UNCECOMP 2023
5th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Athens, Greece, 12—14 June 2023

LEARNING FUNCTIONS DEFINED OVER SETS OF VECTORS WITH KERNEL METHODS

Babacar SOW[1,2], Rodolphe LE RICHE[2], Julien PELAMATTI[3], Sanaa ZANNANE[3], Merlin KELLER[3]

[1]Ecole Nationale Supérieure des Mines de SAINT-ETIENNE
[2] CNRS LIMOS
158 cours Fauriel CS 62362 42023 Saint-Etienne cedex 2, France
e-mail: babacar.sow@emse.fr

[3] EDF R&D
6 Quai Watier, 78400 Chatou
e-mail: {julien.pelamatti,sanaa.zannane,merlin.keller}@edf.fr

Keywords: Surrogate models, Gaussian process, Semi-definite positive kernel, Sets of vectors, Clouds of points.

Abstract. *We consider the problem of learning time-consuming functions defined over unordered sets of vectors. Such functions arise frequently, in particular in the context of networks of devices whose number is not fixed and that interact with each other. A working example is the modeling of a wind farm. Unordered sets of vectors are a mix of integer and continuous input variables suitable for functions that are permutation-invariant. The time-consuming aspect of the functions is, classically, treated by approximating them with a Gaussian process.*
This study addresses the problem of defining valid and efficient covariance kernels over clouds of points in the context of Gaussian process surrogate modeling.
We review methods for defining such kernels. These kernels are compared on a set of analytical functions inspired from different engineering problems, such as the design of experiments and the modeling of wind farms production. The extrapolation properties of the kernels are tested on geometrically transformed clouds.
We show that modeling 2D clouds of points as supports of discrete uniform distributions should be preferred to a Gaussian representation of the clouds. A detailed investigation of the good performance of MMD-based kernels illustrates how they adapt their hyperparameters to the geometrical properties of the studied functions.

Main notations and abbreviations

- Bhatta, Bhattacharyya kernel, probability product kernel with Gaussian distributions, see (11).
- $d \in \mathbb{N}$, dimension of a vector.
- $d_\psi(,)$ a metric composed with the mapping ψ.
- δ_a, Dirac function, $\mathbb{R}^d \to \{1, 0\}, \mathbf{x} \longmapsto$ δ_a is 1 if $\mathbf{x} = a$, 0 otherwise.
- DoE, Design of Experiments.
- $\mathcal{F}$ set of functions over clouds of points, $f : \mathcal{X} \to \mathbb{R}, X \mapsto f(X)$.
- Gauss-Wass, substitution kernel with Wasserstein distance between Gaussian distributions, see (4).
- θ hyperparameters, they can be scalars or vectors.
- Θ vector of hyperparameters.
- K a kernel or covariance function, $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.
- $k_{\mathcal{H}}$ is a kernel used in the calculation of the mean embedding of a distribution.
- MMD, Maximum Mean Discrepancy.
- MMD, substitution kernel with MMD (between discrete uniform distributions), see (8).
- μ_X mean embedding of a measure associated to X.
- n-MeanMap, normalized Mean Map Kernel, see (10).
- n, $n_{\text{min}}, n_{\text{max}} \in \mathbb{N}$, sizes of vector sets, $n_{\text{min}} \leq n \leq n_{\text{max}}$.
- P any distribution defined over $\mathbb{R}^d$.
- P_X a distribution associated to a set of points X.
- RBF, Gaussian Radial Basis Function kernel, also known as squared exponential or Gaussian kernel.
- RFK, Relevant Feature Map kernel, see (2.4).
- RKHS, Reproducing Kernel Hilbert Space.
- Slice-Wass, substitution kernel with Sliced-Wasserstein distance, see (6).
- $X \in \mathcal{X}$ is a set of n unordered points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$. Cloud of points. X is invariant with respect to any point permutation.
- ψ a mapping from $\mathcal{X}$ to a metrical Space.
- $\mathcal{X}$ set of clouds of points (X).

We use indifferently the notation P to denote either a probability distribution (a normed measure, such as $P = \mathcal{N}(0, 1)$) and the corresponding probability density function (pdf) (such as $P(x) = e^{-x^2/2}/\sqrt{2\pi}$).

1 Introduction

1.1 General context

We consider functions having inputs in the form of sets of vectors (or points) and that are invariant under permutation. Furthermore, the functions studied are assumed to be time-consuming and are seen as black boxes, which means that no information regarding their differentiability is known. An example of such function is the simulation of the power production of a wind farm according to the number and the positions of the turbines in a given area: this wind farm is parameterized as the set of vectors $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ where n is the number of turbines and $\mathbf{x}_i$, $i = 1, \ldots, n$ is the vector dimension $d = 2$ of longitude and latitude; $f(X)$ is a realistic simulation of the wind farm production. It is costly because

it involves Computational Fluid Dynamics codes. Other examples of costly functions of clouds of points are systems of water wells, systems of antennas or any other scalable network of interacting devices. A substantial evaluation cost for $f(X)$ prevents analyzing and optimizing the system described by X.

Therefore, the goal of our work is to approximate the function $f(X)$, where X is an unordered set of points, by computationally more efficient functions, $f \simeq \tilde{f} = h(D)$. $\tilde{f}$, the metamodel or surrogate model, is learned from a given input-output set, $D = \{(X_1, f(X_1)), \ldots, (X_N, f(X_N))\}$, N being the number of observations and h a modeling methodology.

In this paper, we focus primarily on Gaussian process regression (1) in order to define $\tilde{f}$. This metamodel is probabilistic, meaning that it defines a full probability distribution for $\tilde{f}$ given D, rather than a single, deterministic value. Gaussian processes are mainly characterized by their covariance functions, called kernels, which must be semi-positive definite (cf. Section 2). The positive definiteness constrains the choice of the kernel.

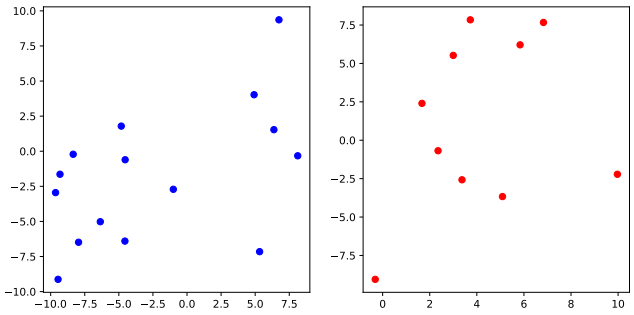

Figure 1: Two clouds of points in $d = 2$ dimensions with $n = 15$ points for the blue cloud and $n = 10$ points for the red one.

This paper addresses the problem of defining semi-definite positive kernels over unordered sets of vectors (i.e., clouds of points). The numbers of vectors in the sets vary. An example of two clouds of points of different sizes is given in Figure 1. It is not obvious to provide an order to the constitutive points of the clouds and their sizes differ, which prevents using Euclidean distance between the two clouds. Although there exist various ways for defining a distance between sets, such as the Hausdorff distance, we will see that not all of them allow to define semi-definite positive kernels.

In the following subsections, we provide an overview of related work, where the definition of kernels over objects such as strings, texts, images, shapes and pyramids is discussed. Like with the clouds of points, many of these objects are formalized as sets of varying sizes.

1.2 Related work

Recent research has covered the topic of kernels over non-vectorial inputs such as character strings and graphs. For instance, (2) provides a general approach to the comparison of such discrete objects through convolution kernels.

Kernels over strings and texts. In (3), kernels over strings relying on Support Vector Machines (SVM) called mismatch kernels are discussed. For the same purpose *Fisher kernels* introduced in (4) can deal with non-vectorial objects such as strings for use with generative models. Although strings objects may have variable length and are not numeric, these methods seem hard to apply on sets of vectors where there is no obvious concatenation structure.
(5) studied relational kernels that can deal with variable-length sequences and weighted automata. Al-

though these contributions do share similar features with the case at hand, mainly handling variable-dimension problems, they come with a stringent framework (concatenation or automata structures) that proved difficult to adapt to sets of points. As with strings, text data can also be learned with kernel methods in classification scenarios with a set of features representation. (6) carried out a matching task on instances across different domains, for instance between documents and tags. Their idea consists in estimating a latent set of features for each instance and embed these representations into a shared RKHS. The authors show the usefulness of a set representation with such data but stop short of comparing the performances of the different kernels and distances between sets.

Kernels over graphs. Many papers are also devoted to constructing semi-definite positive kernels on other discrete objects such as graphs. It is the case in (7), where the authors use diffusion kernels by using the exponential of square matrices defined over graphs. However, this work focuses on discrete sets, which prevents using its methods on clouds of points defined over continuous domains.
Graph kernels can have applications in learning non-vectorial data having an underlying linked structure such as shapes and lines. For instance, it is demonstrated in (8) that learning over point clouds An example of learning over clouds of points carried out with a kernel over graphs can be found in (8). Some of the defined kernels in (8) are proven to be semi-definite positive.
In this article, we do not assume clouds of points to have an underlying graph structure because the local descriptors (individual points) have no obvious and unique linking structure. The authors of (9) also propose kernels for linked, graph-like, objects, but they proceed in a different way. At first, they model a graph as a bag of paths. The authors then define kernels between paths and construct the final kernels between graphs through different averaging techniques of the path kernels. This representation is quite similar to some of the approaches we will soon consider (the MMD and n-MeanMap kernels, see later).

Kernels over images through bags of features representation. Other types of structures, such as images, can be learned with kernel methods. Transforming images to bags of features and considering kernels between these features is the approach proposed in (10). The authors analyze many strategies for averaging the kernel over the features. Though they show some good performances in combining them with support vector machines, no conclusion is drawn regarding the semi-definite positiveness, which is a necessary condition for Gaussian processes. This type of modeling is also mentioned in (11). In this later work, the discussion concerns kernel averages between bags of features with a similarity function over the local descriptors i.e., the elements composing the bag. If the similarity function between the local descriptors is a kernel, called minor kernel, the resulting average function is also a kernel. Any theoretical result about its semi-definite positiveness is proven when the minor kernel is multiplied by a weight function.
Kernels over bags of features for images are also studied in in (12), where the objective is to learn one class at a time on images. The studied kernel computes a similarity between two images by comparing the grey level of pixels with some defined correspondence and has links to the Hausdorff distance. Sets of vectors are not generally associated to levels as in images. Once again, the analogy ends there so that this method cannot be directly applied to our setting.

Kernels over shapes seen as sets of points. In (13), a kernel over sets of points is implemented in the framework of the optimization of a wind farm. The defined kernel is an average kernel over all the pairs of the two sets. Later in our article, we will also consider a such kind of kernel (which we will call n-MeanMap) among other kernels. Several of our test functions will also be related to wind fields, but not all of them.

Similarly, in (14), the automatic detection of road signs using support vector machines based on dissimilarity distances between outlines is discussed. The outlines are modeled as sets of points having specific geometric characteristics, such as triangles or squares. The authors focus on the Hausdorff distance to measure the dissimilarity between two shapes. They show good performances, but show that the distance is not conditionally definite negative, which is necessary in our Gaussian process framework.
A classification application to real human time-series motion data, where the objective is to determine whether a given individual is walking is done in (15). The authors propose a marginalized kernel on bags of vectors with switching linear dynamics. They interpret it as a combination of hidden Markov models and linear dynamics. In this case, the base kernel is a probability product kernel, where the latent distributions are assumed to be Gaussian distributions.

Kernels over pure sets of points with no intermediary representation. Resorting to distances between objects in order to define kernels is not a novel idea. It is the main topic in (16), where distances between sets are identified in order to construct two type of kernels : set kernels in the proximity space based on a representation set as in (17) and set distance substitution kernels. However, many of the resulting kernels are not positive definite.
A well-known paper discussing kernels between two sets of vectors $X = \{\mathbf{x}_1, ..\mathbf{x}_m\}$ and $X' = \{\mathbf{x'}_1, ..., \mathbf{x'}_n\}$ is (18). Here, the core idea is to consider the two sets as being drawn from two distributions P and P' From there, the authors define $k(X, X')$ as $k(P, P')$ based on Bhattacharyya's affinity (19). A benefit of this method is that some of these kernels can be expressed in closed forms. In our paper, we compare them to other types of kernels.
In the same vein, (20) and (21) suggest embedding distributions into a Reproducing Kernel Hilbert Space (RKHS) with a strictly positive kernel, which then allows them to ensure that the induced metric is a distance. The authors demonstrate the positive definiteness of the resulting kernels and show that they can be used in Bayesian optimization. In the following we will, among other things, empirically discuss the influence of the kernel that performs the embedding.

Kernels over sets through histogram pyramid representation. Other techniques, which are not considered in this work, define kernels over non ordered objects, such as the pyramid match kernel developed in (22). It is built on the idea of associating to each set of vectors a histogram pyramid. The similarity between two sets is measured by estimating the overlapping area of the pyramids with different weights.

This paper is structured as follows. First, in Section 2, we discuss the modeling of clouds of points as distributions or vectors and present the associated relevant methods for defining kernels. Then, in Section 3, we introduce a benchmark of test functions and explain our experimental protocol. In particular, extrapolation properties of the kernels are investigated by considering geometrical transformations of possible training data sets. We present the results of the experiments and discuss the performances of the kernels in Section 4. Section 5 summarizes our main conclusions and provides perspectives to this work.

2 Semi-definite positive kernels over clouds of points

We recall here the definition of a kernel and discuss how it is built in practice.

2.1 Kernel functions and Gaussian processes: definitions and conditions

Given a set $\mathcal{X}$, a function $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ is called a kernel if it is symmetric and semi-definite positive, i.e., for any collection $X_1, ...X_N \in \mathcal{X}$ and $c_1, ..., c_N \in \mathbb{R}$ we have,

$$\sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j K(X_i, X_j) \geq 0 \,. \tag{1}$$

Further details about positive definite kernels can be found in (23). The positive semi-definiteness is necessary for kernels to be used in the definition of a Gaussian process. Indeed, if we assume that the function of interest f is a Gaussian process (GP) observed at $X_1, \ldots, X_N$, the variance of any linear combination of the GP values $\sum_{i=1}^{N} c_i f(X_i)$ must be positive and this variance is precisely the left hand side of Equation (1).

The two conditions stated above can be summarized in the following statement.

Theorem 2.1 ((24)) K is a positive semi-definite kernel if and only if there exists a Hilbert space $\mathcal{H}$, and a function $\phi : \mathcal{X} \longmapsto \mathcal{H}$ such that $\forall X, X' \in \mathcal{X}, K(X, X') = \langle \phi(X), \phi(X') \rangle_{\mathcal{H}}$.

The Theorem states that the kernels we are searching for must be scalar products in a Hilbertian space, where the objects of interest are conveyed with a mapping ϕ. However, this mapping and the scalar product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ are not always explicitly known and the image space of $\mathcal{X}$ by $\phi()$ can belong to an infinite dimensional space. This characterization supports the undermentioned distinction between *kernels with explicit mappings* and *kernels with implicit mappings*.

A particular family of interest for kernels with implicit mappings is made up of kernels in the form of $K(X, X') = \sigma^2 \exp(-\frac{\Psi(X,X')}{2\theta^2})$. This requires Ψ to be Hermitian (symmetric in the real case) and *conditionally negative semi-definite* (25) that is, for any N distinct sets of points, the following inequality must hold:

$$\forall c_i \in \mathbb{R},\ \forall i = 1, \ldots, N, \ \text{ with } \sum_{i=1}^{n} c_i = 0 \Rightarrow \sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j \Psi(X_i, X_j) \leq 0 \tag{2}$$

If $\Psi(X, X') = d_\psi(\psi(X), \psi(X'))^2$, where d_ψ is the the Euclidean distance between $\psi(X)$ and $\psi(X')$, the respective images of X and X' into a metric space through a function ψ, the above conditions are equivalent to having a Hilbertian metric, i.e., d_ψ is isomorphic to an L^2 norm (26).

