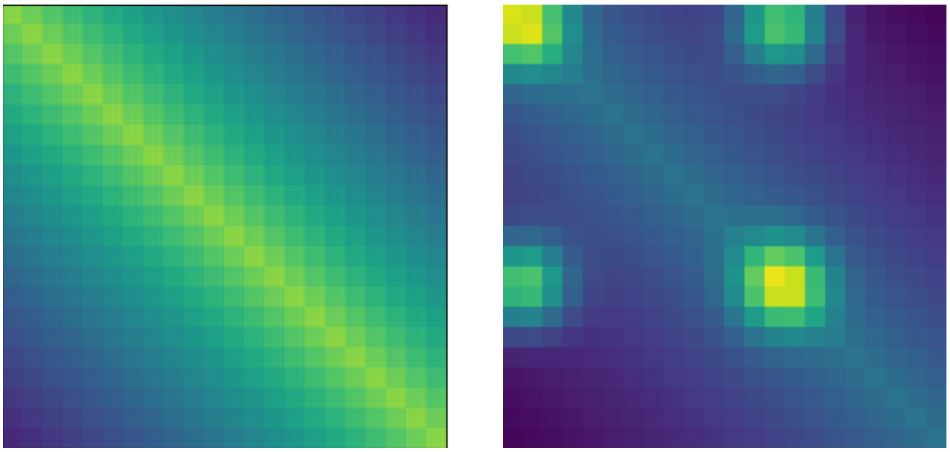
4.2 Neutron Scattering

Neutron scattering is an experimental technique to obtain detailed information about the arrangements of atoms in condensed matter. The data showcasing symmetry comes from the Thales (Three Axis Low Energy Spectrometer) at ILL [14]. The measurements probe a function $S(q_h, q_l, q_k, E)$ which is often symmetric around one or more axes. Figure 10 shows how effectively kernels that impose symmetry (Equation (22)) can be used to approximate the model function and to steer data acquisition in the presence of symmetry. The higher-quality approximation translates into less points that are needed for a targeted model accuracy.

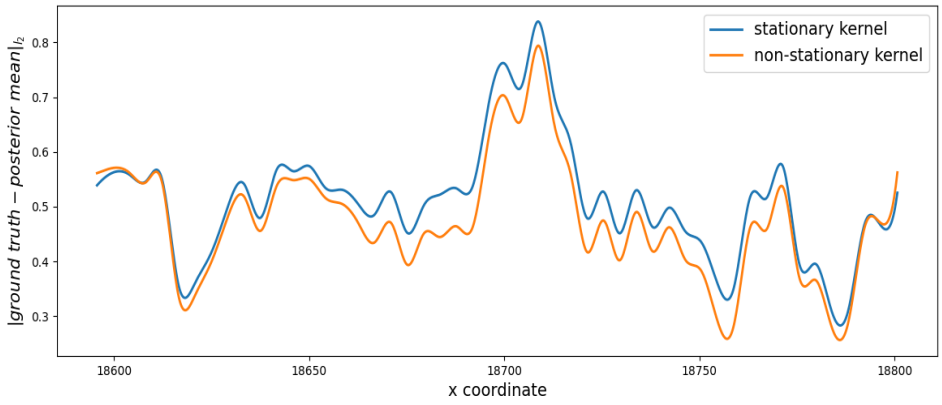
5 Discussion and Conclusion

In this paper, we presented some known and new kernel designs which are of interest for practitioners using Gaussian Processes (GPs). The presented kernels are able to significantly reduce uncertainty of the model, given a number of data points. This was either achieved by using stationary kernels that implicitly subject the posterior mean to hard constraints, such as periodicity or symmetry, or by non-stationary kernels that are able to encode flexible inner products which translate into the ability to learn more complicated heterogeneous covariances across the input space $\mathcal{X}$. This led to a flexible and natural formulation of multi-task GPs.

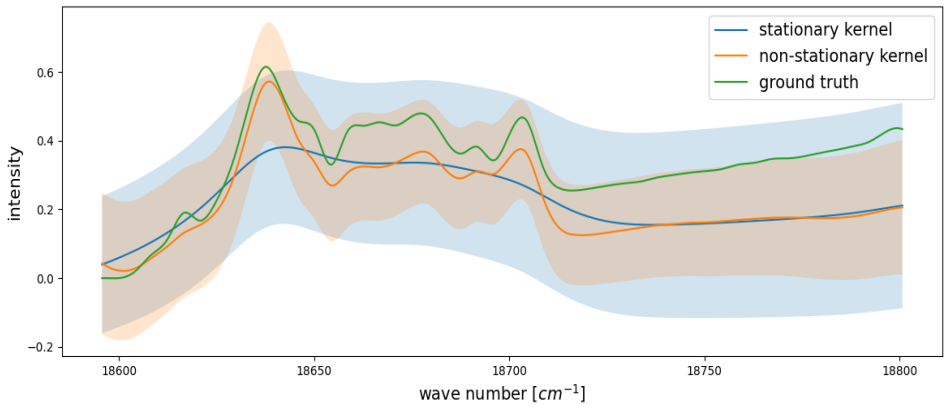
Using the appropriate kernels, knowledge that the model function is additive results in a GP that can propagate information from subsets of the domain to infinity (Fig. 1). Multiplicative kernels have no such property, but are perfectly suited for allowing axial anisotropy in $\mathcal{X}$.



