Geometry-Aware Hierarchical Bayesian Learning on Manifolds

Yonghui Fan
Arizona State University
Tempe, Arizona
yfan61@asu.edu

Yalin Wang
Arizona State University
Tempe, Arizona
ylwang@asu.edu

Abstract

Bayesian learning with Gaussian processes demonstrates encouraging regression and classification performances in solving computer vision tasks. However, Bayesian methods on 3D manifold-valued vision data, such as meshes and point clouds, are seldom studied. One of the primary challenges is how to effectively and efficiently aggregate geometric features from the irregular inputs. In this paper, we propose a hierarchical Bayesian learning model to address this challenge. We initially introduce a kernel with the properties of geometry-awareness and intra-kernel convolution. This enables geometrically reasonable inferences on manifolds without using any specific hand-crafted feature descriptors. Then, we use a Gaussian process regression to organize the inputs and finally implement a hierarchical Bayesian network for the feature aggregation. Furthermore, we incorporate the feature learning of neural networks with the feature aggregation of Bayesian models to investigate the feasibility of jointly learning on manifolds. Experimental results not only show that our method outperforms existing Bayesian methods on manifolds but also demonstrate the prospect of coupling neural networks with Bayesian networks.

1. Introduction

Three-dimensional data on Riemannian manifolds, such as triangle meshes and point clouds as shown in Figure 1, is widely used to describe the shape information in object understanding, scene understanding, and many other vision tasks. Extracting and aggregating geometric features is considered the key to leveraging the intrinsic shape information of this type of data [8]. Recently, the Gaussian process (GP) based Bayesian learning emerges to be a study hotspot [56, 4, 20]. Theoretically, it has been proven that a single fully connected neural network (NN) layer with an infinity width is essentially a GP [38]. Further, this equivalence is extended to deep fully connected NNs and hierarchically connected GPs [24, 32]. Practically, encouraging results have been demonstrated in various applications [6]. In this work, we focus on developing GP-based Bayesian learning methods for solving vision tasks on manifolds. Specifically, we concentrate on two fundamental aspects: GP kernel design and Bayesian learning framework on manifolds.

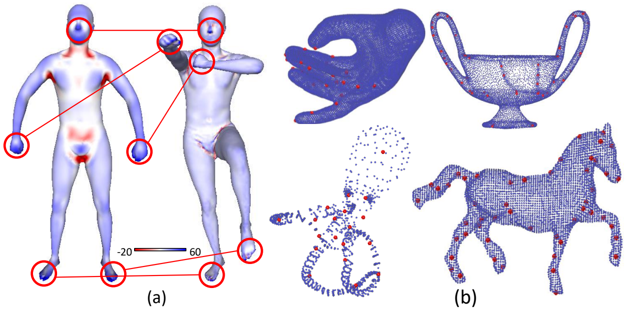


Figure 1. Examples of manifold-valued data. (a) Triangle meshes of two human poses. Meshes are rendered by normalized mean curvatures by MeshLab [10]. ROIs are marked by red circles; (b) Point clouds of different objects from McGill 3D shape benchmark [50]. Salient points based on GP regression are marked by red spheres.

Since the property of a zero-mean GP is largely determined by its kernel function [45], a primary goal is to develop an expressive kernel. Essentially, we aim at integrating two important characteristics into a shallow kernel structure: *geometry-awareness* and *intra-kernel convolution*.

Geometry-awareness stresses the capability of learning geometric features in the prior knowledge so that the posterior inference respects the representative regions of a 3D shape [21]. For example, when distinguishing different human poses, red circled regions of interest (ROIs) in Figure 1(a) are expected to be numerically highlighted because their regional features are more geometrically significant. The current strategy of achieving geometry-awareness is to directly add geometric feature descriptors to the kernel design [22, 19]. However, this strategy heavily relies on computing specific hand-crafted features, which potentially impedes the generality to broader types of applications. We propose a paradigm shift where only the point coordinates are needed. It enables our kernel to be geometry-aware on all commonly-used types of manifold-valued data.