2.2 Modeling clouds of points

2.2.1 Modeling clouds of points as probability distributions

Many of the papers cited in the bibliography model sets of vectors by assuming underlying discrete or continuous probability distributions. It should be noted that this representation changes the input space, as the original space is composed of sets of vectors, which is different from the space of distributions. However, in this work, the objective is not to retrieve a set of points from a distribution but simply to define a kernel between such sets. For the discrete case, the most standard procedure is to consider a set $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ as being associated to a probability distribution $\frac{1}{n}\sum_{i=1}^{n} \delta_{\mathbf{x}_i}$. In this case, there is no ambiguity between the set and the underlying distribution: the corresponding set is just the support of the distribution.

It is is more difficult to find the original set when it is replaced with a continuous distribution. For instance, we can consider a continuous probabilistic representation, typically relying on a parametric distribution. We will soon investigate the possibility of associating a cloud of point $X = (\mathbf{x}_1, ..\mathbf{x}_n)$ to its empirical Gaussian distribution, $\mathcal{N}(m(X), \Sigma(X))$, with mean $m(X) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ and covariance matrix $\Sigma(X) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - m(X))(\mathbf{x}_i - m(X))^\top$. We will denote in the following the probability distribution associated to a cloud X as P_X, regardless of whether it is continuous or discrete.

2.2.2 Modeling clouds of points as vectors of features

It is also possible to represent a set of vectors as a vector of features made of geometrical properties and other characteristics of the cloud. An example of a feature is the size of the set. This representation allows us to circumvent the problems related to the variable size of the sets and the notion of order between their points. Details about this construction will be discussed with the Relevant Feature Map kernel in Subsection 2.3.2.

2.2.3 Defining a kernel with an intermediary mapping

Once one of the two previously described representations is adopted, we are faced with the problem of defining kernels over probability distributions or fixed size vectors. This transposition allows for a broader choice of available kernels. With a kernel K_P defined over a space $\mathcal{P}$ of probability distributions (or K_V defined over a space of vectors E), a mapping $w : \mathcal{X} \longrightarrow \mathcal{P}$ (or E for vectors), we can construct K over $\mathcal{X}$ based on the following result:

Proposition 2.1 (see (27)) Let $w : \mathcal{X} \longrightarrow \mathcal{Y}$ and let $K_{\mathcal{Y}}$ be a kernel function on $\mathcal{Y}$. Then $K_{\mathcal{Y}} \circ w$ is a kernel function on $\mathcal{X}$.

Note that $K_{\mathcal{Y}} \circ w(X, X') = K_{\mathcal{Y}}(w(X), w(X'))$. When defining kernels in this way, we may be confronted with the lack of richness (mainly dimension) of the image space of w. It is generally the case when w is not one-to-one, i.e., it is not injective.

2.3 Kernels with implicit mappings

In the remainder all the considered kernels are homoscedastic: $\forall X \in \mathcal{X}, K(X, X)$ is a constant. We consider kernels with implicit mappings in the form of $K(X, X') = \sigma^2 \exp(-\frac{\Psi(X,X')}{\theta^2})$ where $\Psi(X, X') = d_\psi(\psi(X), \psi(X'))^2$, ψ being a mapping from $\mathcal{X}$ into a probability Space, $\mathcal{P}$, or an Euclidean space E to be defined and $d_\psi(,)$ a distance defined with ψ. We refer to this kind of kernels as *substitution kernels with distance*.

2.3.1 Substitution kernels based on a distance between probabilities

Two important distances between probabilities are the Wasserstein distance and Maximum Mean Discrepancy (MMD) which we now focus on. Under some conditions, these two metrics can be used to define a Gaussian process covariance.

Definition 2.1 (Wasserstein Distance) *Let $(\mathcal{E}, dist)$ be a metric space. For $p \geq 1$, we denote by $P_p(\mathcal{E})$ the set of probability measures on $\mathcal{E}$ endowed with the p-Wasserstein distance, defined as (see (28) for more details)*

$$\mathcal{W}_p^p(P, P') = \inf_{\gamma \in \Pi(P,P')} \int dist(x, x')^p d\gamma(x, x') \tag{3}$$

with $\Pi(x, x')$ all distributions over $\mathcal{E} \times \mathcal{E}$ with marginals P and P'.

We suppose here that $\mathcal{E}$ is $\mathbb{R}^d$.

The Gaussian case. Assuming that $P = \mathcal{N}(m, \Sigma)$, $P' = \mathcal{N}(m', \Sigma')$ and $dist$ is the Euclidean distance, we get

$$\mathcal{W}_2^2(P, P') = ||m - m'||^2 + tr(\Sigma + \Sigma' - 2(\Sigma^{1/2}\Sigma'\Sigma^{1/2})^{1/2}) \, .$$

We use an approximation of $\mathcal{W}_2^2$, as in (29),

$$\mathcal{W}_2^2(P, P') \approx ||m - m'||^2 + ||\Sigma^{1/2} - \Sigma'^{1/2}||^2_{Frobenius} \, .$$

The last approximation becomes an equality when $\Sigma^{1/2}\Sigma'^{1/2} = \Sigma'^{1/2}\Sigma^{1/2}$. Therefore, the approximation being the sum of two Hilbertian metrics (deriving from scalar product), the following is a valid kernel:

$$K(X, X') = \sigma^2 \exp\left(-\frac{||m(X) - m(X')||^2}{\theta_1^2} - \frac{||\Sigma(X)^{1/2} - \Sigma(X')^{1/2}||^2_{Frobenius}}{\theta_2^2}\right) \tag{4}$$

We call this kernel Gauss-Wass.

The discrete uniform case. It is not guaranteed that $\mathcal{W}_2^2$ of Equation (3) always verifies the conditions of Equation (2) when $P = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$, $P' = \frac{1}{m}\sum_{=1}^m \delta_{x'_i}$ and $x_i, x'_j \in \mathbb{R}^d$ with $d \geq 2$, as it is explained in (30). But we know ((31)) that if P and P' are two non-negative unidimensional measures in $\mathbb{R}$ with $P(\mathbb{R}) = P'(\mathbb{R}) = 1$,

$$\mathcal{W}_2^2(P, P') = ||\mathcal{C}_P^{-1} - \mathcal{C}_{P'}^{-1}||^2_{L^p([0,1])}$$

with $\mathcal{C}_P(x) = \int_{-\infty}^x dP$, $\mathcal{C}_{P'}(x) = \int_{-\infty}^x dP'$ their respective cumulative distribution functions and $\forall r \in [0,1], \mathcal{C}_\mu^{-1}(r) = \min_x\{x \in \mathbb{R} \cup \{-\infty\} : \mathcal{C}_\mu(x) \geq r\}$, the r percent quantile. In this one-dimensional case, $\mathcal{W}_2^2(P, P')$ is symmetric and conditionally negative definite (see (32)). The sliced Wasserstein distance, described below, generalizes the well-funded unidimensional Wasserstein distance to many dimensions by averaging projections onto lines.

Definition 2.2 (Sliced Wasserstein Distance on empirical uniform) *Consider the empirical probabilities $P_X = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$, $P_{X'} = \frac{1}{m}\sum_{j=1}^m \delta_{x'_j}$ associated respectively to $\{x_1, ..., x_n\}$ and $\{x'_1, ..., x'_m\}$ with $x_i, x'_j \in \mathbb{R}^d$ and $\mathcal{S} = \{\alpha \in \mathbb{R}^d, ||\alpha|| = 1\}$. The projected empirical measures on the line directed by $\alpha \in \mathcal{S}$, denoted as $\alpha * P_X$ and $\alpha * P'_X$ are respectively given by $\frac{1}{n}\sum_{i=1}^n \delta_{<x_i,\alpha>}$ and $\frac{1}{m}\sum_{i=1}^m \delta_{<x'_i,\alpha>}$. The sliced Wasserstein distance is defined as,*

$$SW_2^2(P_X, P_{X'}) = \int_{\mathcal{S}} \mathcal{W}_2^2 \alpha * P_X, \alpha * P_{X'}) \mathrm{d}\alpha \, . \tag{5}$$

Resorting to this version of the Wasserstein distance, we get,

$$K(X, X') = \sigma^2 \exp\left(-\frac{SW_2^2(P_X, P_{X'})}{\theta^2}\right) . \tag{6}$$

The latter is symmetric and semi-definite positive, as stated in (33). It is denoted as Slice-Wass.

Instead of defining the distances directly between the distributions, it is also possible to map them into another space equipped with an Hilbertian metric. For instance, one can embed the distributions into a RKHS with an embedding kernel. If the embedding kernel is "characteristic", the mapping from the distributions to the functions is injective. The distance between distributions can be measured through distances between their image functions, giving rise to the notion of Maximum Mean Discrepancy (see (21)).

Definition 2.3 (Maximum Mean Discrepancy) *Given a Reproducing Kernel Hilbert Space $\mathcal{H}$ with a characteristic kernel (such as $k_{\mathcal{H}}(x, .) = \exp(-\frac{||x-.||^2}{2\theta^2})$), the mean embedding of a distribution P is defined as $\mu(.) = \int P(x) k_{\mathcal{H}}(x, .)\mathrm{d}x$. This allows to define the Maximum Mean Discrepancy between two distributions P and P' with respect to $k_{\mathcal{H}}$ as MMD$^2(P, P') = \|\mu - \mu'\|^2_{\mathcal{H}}$.*

We denote the embedding of P_X as μ_X. It is easily computable for empirical distributions. With $P_X = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$, we get,

$$\int P_X(x) k_{\mathcal{H}}(x, .)\mathrm{d}x = \frac{1}{n}\sum_{i=1}^n k_{\mathcal{H}}(., x_i) \, ,$$

and

$$\langle \mu_X, \mu_{X'} \rangle = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k_{\mathcal{H}}(x_i, x'_j) \ . \tag{7}$$

We call $\langle \mu_X, \mu_X \rangle$ the energy of the distribution P_X, as in (35). If one develops the square norm between two mean embeddings,

$$\|\mu_X - \mu_{X'}\|^2_{\mathcal{H}} = \langle \mu_X, \mu_X \rangle + \langle \mu_{X'}, \mu_{X'} \rangle - 2\langle \mu_X, \mu_{X'} \rangle \ ,$$

we obtain,

$$\|\mu_X - \mu_{X'}\|^2_{\mathcal{H}} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k_{\mathcal{H}}(x_i, x_j) + \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} k_{\mathcal{H}}(x'_i, x'_j) - 2\frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k_{\mathcal{H}}(x_i, x'_j) \ .$$

Substituting the MMD distance into the squared exponential kernel, we have

$$K(X, X') = \sigma^2 \exp(-\frac{\|\mu_X - \mu_{X'}\|^2_{\mathcal{H}}}{\theta^2}) \tag{8}$$

which is symmetric and semi-definite positive. We denote this kernel MMD.

2.3.2 Substitution kernels based on Euclidean distances

Instead of mapping the clouds into a space of probabilities, one can also define a connection with a vector of features.

Definition 2.4 (Relevant Features Map Kernel) *Let us consider two sets of vectors $X = \{x_1, ..., x_n\}$ and $X' = \{x'_1, ..., x'_m\}$. Suppose that we have a multivalued mapping function $\psi : \mathcal{X} \longrightarrow \mathbb{R}^o$, with o a finite integer. The image of any $X \in \mathcal{X}$ is in the form of of a vector $(\psi_1(X), ..., \psi_o(X)) \in \mathbb{R}^o$. We define $K(X, X') = \sigma^2 \exp\big(-\sum_{j=1}^{o} \frac{|\psi_j(X) - \psi_j(X')|^2}{\theta_j^2}\big)$. K is symmetric and semi-definite positive.*

In this article, we associate the following features to a given cloud of points:

- the coordinates of the mean
- the eigenvalues and eigenvectors of the empirical covariance matrix
- the cardinality of the set
- the largest and the shortest distances between points of the set.

Because the eigenvectors are defined up to their direction (minus eigenvector is also an eigenvector with the same eigenvalue) and because we are in $\mathbb{R}^2$, we choose the eigenvectors in the first and fourth quadrant of the plan. With this choice, there is no ambiguity on their definition. We denote the above kernel RFK.

2.4 Kernels with explicit mappings

Recall with Theorem 2.1 that all kernels functions (symmetric and semi-definite positive) can be written in the form $k(X, X') = \langle \phi(X), \phi(X') \rangle_{\mathcal{H}}$. It is the case of the following kernels,

$$K(X, X') = \langle \frac{\mu_X}{||\mu_X||}, \frac{\mu_{X'}}{||\mu_{X'}||} \rangle_{\mathcal{H}} \tag{9}$$

where, recall, μ_X is the mean embedding of a distribution associated to X. We call this kernel n-MeanMap. The normalization allows to have a kernel verifying the homoscedasticity property, $K(X, X)$

is independent of X. For μ_X and $\mu_{X'}$ the embeddings of the discrete uniform distributions associated to X and X', the detailed expression of the kernel is

$$K(X, X') = \frac{\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k_{\mathcal{H}}(x_i, x'_j)}{\sqrt{\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} k_{\mathcal{H}}(x_i, x_j)}\sqrt{\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} k_{\mathcal{H}}(x'_i, x'_j)}} . \tag{10}$$

It is worth noting that an embedding of the representing distribution is already applied before applying the scalar product. Yet, the scalar product can also be directly defined in the distribution space. It is the case of the Probability Product Kernels which we now introduce.

Definition 2.5 (Probability Product Kernels, see (36)) *Suppose that we have two sets of vectors X, X', their respective probabilistic space representations (Ω, P_X), $(\Omega, P_{X'})$ and $\rho \in [0, 1]$. A Probability Product Kernel is expressed as $K(X, X') = \int_\Omega P_X^\rho(x) P_{X'}^\rho(x)\mathrm{d}x$. Identifying P_X^ρ and $P_{X'}^\rho$ as elements of L^2, we get $K(X, X') =< P_X^\rho, P_{X'}^\rho >_{L^2}$, which makes it symmetric and semi-definite positive since L^2 is a Hilbert Space.*

Applying this definitions to the Gaussian distributions $P_X = \mathcal{N}(m(X), \Sigma(X))$ and $P_{X'} = \mathcal{N}(m(X'), \Sigma(X'))$ gives a closed formula:

$$K(X, X') = (2\pi)^{\frac{(1-2\rho)d}{2}} |\Sigma(X)^+|^{\frac{1}{2}} |\Sigma(X)|^{-\frac{\rho}{2}} |\Sigma(X')|^{-\frac{\rho}{2}} C(X, X') \tag{11}$$

with

$$C(X, X') = \exp\big(-\frac{\rho}{2} m(X)^\top \Sigma(X)^{-1} m(X) - \frac{\rho}{2} m(X')^\top \Sigma(X')^{-1} m(X') + \frac{1}{2}(m^+)^\top (\Sigma^+)^\top m^+\big)$$

where $\Sigma^+ = (\rho\Sigma(X)^{-1} + \rho\Sigma(X')^{-1})^{-1}$ and $m^+ = \rho\Sigma(X)^{-1}m(X) + \rho\Sigma(X')^{-1}m(X')$. If $\rho = \frac{1}{2}$, it is called the Bhattachayra kernel and it is the Expected Likelihood Kernel when $\rho = 1$ (see (18)). In this paper, we will use the estimated Gaussians discussed in 2.2.1 to test the Bhattachayra kernel, which we call *Bhatta*.