(a) (b)



(c)



(d)

Figure 9: Presentation of a GP used for interpolating infrared spectroscopy data deploying stationary and non-stationary kernels. (top) Simplified view of the covariance matrices resulting from stationary (a) and non-stationary (b) kernel definitions (Equations (33) and (34) respectively). The covariance matrix resulting from the stationary kernel (a), cannot identify differing similarities between tasks when their distance is constant; therefore we see a diagonal pattern of the covariance matrix. In (b) we see the covariance matrix resulting from a non-stationary kernel which is able to identify similarities between any two tasks independently. Tasks in this case can be understood as the spectrum intensity at a particular wave number. (c) The average of the Euclidean distance between the posterior means and the ground truth along x. The GP using the non-stationary kernel performs significantly better. (d) Posterior means and the ground truth of a representative spectrum. Not only is the approximation using the non-stationary kernel significantly more accurate, the posterior variance is overall smaller and more detailed.

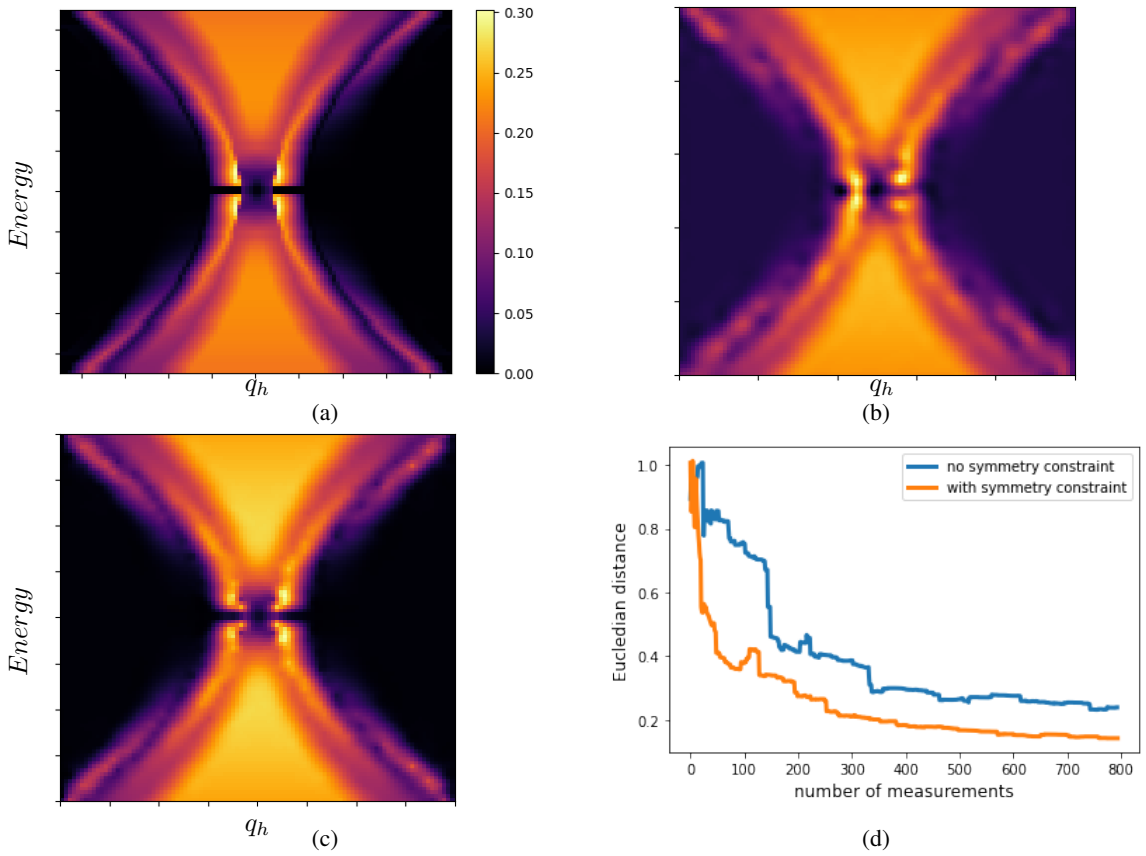


Figure 10: Figure displaying the importance of symmetry in neutron scattering data acquisition. The function $S = S(q_h, q_l, q_k, E)$ is symmetric around $E = 0$ for the slice of interest. Exploiting this fact as a constraint, which is enforced by advanced kernel design, increases the accuracy of the GP interpolation significantly. (a) The ground truth function $S(q_h, q_l = 0, q_k = 0, E)$. (b) The GP posterior mean when a standard exponential kernel is used. (c) The GP posterior mean with a kernel that enforces symmetry. (d) The Euclidean-distance error of both approximations as a function of the number of measurements.