Intra-kernel convolution introduces the convolutional filtering to the kernel construction so that the GP inference has a powerful feature aggregation ability [56]. This characteristic has been widely studied to increase the expressiveness of GPs [57, 15]. Most of the previous work uses the additive patch-wised computational structure to explicitly mimic the mechanism of convolutional NNs (CNNs) [14, 56]. But this approach is not feasible for manifold-valued data because of its off-the-grid structure. As one attempt, the graph convolutional GP (GCGP) in [57] used local coordinates to adjust the inputs to a uniform on-the-grid style. However, some drawbacks, such as the huge computational cost and the strict requirement on the input size, are noticed. Alternatively, we propose an implicit intra-kernel convolution. Mathematically, we rigorously show that the convolutional filtering can be delicately embedded into the kernel definition.

In this paper, we propose a hierarchical Bayesian model for manifold-valued tasks. The core is a kernel derived from a stochastic partial differential equation (SPDE) that generalizes a real physical process called *periodic potential diffusion process*. Two observations explaining why we choose this particular physical process are discussed in Sec. 4.1. Mathematically, we prove that the kernel implicitly integrates both the mean curvature flow, which is an effective geometric feature descriptor in $\mathcal{R}^3$, and a convolutional filtering. For tackling the irregular input dimensionality, we firstly use a GP-based salient point selection algorithm to obtain a uniform and light input, then, feeding it to the Bayesian network. Additionally, because the input of a Bayesian model on manifolds is the geometric features and NNs are strong in learning expressive features, we explore the potential of incorporating NNs with hierarchical Bayesian methods to leverage the strengths of both methods. Our contributions are summarized into three-folds:

(1) A kernel with both geometry-awareness and intra-kernel convolution properties. No hand-crafted feature is needed for involving geometric properties. The method is feasible to all commonly-used manifold-valued data;

(2) A Bayesian network for manifold-valued tasks, including a salient point selection module that non-linearly reduces the data dimensionality and organizes the irregular inputs;

(3) An exploration on NN+Bayesian approaches to leverage both the feature learning ability of NNs and the feature aggregation ability of the Bayesian methods.

Both empirical and numerical experimental results verify the effectiveness of our methods. We hope this work not only makes contributions to the Bayesian learning methods on manifolds but also sheds new light on integrating different learning mechanisms to maximize their learning power.

2. Related Work

Kernel design is always an important topic in GP-based Bayesian learning studies. Our initial thought originated from the problem addressed by Stein in [53] that the infinite differentiability of a Gaussian kernel led to an unrealistic match with the physical processes. Later, Stein proposed the well-known Matérn kernel family as a generalization of the Gaussian radial basis functions (RBF) to solve this problem. An intriguing idea is: why not derive a kernel function directly from the expression of a physical process? In this way, the kernel will intrinsically come with a real physical rationale. This idea was further strengthened by Särkkä's statement [47, 51] that any SPDE was a potential kernel. Given the fact that many physical processes are generalized by SPDEs, it is intuitive to combine the above two opinions together as a trustful theoretical foundation of kernel development [36]. For example, the Matérn kernel is actually the solution of a linear fractional SPDE [12, 48, 27].

However, classical kernels mainly dealt with data in the Euclidean space and considered less on the Riemannian manifolds. One solution was using Riemannian metric and space mapping techniques to achieve the domain transform [18, 37]. But clearly, this approach was not feasible to the data like volumetric meshes and point clouds. Alternatively, wrapping geometric features into kernels was proven to be effective [42, 9, 35, 21, 19]. For example, the weighted GP (W-GP) [22] yielded reasonable inferences after weighing the RBF kernel with the mean curvature and Gaussian curvature; the morphometric GP [19] used wave kernel signature metric and demonstrated good performances on 3-dimensional manifolds. These methods used explicit geometric expressions, which often relied on specific simplicial complex. Conversely, we implement an implicit geometric expression which is only sensitive to the distance lag.