3 Benchmark functions and experimental protocol

3.1 Analytical test functions

We now present several functions taking sets of vectors as inputs. These functions will serve to study the aforementioned kernels, when used within Gaussian processes, with the goal to better understand the relationship between the kernel and the approximated function. These functions are simplifications of some of the many situations naturally parameterized as sets of vectors. Wind farms and well fields are two examples, inspiring the *wind farm proxy* below. Design of experiments problems come down, in some cases, to optimizing the *Mindist* function, which is another example of such functions. Finally, the *inertia function* will also be considered. This function can be encountered, e.g., as the accumulated energy of multi-components systems, or as the regularization term in ridge regression problems.

3.1.1 Wind farm proxy

A first family of test functions is the energy production of a wind farm. Each wind turbine yields a certain amount of energy and is subject to the wake effect caused by neighboring turbines. The inputs are in the form of $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ with $\mathbf{x}_i = (x_{i,1}, x_{i,2}) \in \mathbb{R}^2$ ($d = 2$). Each $\mathbf{x}_i$ represents the Cartesian coordinates of a turbine. The considered test function is given below.

$$F(\{\mathbf{x}_1, ..., \mathbf{x}_n\}) = \sum_{i=1}^{n}\Big(\prod_{j, j\neq i} f_p(\mathbf{x}_j, \mathbf{x}_i)\Big) f_0(\mathbf{x}_i) \tag{12}$$

and it represents the total yield considering all turbines. Details about the function f_p (representing the gain factor of $\mathbf{x}_i$ in the presence of $\mathbf{x}_j$) and f_0 (a constant representing the maximal yield of a given turbine may be found in appendix. f_0 is chosen to be a constant. In the remainder of the paper, we will use the following notations:

- F_θ stands for functions that account for wind coming from one single direction $\theta \in (0, 360)$.
- F_{nd} for functions modeling the average of n directions uniformly distributed in $(0, 360)$.

We consider the followings test functions : F_0, F_{90}, F_{45}, F_{40d}.
We provide graphical representations of $\mathbf{x}_i \longrightarrow f_p(\mathbf{x}_j, \mathbf{x}_i)$ with $\mathbf{x}_j = (0, 0)$ for the 4 scenarios in Figure 2. These graphics show the amplitude of the wake interaction between two turbines, depending on their relative position and the direction of the wind. For instance, in case $\theta = 0$, we can notice that the scale of the interactions are more important in the direction of the wind, and they only occur *behind* the fixed turbine.

As F_θ are obtained by summing f_p over all pairs of points, we give two examples (F_0 and F_{45}) of the functions' layouts in Figure 3, where 9 points are fixed and one is allowed to vary. The values next to the points show their individual contribution to the total production. There is no unit of measure.We can see that the yield of an additional point depends on the direction of the wind and the existing points (already placed).

3.1.2 Mindist Function

The optimization of a design of experiments (DoE) is another important application that involves an unordered set of vectors as input. A classical requirement for DoEs is that they correctly fill the space, which can be expressed as maximizing the smallest distance between two points of the set. In compliance with previous cloud notations, a design is written $X = (\mathbf{x}_1, ..., \mathbf{x}_n)$ with $\mathbf{x}_i \in \mathbb{R}^d$ for some $d \in \mathbb{N}$ (later we will focus on $d = 2$). Our second test function, $F_{minDist}$ for design function, is

$$F_{minDist}(X) = \min_{i \neq j} ||\mathbf{x}_i - \mathbf{x}_j||. \tag{13}$$

This function will also be referred to as the *Mindist* function. *De facto* this function only depends on the values of two elements of the input set, the two closest points, as opposed to the other functions of our benchmark that factor in all of the set components.

3.1.3 Inertia Function

Lastly, we include the function modeling the inertia of a group of points in $\mathbb{R}^d$ (in our case $d = 2$). For $X = \{\mathbf{x}_1, ...\mathbf{x}_n\}$, the Inertia function is given by :

$$F_{inert}(X) = \sum_{i=1}^{n} ||\mathbf{x}_i - \bar{X}||^2 \tag{14}$$

where $\bar{X}$ is the empirical average of the points coordinates, $\bar{X} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$.

3.2 Experiments

Kernel acronyms. The tested kernels are the Maximum Mean Discrepancy (MMD) of Equation (8), the Relevant Feature Kernel (RFK) of definition (2.4), the Sliced-Wasserstein kernel (Slice-Wass) of Equation (6), the Gaussian Wasserstein (Gauss-Wass) of Equation (4), the normalized MeanMap kernel (n-MeanMap) of Equation (9) and the Bhattacharyya kernel (Bhatta) of Equation (11).

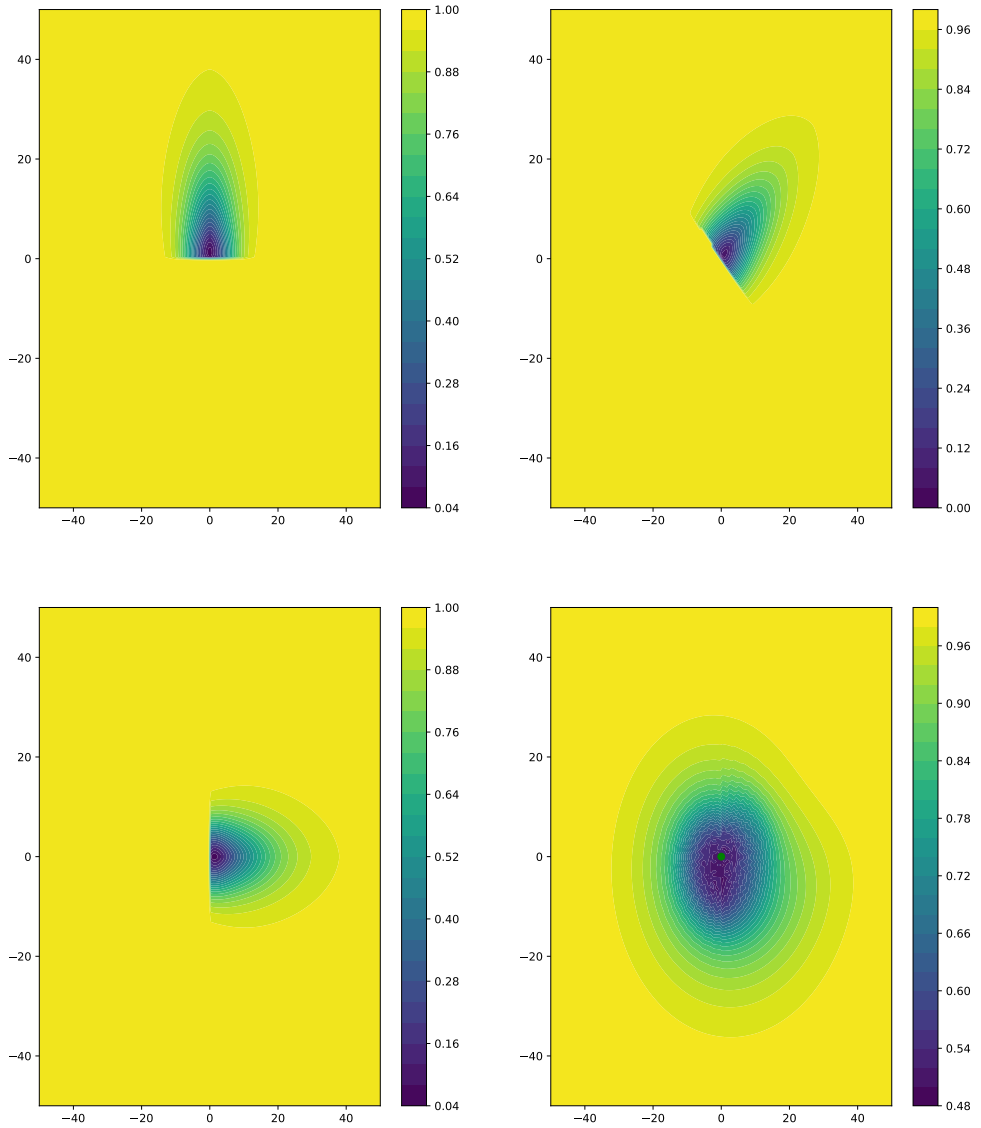

Figure 2: Representation of f_p with $\theta = 90°$ at top left, $\theta = 45°$ top right, $\theta = 0°$ bottom left, and averaged directions at bottom right.

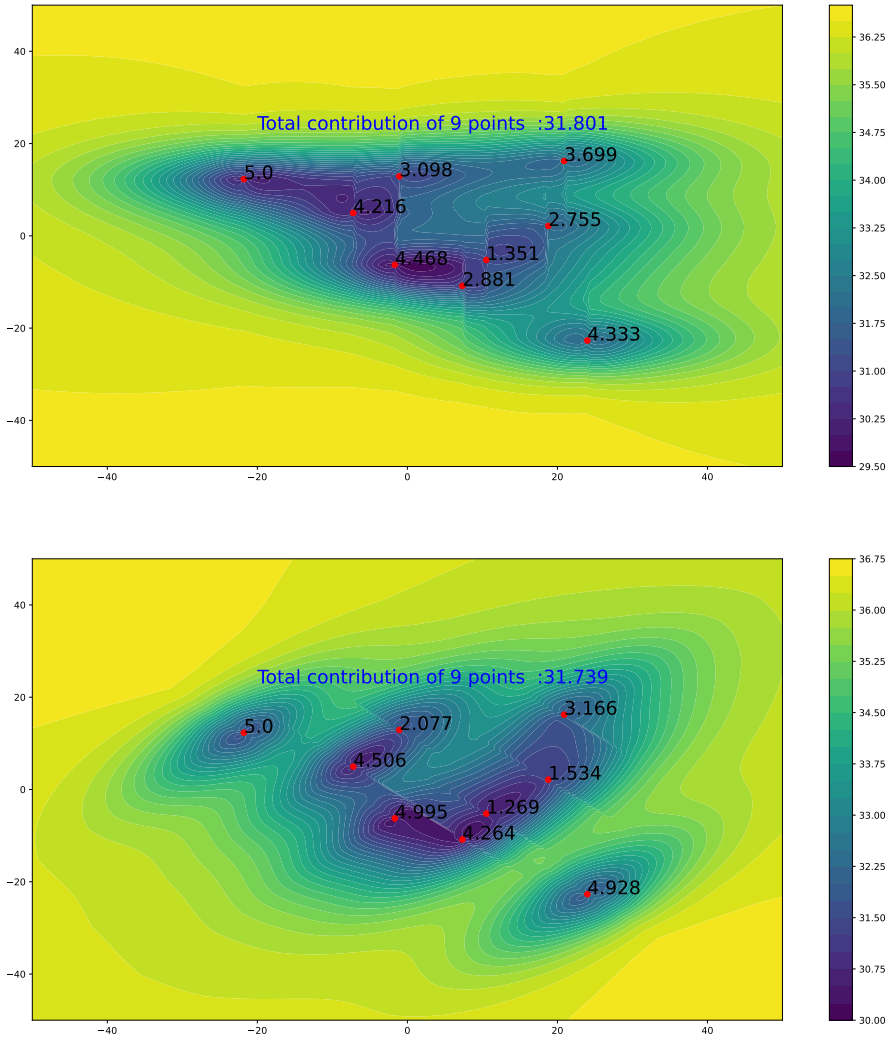

Figure 3: Representation of F_0 and F_{45} with 9 fixed points and a one varying. The maximal contribution of a point is fixed to 5.

3.2.1 Experimental Protocol

The default design of experiments. The objective of the experiments is to study the effect of the kernel on the predictive ability of conditional Gaussian processes. Our test functions are defined over clouds of points of variable size, n, $n \in \{n_{\min} = 10, .., n_{\max} = 20\}$. For all functions except $F_{minDist}$, we choose the initial size of the training set as 300. This size is the average over the 11 sets sizes of $10 \times d \times n$, which yields $\frac{10d}{11} \times \sum_{n=n_{\min}}^{n_{\max}} = n = 300$. For $F_{minDist}$, $n \in \{n_{\min} = 3, ..., n_{\max} = 8\}$ and we choose a design of size 200.

The *random cloud design of experiments* is implemented as follows:

1. The size of each cloud is randomly picked in $n \in \{n_{\min}, \ldots, n_{\max}\}$.
2. Each point is uniformly sampled in the domain of the function.

Clouds of points generated with such procedure are called random in the following. This generative procedure guarantees a uniform distribution of the size n and the points positions (once n is chosen), but it does not guarantee a uniform design in the space of clouds of points with varying sizes. For example, clouds generated through the geometrical transformations which we will soon introduce have a very small probability of occuring. We use the random design to conduct the three types of experiments detailed below with the following objectives: *i)* Study the suitability of the considered kernels to the considered family of test-functions. *ii)* Study the extrapolation performances of the kernels on geometrically modified clouds of points. This helps to assess the robustness of the joint choice of the initial random design and the kernel, with regards to the function at hand.

Study of the embedding kernel, $k_{\mathcal{H}}$. The first experiment allows to choose the embedding kernel for MMD and n-MeanMap. We investigate the influence of the different local kernels on random test clouds of points. The considered embedding kernels are Matérn 5/2, Matérn 3/2, Laplacian and Gaussian kernel (or RBF for radial basis function). We choose a hyperparameter for each dimension in order to consider an anisotropic setting. The tests are carried out on the F_{40d} function. The size of the testing data is 1000 clouds.

Prediction on random clouds of points. In the second experiment, we study the prediction ability of 6 kernels on a random design of cloud of points. The size of the testing data is 1000 clouds.

Prediction on geometrically transformed clouds of points. We want to investigate how learning on random clouds allows to predict the function output for specific clouds of points, unlikely to occur as random clouds. A simple and instructive construction for such alternative designs is by geometrically transforming (ditating and rotating) a given random cloud. In addition to testing the extrapolation properties of the kernels, this third experiment, by confronting anisotropic test functions (the wind farm proxy with one wind direction) with geometrical transformations, is instrumental in better understanding the relationship between the kernel, its hyperparameters, and the test function.

We thus study the prediction of the best learned kernels on horizontally, vertically, isotropically dilated clouds of points and on rotated clouds of points. For a given cloud of points $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, we note, X_r^θ, X_d^δ, X_{dh}^δ, X_{dv}^δ its rotated, isotropically dilated, horizontally dilated and vertically dilated transformations. We have

$$
\begin{aligned}
X_r^\theta &= \{R_\theta \mathbf{x}_1 + (I - R_\theta)\bar{X}, ..., R_\theta \mathbf{x}_n + (I - R_\theta)\bar{X}\}\ , \\
X_d^\delta &= \{D_\delta \mathbf{x}_1 + (I - D_\delta)\bar{X}, ..., D_\delta \mathbf{x}_n + (I - D_\delta)\bar{X}\}\ , \\
X_{dh}^\delta &= \{D_{\delta h} \mathbf{x}_1 + (I - D_{\delta h})\bar{X}, ..., D_{\delta h} \mathbf{x}_n + (I - D_{\delta h})\bar{X}\}\ , \\
X_{dv}^\delta &= \{D_{\delta v} \mathbf{x}_1 + (I - D_{\delta v})\bar{X}, ..., D_{\delta v} \mathbf{x}_n + (I - D_{\delta v})\bar{X}\}\ .
\end{aligned}
$$

Rotations and dilations are done with respect to the point cloud means, $\bar{X}$. In addition,

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, D_\delta = \begin{bmatrix} \delta & 0 \\ 0 & \delta \end{bmatrix}, D_{\delta h} = \begin{bmatrix} \delta & 0 \\ 0 & 1 \end{bmatrix}, D_{\delta v} = \begin{bmatrix} 1 & 0 \\ 0 & \delta \end{bmatrix},$$

where θ and δ are the rotation and dilation factors. The evolution of a cloud of points following 5 dilations is sketched in Figure 4. The initial cloud is gradually dilated into the constraint square. The range of the factor of dilation, $\delta \in [1, 10]$ (20 uniformly spaced values of δ), ensures that the obtained clouds always remain in the square bounding the whole domain. Notice in Figure 4, upper-right plot, that the most isotropically dilated clouds of points ($\delta = 10$) are the most similar to the random clouds of the design of experiments.The domain of the wind farm proxy is the square bounded by $(-50, -50)$ and $(50, 50)$. Dilations are initialized (i.e., $\delta = 1$) as random clouds within the square bounded by $(-5, -5)$ and $(5, 5)$. The domains of the Mindist and Inertia functions are smaller, $(-10, -10)$ à $(10, 10)$ and so is their initialization square when $\delta = 1$ which is spanned by $(-2.5, -2.5)$ and $(2.5, 2.5)$. We will look at the prediction performance for 20 transformations of 50 initial clouds, that is for a total of 1000 new testing clouds.