Figure 2 showed how the quality of the posterior mean can increase when symmetry is present and accounted for by the kernel. The same was shown for experimental data in Figure 10. Symmetry implicitly increases the amount of information that is used in the prediction. For instance, in case of axial symmetry around the x and y axes, every data point contains four times the amount of information – compared to the use of a standard kernel — decreasing the computational cost for a given function-approximation problem by 64 (assuming $O(N^3)$ scaling). We have shown that periodicity can be accounted for in the same manner. Note that the imposed periodicity is not the same as a sine kernel, since imposing periodicity does not impose any particular functional shape.

Lastly, we investigated and drew attention to non-stationary kernels and their impact on a GP. We have seen that a GP with a constant length scale is prone to both over- and underestimating the posterior variance (Figures 6 and 7) which has major implications for decision-making algorithms which use the posterior covariance, e.g., for optimal experiment design. Non-stationary kernels can also be used to obtain a flexible and simple implementation of multi-output GPs. In an experimental setting, the non-stationary kernels led to an overall high-quality approximation (Fig. 9). We have rediscovered that the main difference between single and multi-task GPs can be entirely contained within the kernel design, which leaves the basic theory of GPs untouched.

The use of advanced kernel designs does not come without issues, especially in the multi-task setting. The biggest issue is the added computational costs. Clearly, the marginal log-likelihood optimization becomes a much more involved process as the number of hyperparameters increases. But there are ways to overcome this shortcoming. Traditional GP training uses standard optimization procedures to find the hyperparameters, such as multi-start gradient descent. When the optimization has to find more hyperparameters, more sophisticated, HPC-ready algorithms have to be used. In an optimal and sequential design setting, the optimization can happen asynchronously, so that the costs of the training are hidden.

ACKNOWLEDGMENT

The work 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 Department of Energy's Office of Science, under Contract No. DE-AC02-05CH11231. The data was provided by the BSISB program (DOE No. DE-AC02-05CH11231) and the Thales project at ILL, France. We want to thank the groups for the collaboration and data.

AUTHOR CONTRIBUTION

M.M.N developed the mathematics and wrote the first draft of the manuscript. J.A.S. supervised the work, verified the correctness of the mathematical derivations and revised the manuscript.

REFERENCES

[1] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233, 2015.

[2] David Ginsbourger, Nicolas Durrande, and Olivier Roustant. Kernels and designs for modelling invariant functions: From group invariance to additivity. In *mODa 10–Advances in Model-Oriented Design and Analysis*, pages 107–115. Springer, 2013.

[3] Janine Matschek, Andreas Himmel, Kai Sundmacher, and Rolf Findeisen. Constrained gaussian process learning for model predictive control. *arXiv preprint arXiv:1911.10809*, 2019.

[4] Marcus M Noack and Simon W Funke. Hybrid genetic deflated newton method for global optimisation. *Journal of Computational and Applied Mathematics*, 325:97–112, 2017.

[5] Marcus M Noack et al. Autonomous materials discovery driven by gaussian process regression with inhomogeneous measurement noise and anisotropic kernels. *Scientific Reports*, 10:17663, 2020.

[6] Marcus M et al. Noack. Autonomous data acquisition for large scale facilities. *Under Review*, 2021.

[7] Marcus Michael Noack, David Perryman, Harinarayan Krishnan, and Petrus H Zwart. High-performance hybrid-global-deflated-local optimization with applications to active learning. In *2021 3rd Annual Workshop on Extreme-scale Experiment-in-the-Loop Computing (XLOOP)*, pages 24–29. IEEE, 2021.

[8] Christopher J Paciorek and Mark J Schervish. Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics: The official journal of the International Environmetrics Society*, 17(5):483–506, 2006.

[9] Karl Ezra Pilario, Mahmood Shafiee, Yi Cao, Liyun Lao, and Shuang-Hua Yang. A review of kernel methods for feature extraction in nonlinear process monitoring. *Processes*, 8(1):24, 2020.

[10] Mark D Risser and Catherine A Calder. Regression-based covariance functions for nonstationary spatial modeling. *Environmetrics*, 26(4):284–297, 2015.

[11] Laura Swiler, Mamikon Gulian, Ari Frankel, Cosmin Safta, and John Jakeman. A survey of constrained gaussian process regression: Approaches and implementation challenges. *arXiv preprint arXiv:2006.09319*, 2020.

[12] Mark van der Wilk, Vincent Dutordoir, ST John, Artem Artemev, Vincent Adam, and James Hensman. A framework for interdomain and multioutput gaussian processes. *arXiv preprint arXiv:2003.01115*, 2020.

[13] Xiaojing Wang and James O Berger. Estimating shape constrained functions using gaussian processes. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1–25, 2016.

[14] Tobias Weber, Johannes Waizner, Paul Steffens, Andreas Bauer, Christian Pfleiderer, Markus Garst, and Peter Böni. Polarized inelastic neutron scattering of nonreciprocal spin waves in mnsi. *Physical Review B*, 100(6):060404, 2019.

[15] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[16] Kai Yu, Volker Tresp, and Anton Schwaighofer. Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on Machine learning*, pages 1012–1019, 2005.