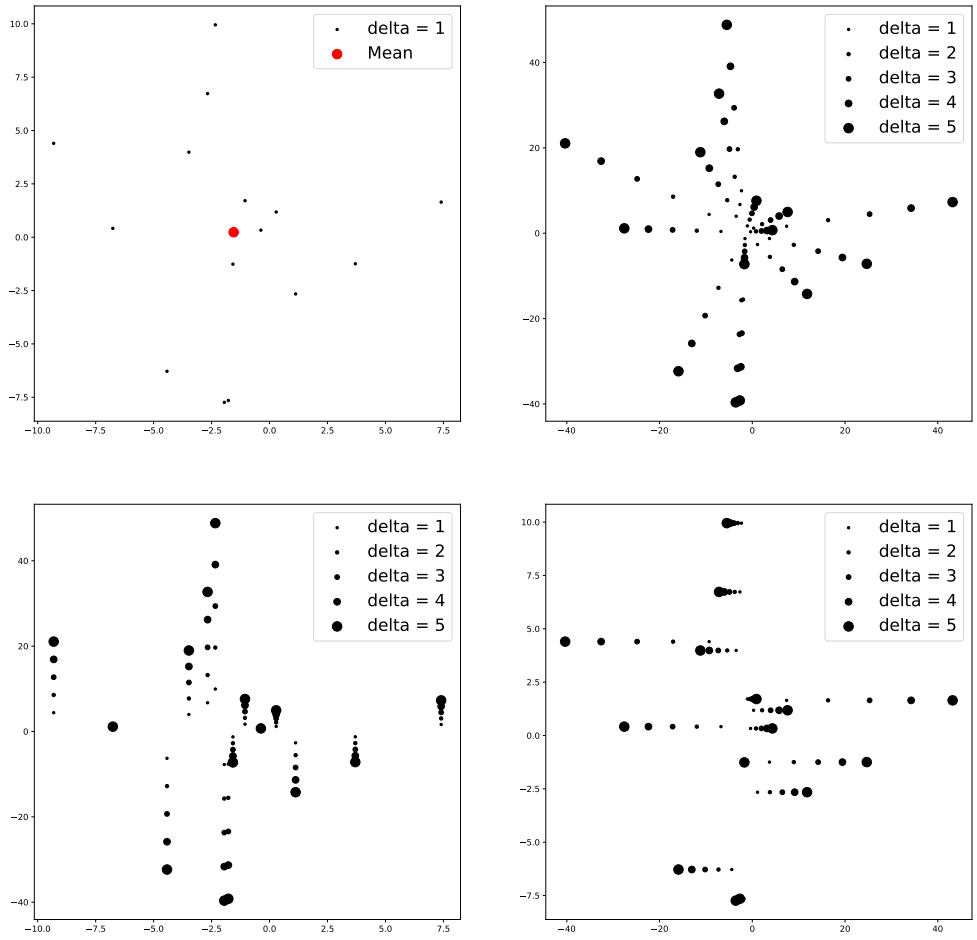

Figure 4: Illustration of the dilation transformation of clouds: initial cloud at top left with its mean (red bullet), the 5 isotropic dilations at top right, 5 vertical dilations at bottom left and 5 horizontal dilations at bottom right. Note that the horizontal and vertical ranges vary between the plots.

3.3 Estimation of hyperparameters.

All the kernels that are investigated here are in the factorized form $K = \sigma^2 R$, with R a correlation matrix. Adding the nugget effect as a hyperparameter, the kernels become $K = \sigma^2(R + \upsilon I) = \sigma^2 R'$. We denote the vector of hyperparameters as Θ. Their value is typically determined by maximizing the log-likelihood criterion with respect to the data :

$$\log p(Y|D, \Theta) = -\frac{1}{2} Y^\top K^{-1} Y - \frac{1}{2} \log |K| - \frac{n}{2} \log(2\pi) \, . \tag{15}$$

In the case where $K = \sigma^2 R'$, we have

$$\sigma^2(\Theta) = \frac{1}{n} Y^\top R'^{-1} Y \ ,$$
$$\log p(Y|D,\Theta) = -\frac{1}{2}(n \log(\sigma^2(\Theta)) + \log|R'| + n + n\log(2\pi))$$

All hyperparameters are estimated by maximizing the log-likelihood with a BFGS algorithm, except for Bhatta. For the latter, we choose a list of nugget effects and pick the one yielding the best likelihood value since it is the only hyperparameter. Concerning the initialization of BFGS for each of the above listed kernels and each function, 5 different initial sets of hyperparameters (except for RFK, where we choose 9 initial points) are chosen and we keep the one yielding the largest log-likelihood at the end of the optimization. Finally, given that the focus of this paper revolves around the GP covariance model, the GP trend is considered to be constant and equal to 0, and is therefore not optimized.

4 Results and discussions

In this Section, we report the prediction performances of conditional GPs built from the investigated kernels. The performances are quantified with the $Q2$ criterion. Suppose we have a function F to approximate and M real observed input-outputs, $\{(X_1, y_1), ..., (X_M, y_M)\}$, not seen by the model during training. Let $\widehat{Y} = \{\widehat{y}_1, ..., \widehat{y}_M\}$ be the predicted values of a model, then we have,

$$Q2 = 1 - \frac{\sum_{i=1}^{M}(\widehat{y}_i - y_i)^2}{\sum_{i=1}^{M}(y_i - \bar{y})^2} \ , \tag{16}$$

with $\bar{y} = \frac{1}{M}\sum_{i=1}^{M} y_i$. Note that $-\infty \leq Q2 \leq 1$ and that the closer the $Q2$ value is to 1, the more accurate we can estimate the model to be. Additionally, if the variance of $\widehat{Y}$ is very small, meaning the function is flat in the testing zone, then it is difficult to obtain a positive $Q2$.

The performances of conditional GPs are also computed under the form of Mean Average Error (MAE), which may sometimes give a different insight than the $Q2$ criterion. For the sake of clarity, the results in terms of MAE are only provided in Appendix A.

4.1 Predictions on the wind farm proxy

Study of the embedding kernel on F_{40d}. The effect of the embedding kernel is investigated with the multi-directional wind farm proxy, F_{40d}. Only two kernels depend explicitly on an embedding kernel, MMD and n-MeanMap. The results, provided in Table 1, show that for the MMD based kernel, the embedding kernel choice has a limited influence, as the obtained $Q2$ values are all around 0.9. The n-MeanMap kernel seems slightly more sensitive to the embedding kernel, RBF and Matérn 5/2 embedding kernels yielding better predictive performances. In the following, we choose Matérn 5/2 for MMD and RBF for n-MeanMap. It is important to note that this is a choice made for the sake of simplicity, and that there is no guarantee that the optimal embedding kernels for the other test functions considered in this work correspond to the ones that are selected here.

K \ $k_{\mathcal{H}}$	RBF	Laplacian	Matérn 3/2	Matérn 5/2
MMD	0.901	0.895	0.907	0.906
n-MeanMap	0.734	0.625	0.681	0.700

Table 1: $Q2$ with different embedding kernels on F_{40d}, the test clouds are a random design.

Results on random clouds of points. Table 2 gathers the results obtained with all kernels on all the wind farm proxy functions. Remember that the test group is generated with the random procedure of Subsection 3.2.1.
The first observation concerns the probability-based kernels. On these functions, kernels based on a discrete uniform distribution (Slice-Wass, MMD, n-MeanMap) outperform Gaussian-based kernels (Bhatta, Gauss-Wass). Among the kernels based on uniform discrete distributions, we can notice that MMD and Slice-Wass yield better results than n-MeanMap. Besides kernels relying on the discrete uniform distribution, the RFK (relevant features kernel) has a better average Q2 than Slice-Wass. The overall best competitor for this set of experiments is the MMD kernel.
The second observation is a slight improvement of Gaussian based kernels and n-MeanMap on F_{40d}. This is related in part to the loss of individual point information with Gaussian measures modeling: with a single wind orientation, the relative position of each pair of points and its relation to the wind direction is the physical driving phenomenon; with an averaged effect over 40 wind directions, the relative positions between turbines are less of a driving factor and the cloud as a whole starts having more effect on the output. Concerning the wind farm proxy, only the three best performing kernels, namely, MMD, RFK and Slice-Wass, are further investigated in the context geometrical transformations of random clouds.

Function \ Kernels	MMD	n-MeanMap	BHATTA	RFK	Slice-Wass	Gauss-Wass
F_0	0.906	0.647	0.146	0.897	0.828	0.177
F_{45}	0.868	0.623	0.160	0.893	0.821	0.187
F_{90}	0.899	0.639	0.145	0.871	0.843	0.172
F_{40d}	0.906	0.734	0.261	0.799	0.824	0.308

Table 2: $Q2$ of 6 kernels on all the wind farm proxy functions, the testing clouds come from a random design.

Results on horizontally dilated clouds of points. Recall from Figure 4 that horizontally dilated clouds of points and the design of experiments are different. Horizontally dilated clouds of points are contained in rectangles with a small height and a varying width (growing with δ), whereas the clouds of the random design are distributed in the largest encompassing square. This test can thus be considered as an extrapolation test and degraded results are expected with respect to tests on a random design of clouds. Table 3 shows the global $Q2$ for 1000 clouds of points containing 50 initial clouds of points each dilated 20 times. It is seen that the MMD kernel yields better $Q2$'s than RFK on F_0, F_{45}, and F_{40d}, but presents the worst score on F_{90}. We can also note the poor performance for RFK and Slice-Wass on F_0. Overall, MMD tends to provide the best predictive performance on this specific case, whereas RFK tends to behave considerably worse than the other kernels.
More detailed results can be found in Figure 5 which provides the real wind farm proxies outputs (in green) and the predictions of the different GPs as a function of the horizontal dilation. Observe in Figure 5 that the MMD predictor decreases more rapidly than the other kernels predictors (RFK for instance) when δ becomes smaller. This is explained by the fact that the norm of the difference in distribution embeddings (used to define the MMD) is sensitive to all clouds dilations. On the contrary, the RFK includes some features that are invariant under the geometrical transformations, such as the size of a set (the number of points), thus the GP with RFK predictions decrease more slowly. More explanations can be found in the discussion Section 4.3.

Function \ Kernels	MMD	RFK	Slice-Wass
F_0	0.05	-15.535	-10.033
F_{45}	0.519	-0.879	0.397
F_{90}	0.518	0.711	0.631
F_{40d}	0.103	-2.415	-0.827

Table 3: $Q2$ observed on horizontally dilated clouds of points

Results on vertically dilated clouds of points. We observe in Table 4 the $Q2$ values obtained for the vertical dilations of point clouds. Because the vertical dilation is perpendicular to the horizontal one, the relative ranks of the kernels is inverted for the F_0 and F_{90} functions, which are the same functions after a $90°$ rotation. On the contrary of F_0 and F_{90}, the F_{45} and F_{40d} functions are symmetric with respect to a vertical or horizontal dilation. The relative rank of the kernels for these functions is the same for horizontal and vertical dilations. The MMD yields better $Q2$ values on F_{45} and F_{90}, and has the worst Q2 on F_0. We also observe a poor $Q2$ of RFK and Slice-Wass on F_{90} while both are the best options on F_0. A detailed comparison of the GP predictions for varying δ is provided in Figure 6.

Function \ Kernels	MMD	RFK	Slice-Wass
F_0	0.422	0.698	0.732
F_{45}	0.625	-0.732	0.146
F_{90}	-0.079	-11.204	-12.407
F_{40d}	0.247	-1.894	0.503

Table 4: $Q2$ observed over vertically dilated clouds of points.

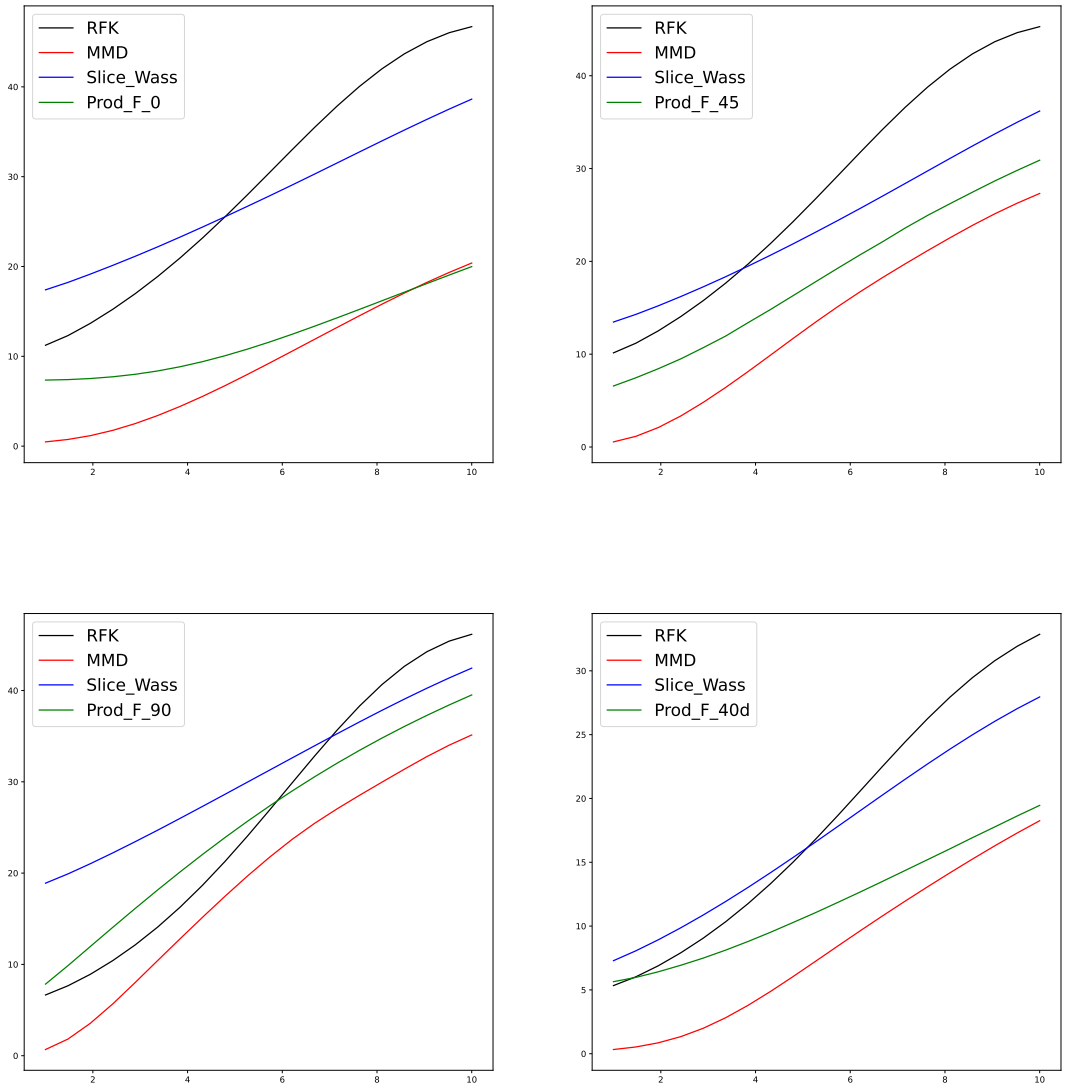

Figure 5: Wind farm proxy outputs as a function of the horizontal dilation δ: function output in green, Gaussian process with RFK, MMD and Slice-Wass kernels in black, red, and blue. Wind orientations are $0°$, $45°$, $90°$ and the 40 directions (i.e., F_0, F_{45}, F_{90}, F_{40d}) from left to right and top to bottom. The curves are averaged over 50 clouds.

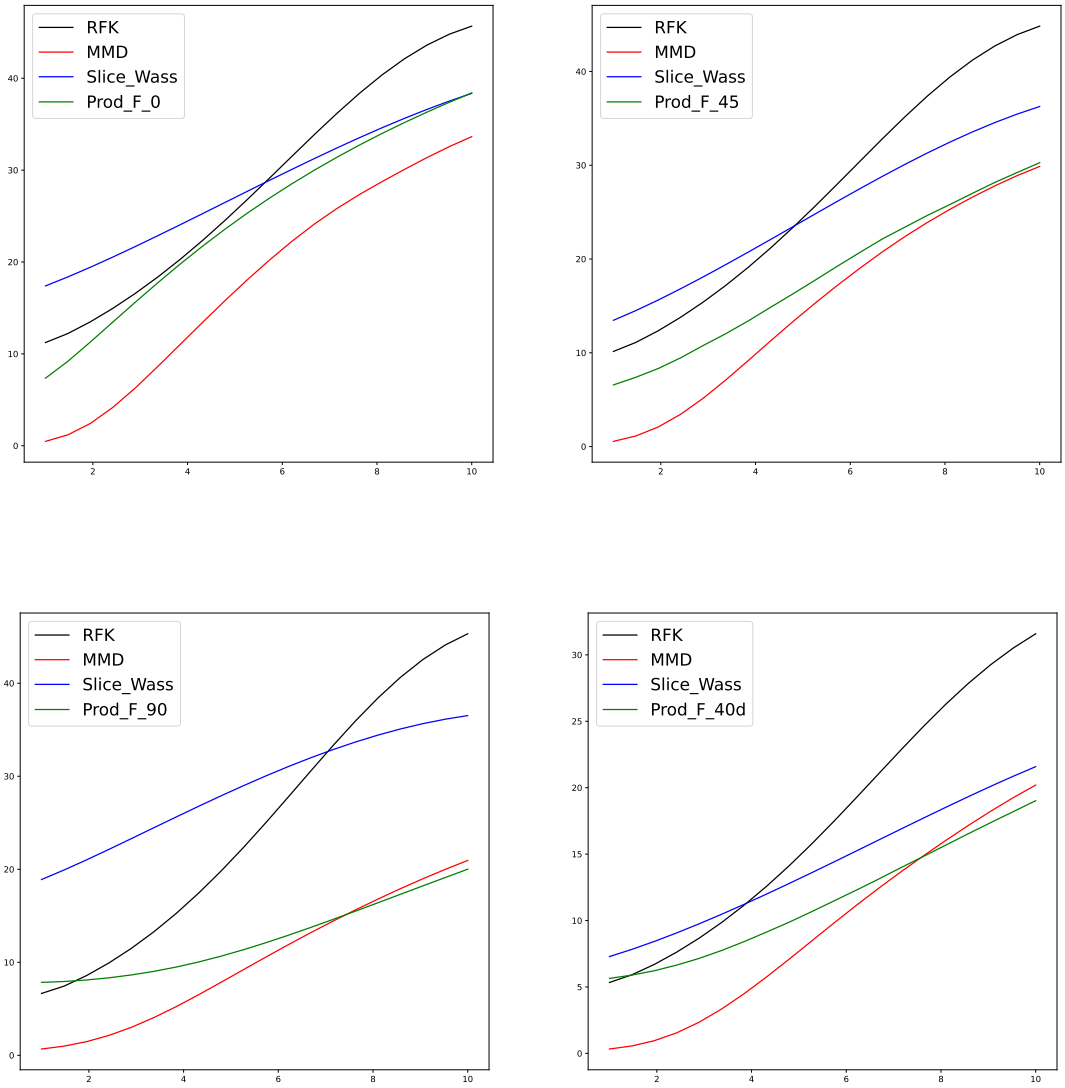

Figure 6: Wind farm proxy outputs as a function of the vertical dilation δ: function output in green, Gaussian process with RFK, MMD and Slice-Wass kernels in black, red, and blue. Wind orientations are $0°$, $45°$, $90°$ and the 40 directions (i.e., F_0, F_{45}, F_{90}, F_{40d}) from left to right and top to bottom. The curves are averaged over 50 clouds.

Results on isotropic dilated clouds of points. When the clouds are isotropically expanded, tests of the kernels on the wind farm functions yield the $Q2$ values of Table 5. All of the tested kernels yield $Q2$ values around 0.9. As can be seen from the complementary Figure 7, all GP predictions, irrespectively of the kernel, capture the trend of the true function. The predictions are particularly accurate for large δ's. Indeed, the training data set which is used to condition the Gaussian processes is similar to the clouds generated with an isotropic maximal dilation where $\delta = 10$. The $Q2$ values obtained here (Table 5) are slightly better than the ones obtained with the random design (Table 2). This can be explained by the $Q2$ metric: the range of the test values is larger with these dilations than it was with the random design, which creates a large variance at the denominator of the last term in Equation 16. This contributes to a higher $Q2$.

Function \ Kernels	MMD	RFK	Slice-Wass
F_0	0.933	0.952	0.893
F_{45}	0.939	0.954	0.933
F_{90}	0.942	0.931	0.879
F_{40d}	0.940	0.974	0.975

Table 5: $Q2$ observed on isotropically dilated clouds of points

Results on rotated clouds of points. When the test clouds are created by rotating random clouds, the prediction qualities of the GPs are those of Table 6. We note that RFK has the best $Q2$ on F_0, F_{45}, F_{90}. Slice-Wass has the highest $Q2$ on F_{40d}. MMD being less sensitive to the rotation than RFK, it is beaten by RFK, although its predictions are quite good. The accompanying prediction curves for varying θ are given in Figure 8. The $Q2$ values obtained at the rotated clouds of points are smaller than those at dilated (isotropic) clouds. While the dilation were simple motions away (for diminishing δ) from the observed clouds with a monotonous decrease in output (return to the null GP mean, concentration of the turbines) which matched the true function, the rotation is a more complex motion through the data base and the true function has a non-trivial periodicity. In addition, the $Q2$ metric is penalized in the rotation case by the small range of magnitude of the true function which can be seen in Figure 8.

Function \ Kernels	MMD	RFK	Slice-Wass
F_0	0.808	0.863	0.780
F_{45}	0.780	0.877	0.802
F_{90}	0.800	0.881	0.797
F_{40d}	0.701	0.771	0.775

Table 6: $Q2$ observed on rotated clouds of points

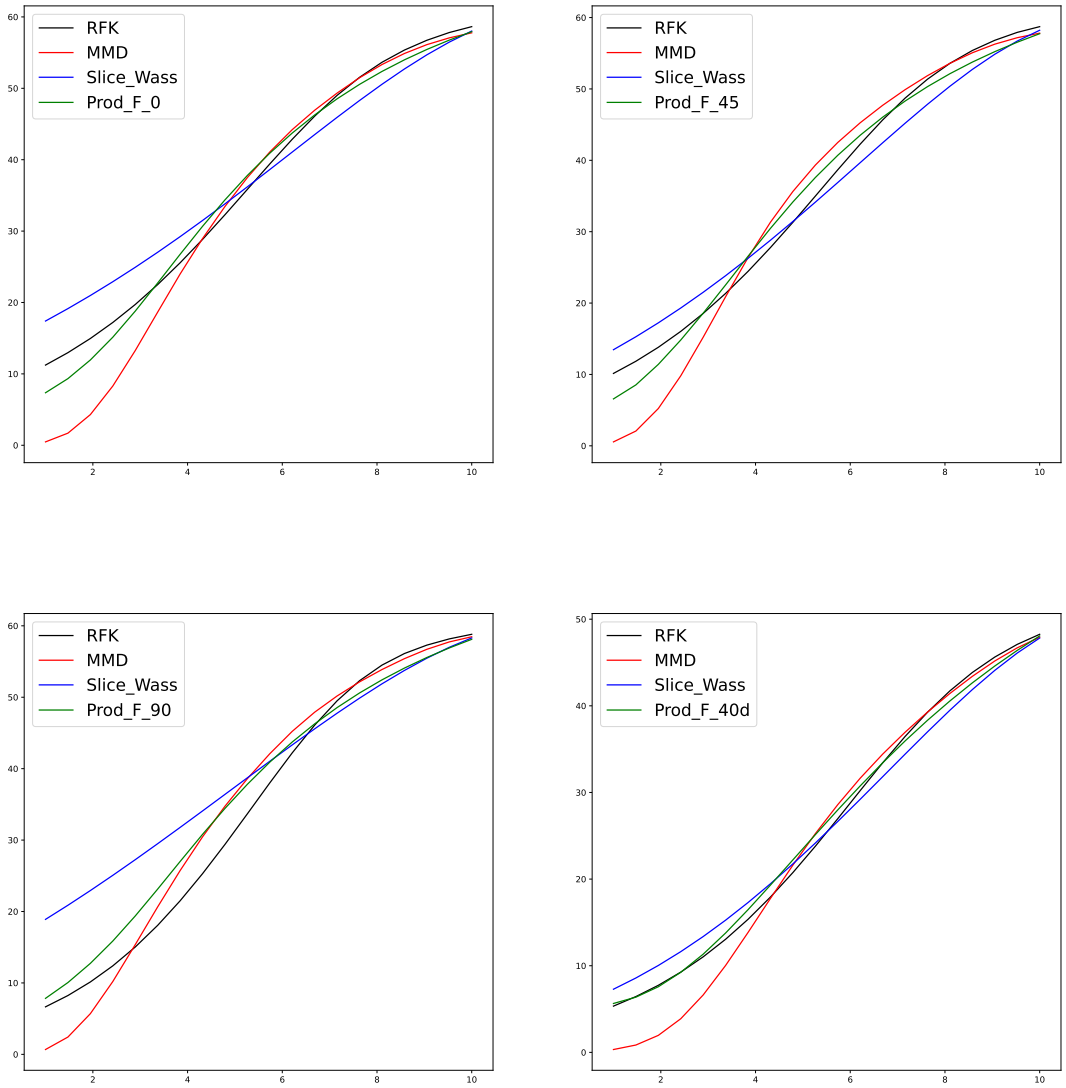

Figure 7: Wind farm proxy outputs as a function of the isotropic dilation δ: function output in green, Gaussian process with RFK, MMD and Slice-Wass kernels in black, red, and blue. Wind orientations are $0°$, $45°$, $90°$ and the 40 directions (i.e., F_0, F_{45}, F_{90}, F_{40d}) from left to right and top to bottom. The curves are averaged over 50 clouds.

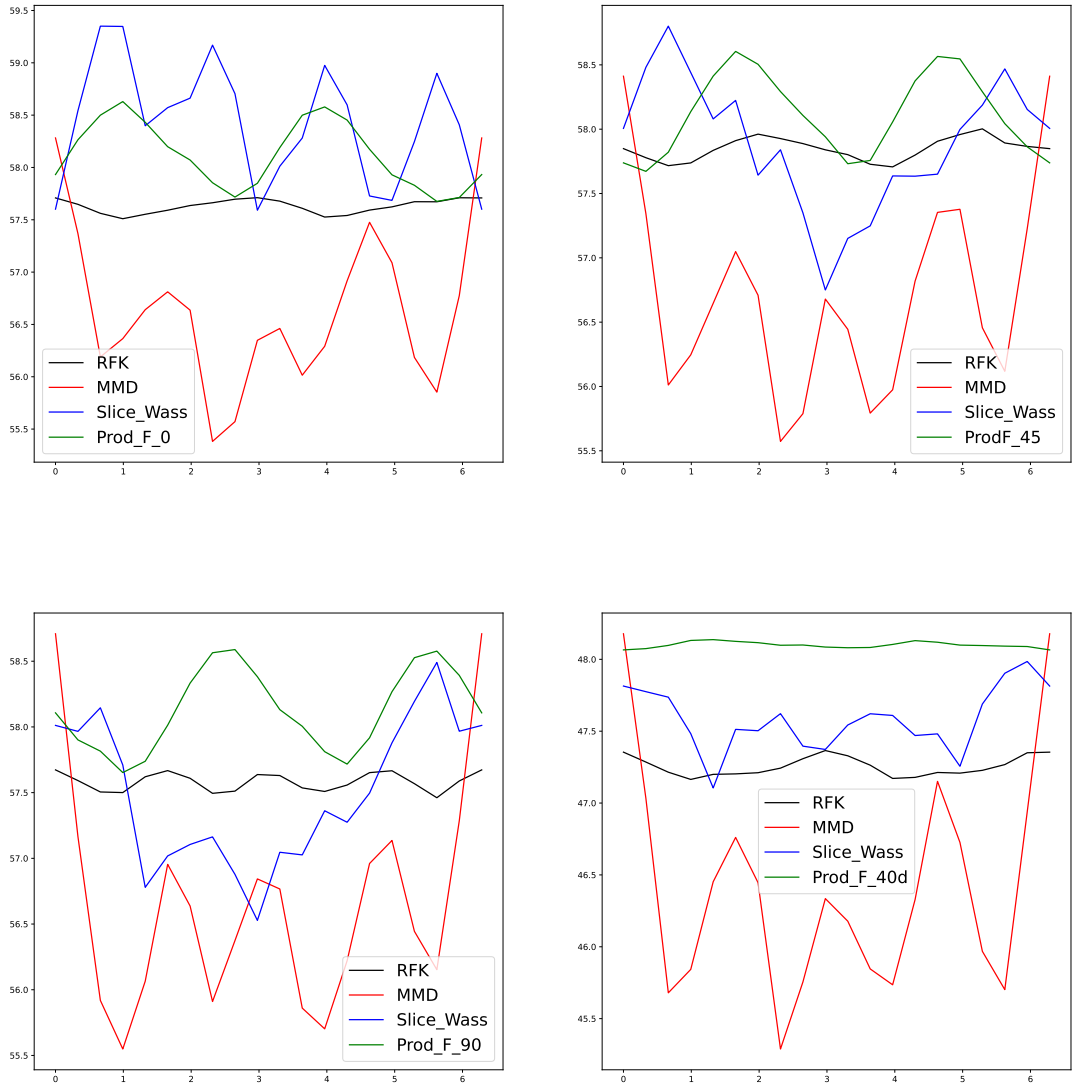

Figure 8: Wind farm proxy outputs as a function of the rotation θ: function output in green, Gaussian process with RFK, MMD and Slice-Wass kernels in black, red, and blue. Wind orientations are $0°$, $45°$, $90°$ and the 40 directions (i.e., F_0, F_{45}, F_{90}, F_{40d}) from left to right and top to bottom. The curves are averaged over 50 clouds. Notice the small range in outputs (vertical axis).

4.2 Predictions on the Inertia and Mindist functions

Results on random clouds of points. We can observe that on the Mindist function ($F_{minDist}$), all 6 kernels perform poorly, except for RFK and Slice-Wass (see Table 7). On F_{inert}, while the embedding-based kernels (MMD, n-MeanMap) yield better results in terms of prediction accuracy than the Gaussian-based kernels (Gauss-Wass and Bhatta), their performance is slightly decreased when compared to the wind farm proxy, while the Gaussian-based kernel performance is improved (compare Tables 7 and 2).

Function \ Kernels	MMD	n-MeanMap	BHATTA	RFK	Slice-Wass	Gauss-Wass
F_{inert}	0.734	0.506	0.463	0.988	0.905	0.502
$F_{minDist}$	-0.051	0.035	-0.124	0.997	0.587	-0.064

Table 7: Summary of the $Q2$ observed on $F_{minDist}$ and F_{inert}

Results on the dilated clouds of points. RFK also best extrapolates on isotropically dilated clouds as seen from the $Q2$ values in Table 8) for F_{inert} and $F_{minDist}$. The accompanying prediction curves for varying δ are in Figure 9.

Function \ Kernels	MMD	RFK	Slice-Wass
F_{inert}	0.901	0.982	0.845
$F_{minDist}$	-0.802	0.998	0.280

Table 8: $Q2$ observed on isotropically dilated clouds of points for the F_{inert} and $F_{minDist}$ functions.

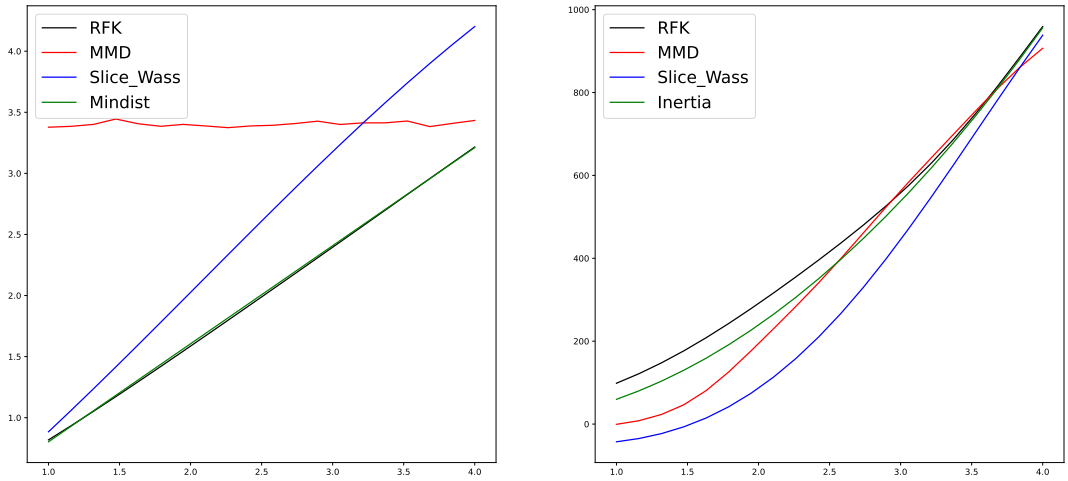

Figure 9: GP predictions and true output as a function of the isotropic dilation factor δ. The GP predictions based on the RFK, MMD and Sliced Wasserstein kernels are in black, red and blue, while the true function is in green. The functions are Mindist (left) and Inertia (right). The results are averaged over 50 clouds.

Results for rotated clouds of points. As can be seen in Table 9, the GP with RFK remains the best model when predicting on rotated clouds for both the Inertia and the Mindist functions. The performance of RFK on $F_{minDist}$ is not surprising since one of its features is precisely the function to learn. The information for the Inertia function is also present in the features of the RFK since the eigenvalues of the covariance matrix of the vectors set sum up to the Inertia function. This explains the high $Q2$ values of RFK on F_{inert}. The sliced Wasserstein kernel is quite good at handling rotations for F_{inert} and, to a lesser extent, for $F_{minDist}$. MMD based kernels are not able to capture the Mindist information. Furthermore, as is visible on the right plot of Figure 10, they are quite sensitive to rotations of the clouds, falling back towards the GP mean more rapidly than other kernels as θ departs from $(0 \bmod 2\pi)$.

Function \ Kernels	MMD	RFK	Slice-Wass
F_{inert}	0.422	0.988	0.854
$F_{minDist}$	-0.025	0.998	0.206

Table 9: $Q2$ observed on rotated clouds of points for the F_{inert} and $F_{minDist}$ functions.

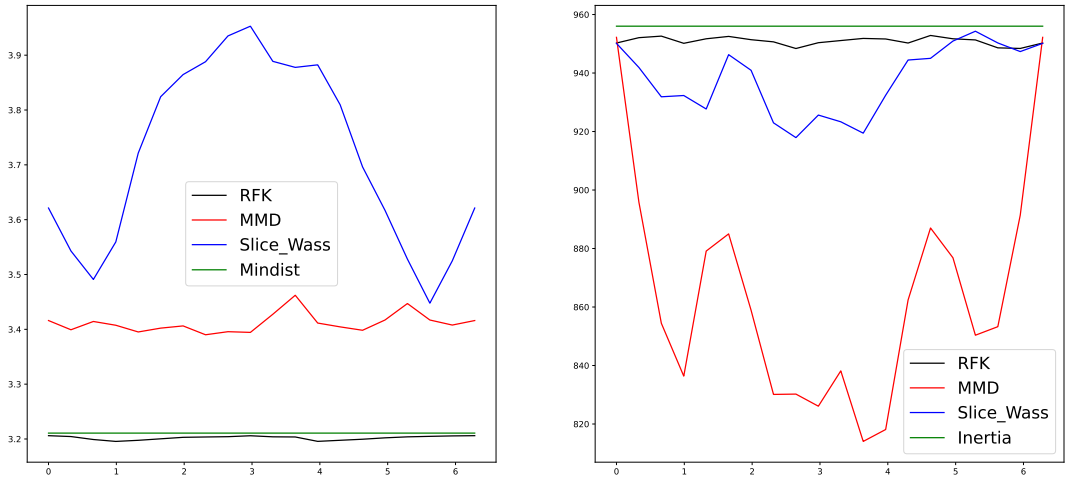

Figure 10: GP predictions and true output as a function of the rotation angle θ. The GP predictions based on the RFK, MMD and Sliced Wasserstein kernels are in black, red and blue, while the true function is in green. The functions are Mindist (left) and Inertia (right). The results are averaged over 50 clouds.

4.3 Discussion

We revisit our results with explanations coming either from the intrinsic nature of the kernels or from the design of experiments.

4.3.1 Embedding based kernels

Regarding embedding based kernels, we focus on the optimal length scale values resulting from the likelihood optimization. These characteristics are more related to the training data set and the test function than to the intrinsic nature of the kernel.

MMD. The MMD squared exponential embedding kernels are controlled by two length scales, θ_1 and θ_2, scaling the dimensions between two points $\mathbf{x}$ and $\mathbf{x}'$ through $|x_1 - x_1'|/\theta_1$ and $|x_2 - x_2'|/\theta_2$. In Figure 11, the two length scales learned by MMD for various wind directions are represented as vectors. We can notice that they are symmetrical for F_0 and F_{90} with respect to the first bisector. The vector $(\theta_1, \theta_2)^\top$ points more in the direction of the wind (than in the perpendicular one) whereas for F_{45}, it is almost directed as the bisector. When 40 wind directions are considered on the average (F_{40d} function), $(\theta_1, \theta_2)^\top$ is similar to that of F_{45} with larger length scale values.

The length scales should capture sensitivities that come from the physics inherent to the wind farm proxies. For example, when the wind is at $0°$, the wind farm production changes more with vertical displacements of the turbines (which rapidly enter or leave an interaction area) than with horizontal

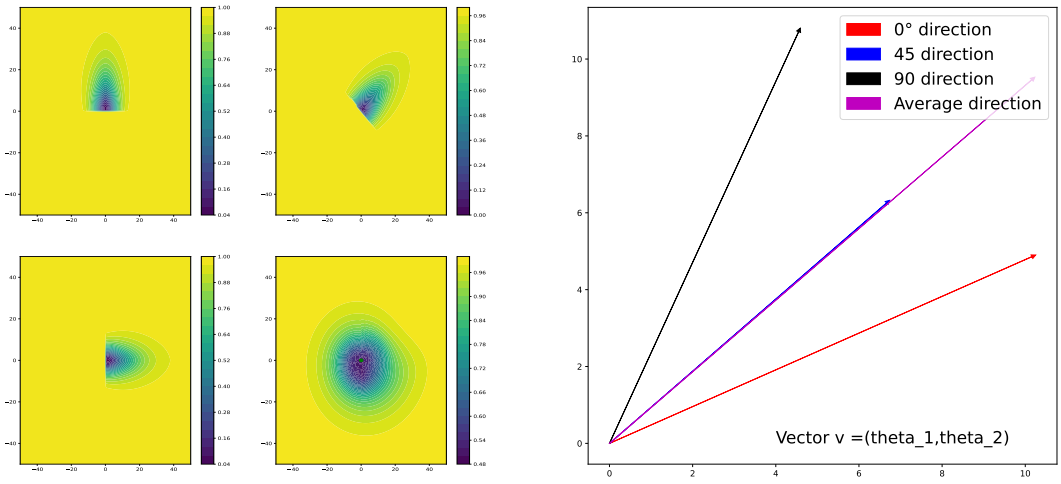

Figure 11: Vectors of length scales of the MMD embedding Matérn 5/2 kernel learned by maximum likelihood on the wind farm proxy for various wind directions. Left: reminder of the turbine contributions for winds at $90°$,$45°$,$0°$ and 40 directions (left to right, top to bottom). Right: $(\theta_1, \theta_2)^\top$ vectors of length scales of the embedding kernel.

displacements. This sensitivity is retrieved in the MMD-based model first through $\theta_1 > \theta_2$, and then through changes in the energy of the embedding, where the energy of the embedding is $\langle \mu_X, \mu_X \rangle$ (cf. Equation (7)). This is illustrated in Figure 12, where F_0 varies more quickly with vertical dilation than with horizontal ones. The same property is observed for the energies of the embeddings. This characteristic of the MMD model helps it to have a better $Q2$ on the horizontal dilation on F_0 than RFK and Slice-Wass (see Table 3). A complementary illustration is the top left plot of Figure 5 where the MMD predictor augments as slowly as the function.

However, the vertical hyperparameter, θ_2, is not optimally learned since the predictor augments very slowly regarding the function. These differences can be explained in part by the following: for F_0, θ_1 corresponds to the x-axis and the direction of the wind is better learned since in this direction the variation is more continuous (there are always interactions). However, θ_2 which corresponds to the the direction perpendicular to the wind, suffers from interactions that cease more quickly and abruptly and is not accurately tuned.

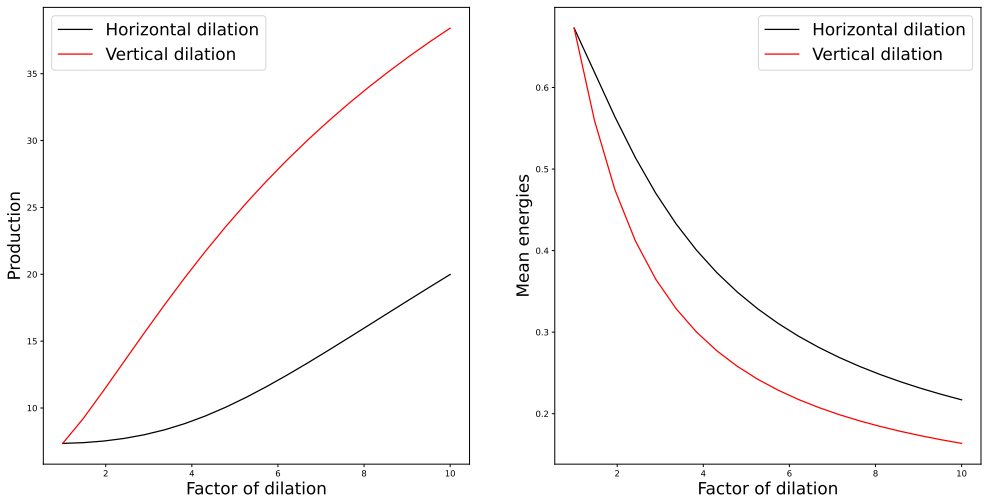

Figure 12: Variations of F_0 (left) and averaged squared norm of the embedding (right) with respect to horizontal and vertical expansions of the clouds. The length scales are learned on F_0.

The function F_{90} varies more quickly under horizontal dilation than under vertical one. This is also present in the learned energies. We observe the opposite phenomenon than for F_0.
F_{45} varies similarly under horizontal and vertical dilation. As shown in Figure 13, we can also see this on the embedding energies estimated in the MMD. The two hyperparameters are correctly learned to have almost equal values and the kernel has a better $Q2$ on the two scenarios than RFK and Slice-Wass. The predictions on F_{45} under vertical and horizontal dilations can be seen on the top right of Figures 5 and 6.

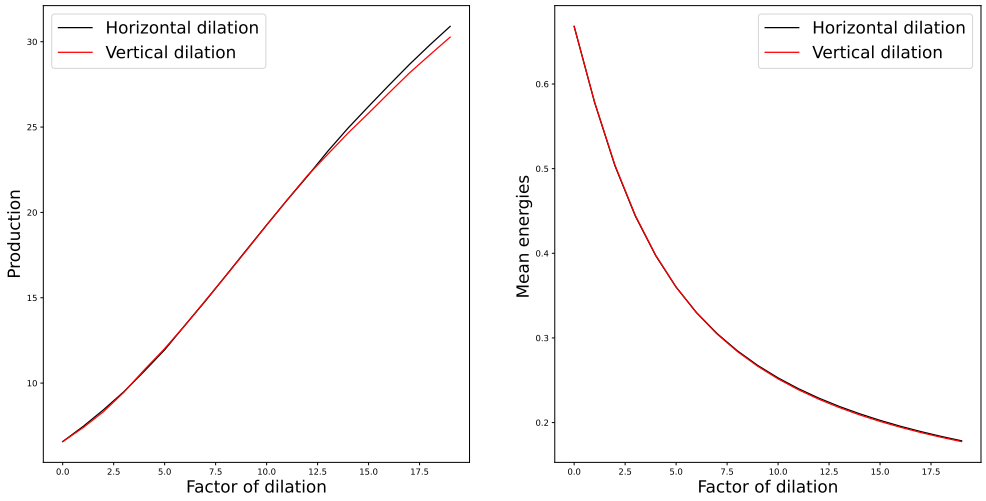

Figure 13: Variations of F_{45} (left) and averaged squared norm of the embedding (right) with respect to horizontal and vertical expansions of the clouds. The length scales are learned on F_{45}.

F_{40d} is characterized by an averaged effect of multiple wind directions. Therefore, there is not a

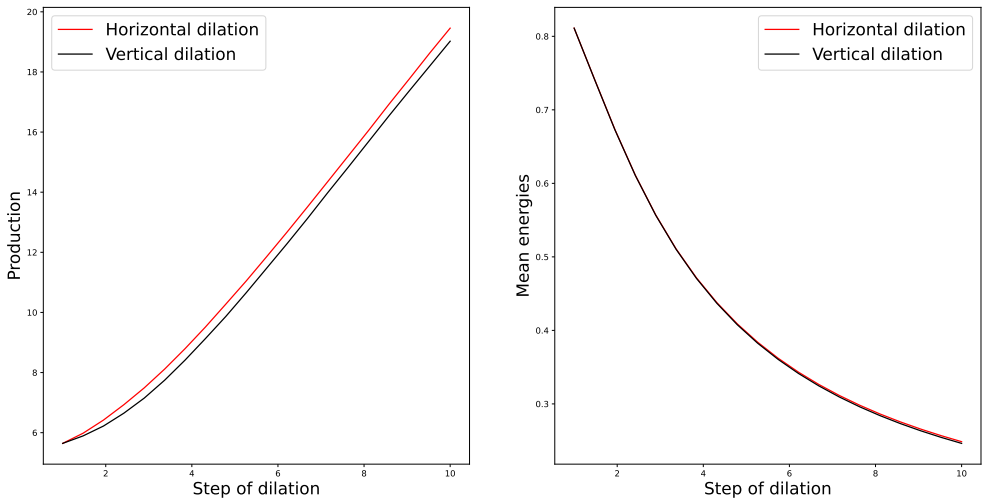

Figure 14: Variations of F_{40d} (left) and averaged squared norm of the embedding (right) with respect to horizontal and vertical expansions of the clouds. The length scales are learned on F_{40d}.

very privileged direction and the hyper-parameters learned are almost the same. This equivalence is also present in Figure 14 when looking at the effects of horizontal and vertical dilations on the MMD model. Similarly, the $Q2$ obtained with MMD on F_{40d} on vertical and horizontal dilations are not so different as read in Tables 3 and 4.
On all the wind farm proxy functions presenting directions of interactions, the MMD has shown different behaviours under horizontal and vertical expansions of the inputs. These two types of sensitivities make it less efficient than RFK under isotropic dilation, see Figure 7 and Table 5. As far as rotation is concerned (see Table 6 and Figure 8), MMD has worse performances than RFK. We can observe that the predictor is very sensitive to rotation unlike the other kernels, as seen in Figure 8. The observed results suggest that MMD based kernels are more relevant to the modeling of functions having privileged directions of variability.

n-MeanMap. As with MMD, the vector of length scales learned with the n-MeanMap kernel points more towards the wind direction than towards its perpendicular (see Figure 15). Again, this is related to the physics underlying the wind farm proxies, where turbines have longer interactions lengths along the wind direction, which induces smaller output variations for turbine displacements in the wind direction, and vice versa for displacements perpendicular to the wind direction. However, contrarily to the MMD, the n-MeanMap kernel is linear in the space of the embeddings since it is a scalar product, cf. Equation (9). This explains in part its less accurate predictions in comparison with the MMD.

4.3.2 Wasserstein distance based kernels

Sliced Wasserstein. The sliced Wasserstein kernels have a good score on most of our benchmark functions. Thanks to the projections of the vectors onto a set of directions, anisotropic functions which behave differently in each direction, can be learned. Thanks to the averaging, isotropic functions are well represented too. Slice-Wass presents good performances on F_0, F_{45} and F_{90} with just 10 projected directions. For F_{40d} we augment the number of directions to 40, so that the number of projections matches the number of considered wind directions. We have also checked that $Q2$ on F_{40d} is invariant under

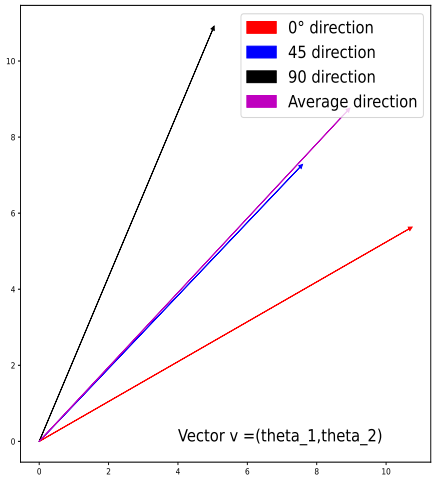

Figure 15: Vectors of length scales, $(\theta_1, \theta_2)^\top$, of the n-MeanMap embedding kernel learned by maximum likelihood on the wind farm proxy for various wind directions.

$\pi/2$ rotation of the directions of projections of the kernel, it remains equal to 0.824. The large number of projection directions of the kernel allows it to be less sensitive with respect to cloud rotations for isotropic functions. This is quantified by the largest average $Q2$ on F_{40d} (see Table 6). However, Sliced Wasserstein constructed with uniform directions does not structurally (besides the training data base) present a privileged direction. This limitation is in part quantified by the negative $Q2$ values on horizontal dilated clouds of F_0 (see Table 3) and vertical dilated clouds of F_{90} (see Table 4). Sliced Wasserstein kernel also presents good performances on $F_{minDist}$ (although, of course, behind RFK), and on F_{inert}. The various results presented in this paper, as well as its theoretical properties, show that the Sliced Wasserstein makes for a robust kernel over sets of points in the context of Gaussian processes.

Gaussian Wasserstein. We observe in Tables 2 and 7 that, under the scenarios of our experiments, Gaussian Wasserstein based kernels do not yield good scores in comparison with the other kernels. This can be partially explained by the fact that the individual characteristics of the points in the cloud are lost when representing a given cloud through a single Gaussian distribution. The Gaussian distribution presents a more isotropic nature than the uniform discrete ones. Also, Gaussian modeling suffers from a lack of representation capacity (flexibility) as seen through its small number of parameters (cf. Equation 4). However, we notice a slight, relative (with respect to itself), improvement on F_{40d} and F_{inert} of its performances. This improvement can be related to the more isotropic nature of the modeled functions.

4.3.3 Feature Based Kernels

The following discussion addresses the chosen features to model a cloud of points as a vector. Different results could be obtained with other features. First of all, note that RFK has good performances on several of the considered functions (see Table 2 and Table 7). RFK cannot deal well with horizontal and vertical dilation in general. It augments quickly under horizontal and vertical dilation (see Figure 5 and Figure 6). For example, F_0 varies very slowly under vertical dilation and F_{90} varies slowly under horizontal dilation whereas RFK always augments very quickly. This can be explained by the training data set in which clouds of points dilated in a specific direction are not included. The kernel is not parameterized in such a way to be sensitive to such geometrical transformations. This justifies the poor $Q2$ observed with F_{90} and F_0 respectively under vertical and horizontal dilation, see Tables 3 and 4.

Given that the kernel has no privileged direction of dilation, it presents good results under isotropic dilation, as is shown in Table 5 and Figure 7.

Under rotation, the kernel is less sensitive. RFK is constructed with several features invariant under rotation, such as the mean, the size, the shortest and greatest distances between points and the shortest and greatest distances to the mean. On Table 6, we can observe that RFK has the largest $Q2$ values on F_0, F_{45} and F_{90}. It is only beaten by Slice-Wass on F_{40d}. This is because the considered functions are not particularly sensitive to point clouds rotations, as shown in Figure 8 .

We should notice that RFK has a $Q2$ value larger than 0.99 on $F_{minDist}$ (the best one compared with the other kernels) because the function to approximate is a one of its features. This is further illustrated in Figure 16, where we remove each feature (independently) by setting the corresponding length scale hyperparameter equal to infinity, and observe that the predictor (based on RFK) fall in performance only when we remove the shortest distance between points. All these results make it a good candidate kernel over sets of points in general. However it is not always obvious to choose relevant features for given functions.

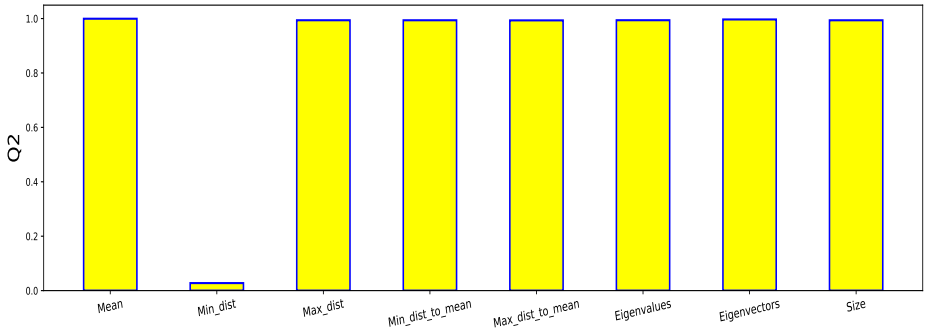

Figure 16: Effect of removing a feature on the $Q2$ of a GP with RFK on the $F_{minDist}$ function. The test set is random.

4.3.4 Bhattacharyya kernel

With the Bhattacharyya kernel, we observe in Tables 2 and 7 almost the same phenomenon that happens with Gaussian Wasserstein. We see that, under our experimental protocol, Bhatta yields poor predictive performances on all the considered test functions. However, there is a slight improvement of its scores on F_{40d} and F_{inert}. This is linked to the more isotropic nature of these functions, which better suits the Gaussian representation.

As a side comment, we provide below a closed form expression of the correlation according to the Bhattacharrya kernel, for two sets of points (in $\mathbb{R}^2$) that are dilations of each other. Notice that the expression does not depend on the set X.

Proposition 4.1 For a given isotropic dilation factor $\rho = 0.5$, $K(X, X_d)$ can be computed in a closed form: $K(X, X_d) = \frac{2\delta}{1+\delta^2}$.

The proof of this Proposition and a complementary illustration are given in Appendix B.

5 Conclusions and Perspectives

We have studied several candidate kernels for the Gaussian process modeling of functions taking unordered sets of vectors as inputs. We have showed, through numerical tests, that the performances of MMD-based kernels can be explained by the adaptation of its hyperparameters to the geometrical properties of the studied functions. In contrast, it is found that the kernels based on Gaussian representation

yield the worse performances, because these kernels tend to drown out the individual contribution of each point of the cloud into an average effect, causing a loss of information. We have also seen with the Relevant Features Kernel (RFK) that mapping the clouds of points into an Euclidean space and resorting to standard kernels (e.g., Squared Exponential) can perform better than existing methods such as n-MeanMap and compete, for some functions (and test cases), with MMD and Sliced Wasserstein kernels.These results obtained with the Relevant Feature kernel encourage us to search for generic methods for defining and computing suitable features on any given function.

Sliced Wasserstein kernels have also produced predictive models on many functions. These observations encourage to search for algorithms to find a reasonable number and adapted projection directions to approximate the Sliced Wasserstein Distance, for instance through a parametric representation of the projection directions coupled with a maximum of likelihood estimation.

The fact that MMD stumbles on the Mindist function calls for a non-uniform discrete probability model. This could allow for varying the weight of each point, depending on the importance of its contribution to the function of interest.

The ability of the different kernel-based models to extrapolate at the geometrically transformed (rotated and dilated) clouds is limited. Obviously, all kernels become less predictive on horizontal, vertical and rotated clouds of points. The decrease was more important with the horizontal and vertical dilations that departed more from the randomly generated training set. Structural properties of the kernels, independently of the learning set, also impact their sensitivity to certain transformations which, in turns, control the rate at which the GP predictions return to the GP mean. For example, MMD based kernels are more sensitive to dilations than our Relevant Features Kernel. Depending on the application, one should include in the design of experiments (DoE) these kinds of clouds of points or think for a more general criterion for the definition of the training data set. It not certain that a uniform design of experiments is always the right choice for an application in which only specific characteristics of the point clouds are of interest.

Finally, it should be noted that all our numerical tests have been performed on two-dimensional point clouds. Further experiments, assessing the coupled influence of the points dimension and the kernels on the GP predictions is still necessary.

References

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

[2] Haussler, David. Convolution Kernels on Discrete Structures. *Technical Report*. Departement of Computer Science, University of California at Santa Cruz. Vol. 646. 1999.

[3] Leslie, Christina and Eskin, Eleazar and Weston, Jason and Noble, William Stafford, Mismatch string kernels for SVM protein classification. *Advances in neural information processing systems*, MIT; 1998, 1441–1448, 2003.

[4] Jaakkola, Tommi and Diekhans, Mark and Haussler, David, A discriminative framework for detecting remote protein homologies. *Journal of computational biology*, Mary Ann Liebert, Inc. Vol. 7, 1-2, 95–114, 2000.

[5] Cortes, Corinna and Haffner, Patrick and Mohri, Mehryar, Rational kernels: Theory and algorithms. *Journal of Machine Learning Research*, Vol. 5, Aug, 1035–1062, 2004.

[6] Yoshikawa, Yuya and Iwata, Tomoharu and Sawada, Hiroshi and Yamada, Takeshi, Cross-domain matching for bag-of-words data via kernel embeddings of latent distributions. *Advances in Neural Information Processing Systems*, Vol. 28, 2015.

[7] Risi, Imre Kondor, Diffusion kernels on graphs and other discrete input spaces. *Proc. 19th Int. Conf. Machine Learning, 2002*, 2002.

[8] Bach, Francis R, Graph kernels between point clouds. *Proceedings of the 25th international conference on Machine learning*, 25–32, 2008.

[9] Suard, Frédéric and Rakotomamonjy, Alain and Bensrhair, Abdelaziz, Kernel on Bag of Paths For Measuring Similarity of Shapes. *proceedings - European Symposium on Artificial Neural Networks (ESANN'2007)*, Bruges (Belgium), 25-27 April 2007, ISBN 2-930307-07-2. 355–360, 2007.

[10] Gosselin, Philippe H and Cord, Matthieu and Philipp-Foliguet, Sylvie, Kernels on bags for multi-object database retrieval. *Proceedings of the 6th ACM international conference on Image and video retrieval*, 226–231, 2007.

[11] Precioso, Frédéric and Cord, Matthieu and Gorisse, David and Thome, Nicolas, Efficient bag-of-feature kernel representation for image similarity search. *2011 18th IEEE International Conference on Image Processing*, IEEE, 109–112, 2011.

[12] Barla, Annalisa and Odone, Francesca and Verri, Alessandro, Hausdorff kernel for 3D object acquisition and detection. *European Conference on Computer Vision*, Springer, 20–33, 2002.

[13] Chugh, Tinkle and Ymeraj, Endi, Wind Farm Layout Optimisation using Set Based Multi-objective Bayesian Optimisation. *arXiv preprint arXiv:2203.17065*, 2022.

[14] Simon, Ludovic and Tarel, Jean-Philippe, Utilisation de la Distance de Hausdorff dans un SVM. *congrès francophone des doctorants en STIC (MajecSTIC'06)*, 2006 (in French).

[15] Shimosaka, Masamichi and Mori, Taketoshi and Harada, Tatsuya and Sato, Tomomasa, Marginalized bags of vectors kernels on switching linear dynamics for online action recognition. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, IEEE, 3072–3077, 2005.

[16] Woznica, Adam and Kalousis, Alexandros and Hilario, Melanie, Distances and (indefinite) kernels for sets of objects. *Sixth International Conference on Data Mining (ICDM'06)*, IEEE, 1151–1156, 2006.

[17] Pekalska, Elzbieta and Paclik, Pavel and Duin, Robert PW, A generalized kernel approach to dissimilarity-based classification. *Journal of machine learning research*, Vol. 2, Dec, 175–211, 2001.

[18] Jebara, Tony and Kondor, Risi, Bhattacharyya and expected likelihood kernels. *Learning theory and kernel machines*, Springer, 57–71, 2003.

[19] Bhattacharyya, Anil, On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, Vol. 35, 99–109, 1943.

[20] Buathong, Poompol and Ginsbourger, David and Krityakierne, Tipaluck, Kernels over sets of finite sets using RKHS embeddings, with application to Bayesian (combinatorial) optimization. *International Conference on Artificial Intelligence and Statistics*, PMLR, 2731–2741, 2020.

[21] Muandet, Krikamol and Fukumizu, Kenji and Sriperumbudur, Bharath and Schölkopf, Bernhard and others, Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, Now Publishers, Inc., Vol. 10, 1-2, 1–141, 2017.

[22] Grauman, Kristen and Darrell, Trevor, The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, Now Publishers, Inc., Vol. 8, 4, 2007.

[23] Cuturi, Marco, Positive definite kernels in machine learning. *arXiv preprint arXiv:0911.5367*, 2009.

[24] Aronszajn, Nachman, Theory of reproducing kernels. *Transactions of the American mathematical society*, Vol. 68, 3, 337–404, 1950.

[25] Berg, Christian and Christensen, Jens Peter Reus and Ressel, Paul, *Harmonic analysis on semigroups: theory of positive definite and related functions.* Graduate Texts in Mathematics, Vol. 100, Springer, 1984.

[26] Haasdonk, Bernard and Bahlmann, Claus, Learning with Distance Substitution Kernels, *Joint pattern recognition symposium*, LNCS, Vol. 3175, Springer, 220–227, 2004.

[27] Paulsen, Vern I and Raghupathi, Mrinal, *An introduction to the theory of reproducing kernel Hilbert spaces.* Cambridge university press, Vol. 152, 2016.

[28] Panaretos, Victor M and Zemel, Yoav, Statistical aspects of Wasserstein distances. *Annual review of statistics and its application*, Annual Reviews, Vol. 6, 405–431, 2019.

[29] Bui, Thi Thien Trang and Loubes, Jean-Michel and Risser, Laurent and Balaresque, Patricia, Distribution regression model with a Reproducing Kernel Hilbert Space approach. *arXiv preprint arXiv:1806.10493*, 2018.

[30] De Plaen, Henri and Fanuel, Michaël and Suykens, Johan AK, Wasserstein exponential kernels. *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 1–6, 2020.

[31] Peyré, Gabriel and Cuturi, Marco and others, Computational optimal transport. *Center for Research in Economics and Statistics Working Papers*, IEEE, 2017-86, 2017.

[32] Kolouri, Soheil and Zou, Yang and Rohde, Gustavo K, Sliced Wasserstein kernels for probability distributions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5258–5267, 2016.

[33] Carriere, Mathieu and Cuturi, Marco and Oudot, Steve, Sliced Wasserstein kernel for persistence diagrams. *International conference on machine learning*, PMLR, 664–673, 2017.

[34] Muandet, Krikamol and Fukumizu, Kenji and Dinuzzo, Francesco and Schölkopf, Bernhard, Learning from distributions via support measure machines. *Advances in neural information processing systems*, Vol. 25, 2012.

[35] Pronzato, Luc and Zhigljavsky, Anatoly, Minimum-energy measures for singular kernels. *Journal of Computational and Applied Mathematics*, Elsevier, Vol. 382, 2021.

[36] Jebara, Tony and Kondor, Risi and Howard, Andrew, Probability product kernels. *The Journal of Machine Learning Research*, JMLR. org, Vol. 5, 819–844 2004.

A Tables of the Mean Absolute Errors (MAE) of the predictors

In the following m_1 and σ_1 correspond respectively to the mean and standard deviation of the real values of the test functions.

K \ $k_{\mathcal{H}}$	RBF	Laplacian	Matérn 3/2	Matérn 5/2
MMD	1.807	1.767	1.725	1.719
n-MeanMap	2.968	3.476	3.125	3.212

Table 10: MAE with different embedding kernels on F_{40d} with a random design of experiments.

Function \ Kernels	MMD	n-MeanMap	BHATTA	RFK	Slice-Wass	Gauss-Wass
F_0	2.405	4.747	7.424	2.399	3.129	7.240
F_{45}	2.873	4.916	7.388	2.412	3.193	7.238
F_{90}	2.497	4.809	7.450	2.655	2.985	7.319
F_{40d}	1.725	2.968	4.989	2.505	2.323	4.811

Table 11: MAE of 6 kernels on the 4 wind farm proxy functions with a random design of experiments.

Function	m_1	σ_1
F_0	57.129	9.541
F_{45}	57.185	9.554
F_{90}	57.090	9.545
F_{40d}	47.159	7.05

Table 12: m_1 and σ_1 of the testing data on the wind farm functions with a random design of experiments.

Function \ Kernels	MMD	RFK	Slice-Wass
F_0	3.990	17.170	15.424
F_{45}	4.835	9.533	5.680
F_{90}	6.161	4.314	5.328
F_{40d}	3.830	6.600	5.533

Table 13: MAE observed on horizontally dilated clouds of points.

Function	m_1	σ_1
F_0	11.832	4.522
F_{45}	18.527	8.206
F_{90}	25.413	10.229
F_{40d}	11.832	4.522

Table 14: m_1 and σ_1 of the testing data on the wind farm functions considering horizontally dilated clouds of points.

Function \ Kernels	MMD	RFK	Slice-Wass
F_0	6.917	4.194	4.143
F_{45}	4.159	8.921	6.896
F_{90}	3.944	12.483	16.254
F_{40d}	3.365	5.968	2.815

Table 15: MAE observed on vertically dilated clouds of points.

Function	m_1	σ_1
F_0	24.799	10.031
F_{45}	18.375	8.091
F_{90}	12.565	4.569
F_{40d}	11.494	4.507

Table 16: m_1 and σ_1 of the testing data on the wind farm functions considering vertically dilated clouds of points.

Function \ Kernels	MMD	RFK	Slice-Wass
F_0	3.601	3.203	4.691
F_{45}	3.479	2.975	3.762
F_{90}	3.292	3.630	4.715
F_{40d}	2.923	1.749	1.875

Table 17: MAE observed on isotropically dilated clouds of points.

Function	m_1	σ_1
F_0	36.067	17.482
F_{45}	35.777	17.601
F_{90}	36.311	17.319
F_{40d}	26.289	14.558

Table 18: m_1 and σ_1 of the testing data on the wind farm functions considering isotropically dilated clouds of points.

Function \ Kernels	MMD	RFK	Slice-Wass
F_0	2.910	2.538	3.251
F_{45}	3.187	2.367	3.076
F_{90}	2.986	2.318	3.111
F_{40d}	2.454	2.350	2.334

Table 19: MAE observed on rotated clouds of points

Function	m_1	σ_1
F_0	58.121	8.670
F_{45}	58.109	8.659
F_{90}	58.127	8.680
F_{40d}	48.099	7.052

Table 20: m_1 and σ_1 of the testing data on the wind farm functions considering rotated clouds of points.

Function \ Kernels	MMD	n-MeanMap	BHATTA	RFK	Slice-Wass	Gauss-Wass
F_{inert}	111.152	150.128	162.224	18.250	63.873	154.126
$F_{minDist}$	2.071	2.106	2.246	0.072	1.387	2.124

Table 21: MAE observed on $F_{minDist}$ and F_{inert} on random designs.

Function	m_1	σ_1
F_{inert}	945.585	268.012
$F_{minDist}$	3.821	2.652

Table 22: m_1 and σ_1 of the testing data on $F_{minDist}$ and F_{inert} on random designs.

Function \ Kernels	MMD	RFK	Slice-Wass
F_{inert}	110.191	33.878	79.292
$F_{minDist}$	1.906	0.051	0.918

Table 23: MAE observed on dilated clouds of points.

Function	m_1	σ_1
F_{inert}	422.974	310.933
$F_{minDist}$	2.006	1.577

Table 24: m_1 and σ_1 of the testing data on $F_{minDist}$ and F_{inert} considering dilated clouds of points.

Function \ Kernels	MMD	RFK	Slice-Wass
F_{inert}	151.726	18.824	74.663
$F_{minDist}$	1.732	0.058	1.391

Table 25: MAE observed on rotated clouds of points.

Function	m_1	σ_1
F_{inert}	956.016	272.758
$F_{minDist}$	3.210	2.102

Table 26: m_1 and σ_1 of the testing data on $F_{minDist}$ and F_{inert} considering rotated clouds of points.

B Dilation of the Bhattacharyya kernel

We want to compute the Bhattacharyya kernel $K(X, X_d)$ in the case where X_d is the dilated version of X.

We know that if $X = \{x_1, ..., x_n\}$, then $X_d = \{D_\delta x_1 + (I - D_\delta)\bar{X}, ..., D_\delta x_n + (I - D_\delta)\bar{X}\}$.

First, consider the case where $\bar{X} = m(X) = 0$, and by consequence $\bar{X}_d = m(X_d) = D_\delta \bar{X} = 0$. This implies that $K(X, X_d) = (2\pi)^{(1-2\rho)D/2}|\Sigma^+|^{1/2}|\Sigma(X)|^{-\rho/2}|\Sigma(X_d)|^{-\rho/2}$.

Knowing that $D_\delta = \delta I$, we have $\Sigma(X_d) = \delta^2\Sigma(X) \Rightarrow (|\Sigma(X_d)|)^{-\rho/2} = \delta^{-2\rho}(|\Sigma(X)|)^{-\rho/2}$.
We have $\Sigma^+ = (\rho(\Sigma(X))^{-1} + \rho\Sigma(X_d)^{-1})^{-1}$ and $\Sigma(X_d)^{-1} = \frac{1}{\delta^2}\Sigma(X)^{-1}$, then $\Sigma^+ = \frac{1}{\frac{\rho}{\delta^2}+\rho}\Sigma(X)$.
Therefore $K(X, X_d) = (2\pi)^{(1-2\rho)D/2}\frac{1}{\rho}\frac{\delta^2}{1+\delta^2}(|\Sigma(X)|)^{1/2}\delta^{-2\rho}(|\Sigma(X)|)^{-\rho}$

In the case where $\rho = 1/2$, we get $K(X, X_d) = \frac{2\delta}{1+\delta^2}$, which does not depend on the cloud X.
In the case where the cloud is not centered, we have $K(X, X_d) = \int_\Omega P(x)^\rho P'^\rho(x)\mathrm{d}x$ with $\Omega = \mathbb{R}^2$. A simple translation allows us to return to centered clouds and 0 mean Gaussians.
The closed-form expression for the correlation is compared to the expression based on Bhattacharyya kernel for varying δ in Figure 17. The two curves are perfectly superimposed (red on black).

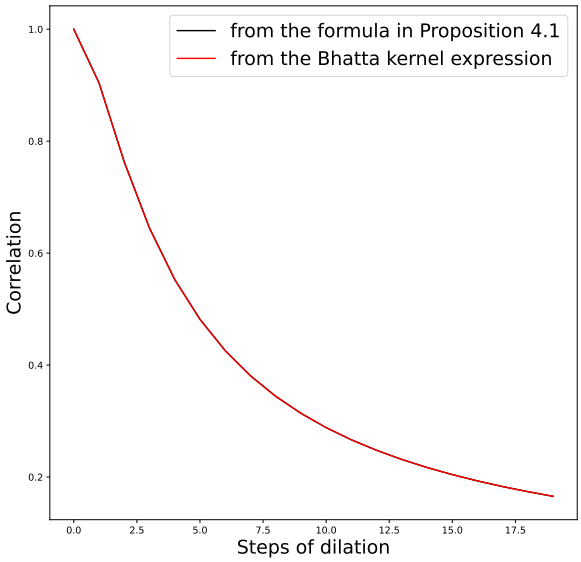

Figure 17: Correlation between two clouds isotropically dilated by a factor δ as expressed from the Bhattacharyya kernel (Equation (11)) and its specialization to this case (Proposition 4.1). The two curves are perfectly superimposed.

C Details about wind farm proxy

We consider the canonical basis of the plan (e_1, e_2) with $e_1 = (1, 0) \in \mathbb{R}^2$ and $e_2 = (0, 1) \in \mathbb{R}^2$. We denote by R_α the following transformation: $\begin{bmatrix}\cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha\end{bmatrix}$. For a given wind in the direction of e_1, the production function can be written as:

$$F(\{x_1, ..., x_n\}) = \sum_{i=1}^{n}\left(\prod_{j, j\neq i} f_p(x_j, x_i)\right) f_0(x_i) \tag{17}$$

We consider the case where f_0 is constant and $f_p(x_j, x_i)$ represents the wake effect on x_i that is caused by x_j. Its general formula, in the case where the direction of interaction coincides with the x-axis, is displayed below:

$$f_p(x_j, x_i) = \begin{cases} 1 & \text{if} \quad x_{i,1} \leq x_{j,1} \\ (\frac{||x_i - x_j||}{1+||x_i - x_j||} f_L(x_j, x_i) + \frac{1}{1+||x_i - x_j||} f_\theta(x_j, x_i)) f_{prox}(x_j, x_i) & \text{if} \quad x_{i,1} > x_{j,1} \end{cases} \tag{18}$$

Before describing f_L, f_θ and f_{prox}, we mention that the two terms of f_p are weighted with by functions depending on the distance between the two points. f_L corresponds to a power gain depending on the distance between turbines. Its expression is the following:

$$f_L = \frac{1}{1 + \exp(-p_1(\Delta - l))}$$

with p_1 and l parameters of the function and $\Delta = (x_{i,1} - x_{j,1})^2 + 6(x_{i,2} - x_{j,2})^2$. Concerning the second term, we have:

$$f_\theta(x_j, x_i) = \begin{cases} 0 & \text{if} \quad x_i = x_j \\ 1 & \text{if} \quad x_{i,1} = x_{j,1}, x_i \neq x_j \\ \frac{2}{\pi} \arctan(\frac{|x_{i,2} - x_{j,2}|}{|x_{i,1} - x_{j,1}|}) & \text{otherwise} \end{cases} \tag{19}$$

The component f_θ quantifies the gain depending on the angle $\overrightarrow{u}, \overrightarrow{x_i x_j}$, with $\overrightarrow{u}$ the direction of interaction. The result is multiplied with a third function in order to further penalize the proximity between turbines. We have

$$f_{prox}(x_j, x_i) = \frac{1}{1 + \exp(-p_2(\Delta' - radius))}$$

with p_2, $radius$ positive parameters and $\Delta' = \|x_i - x_j\|$. We can notice that all the terms of Eq. 18 are between 0 and 1, which results in $f_p \in [0, 1]$. In order to implement a function modeling the effect of a wind in the direction of $R_\alpha e_1$ (with $\alpha \in (0, 2\pi)$), we compute the coordinates of the inputs in the basis $(R_\alpha e_1, R_{\alpha+\pi/2} e_1)$ and apply the last function on the new inputs (with the new coordinates). The values of p_1, p_2 for the 3 functions defined above are respectively 0.15 and 0.5. We choose $l = 10$ and $radius = 3$ for F_0 and F_{45}. Concerning F_{40d}, for the parameters l and $radius$, 40 values are uniformly chosen respectively in $(1, 30)$ and $(1, 15)$.

D Invariant translation kernel

We know than we can make a kernel invariant under translation by computing the correlation between centered clouds of points. However, this approach is not considered in this paper in the above kernels. A numerical illustration is given below.

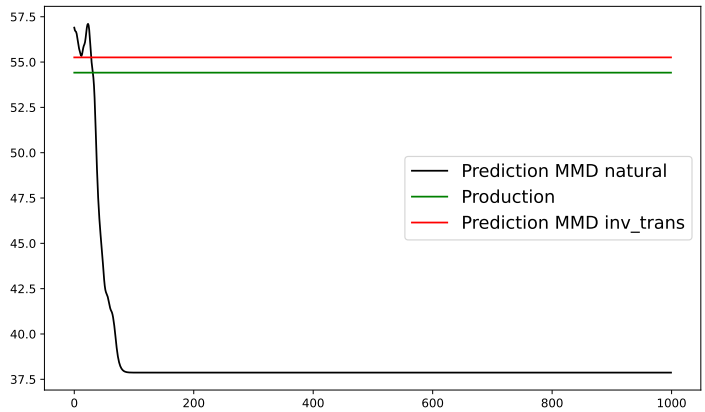

Figure 18: Prediction for a translated group of turbines, considering the wind farm proxy function.