Bayesian learning architecture is also an emerging topic. Additive GP [14] directly enabled the convolutional GP (CGP) [56]. We also use additive structure in our kernel design. The aforementioned GCGP is an implementation of CGP on graphs. It adjusted the irregular inputs by referring to the method used in graph convolutional networks (GCNs). Instead, we use a manifold learning strategy to organize the inputs. The studies of sparse variational GP and posterior estimation facilitated the development of deep GP (DGP) [13, 49, 28, 4]. In our hierarchical Bayesian model, we follow the framework of DGPs and use the doubly stochastic variational inference method [46].

3. Preliminaries

Some notations are defined here. Given a manifold-valued data $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F}) \in \mathbb{R}^d$ with a vertex set $\mathcal{V}$ of size $|\mathcal{V}|$, an edge set $\mathcal{E}$ of size $|\mathcal{E}|$ and a face set $\mathcal{F}$ of size $|\mathcal{F}|$. $\mathcal{E}$ and $\mathcal{F}$ can be empty. A distance lag between vertex $v \in \mathcal{V}$ and its neighborhood v' is denoted as $\|v\| = \|v - v'\|^2$. Define a GP, $GP(m, K)$, as a random process where the joint distribution of a finite collection of observations $Y = \{y_1, ..., y_n\}$ of samples $X = \{x_1, ..., x_n\}$ follows a multivariate Gaussian distribution: $p(Y|X) \sim \mathcal{N}(m, K)$, where m is the mean function and K is the covariance function or kernel. The dimension of the kernel matrix K is denoted by subscripts.

3.1. Periodic Potential Diffusion Process

Our kernel derivation originates from the periodic potential diffusion process. It is a special case of the reaction diffusion process. A reaction diffusion process $T(v,t)$ is the solution to a reaction diffusion equation [17, 54]:

$$\frac{\partial T(v,t)}{\partial t} = \alpha \Delta T(v,t) + F(v,t), \quad t \geq 0 \tag{1}$$

where Δ is the Laplace operator, constant α is 1. The initial condition is 0. F is the reaction function that defines the property of the energy source [16]. Given a certain F, there exists a corresponding physical scenario [54, 39]. The reaction function F can be expressed as the multiplication of a Dirac delta function at location v and a temporal function $h(t)$: $F = h(t)\delta(v - v')$. By defining the Green's function of Laplace operator under the Dirichlet boundary condition as $G(v, v', t)$ [2], T is equal to:

$$T(v, v', t) = \int_0^t G(v, v', t - s)h(s)ds \tag{2}$$

Reminding that the Green's function in an $\mathbb{R}^d$ diffusion problem has the standard form: $G = \frac{e^{-v^2/4t}}{(4\pi t)^{d/2}}$. When $h(t)$ is periodic: $h(t) = cos(\omega t)$, Eq. (2) is further derived as:

$$T = \int_0^t cos(\omega(t - s))\frac{e^{-v^2/4s}}{(4\pi t)^{d/2}}ds \tag{3}$$

Eq. (3) is called the *periodic potential diffusion process*, which is the theoretical foundation of our kernel.

3.2. Gaussian Process Regression

A GP regression (GPR) aims at learning a multivariate distribution that fits with the training data and predicts the observation y_{n+1} when a testing sample x_{n+1} arrives [45]. The Bayes' theorem is used to transform the prior knowledge to posterior inference in the learning process. As known, every finite marginal distribution of a GP still follows a multivariate Gaussian distribution. Therefore, the predictive distribution $\mathcal{N}(m', K')$ can be uniquely determined by the standard rules for conditioning Gaussian distributions:

$$m'_{(n+1)\times 1} = K^T_{n\times(n+1)} K^{-1}_{n\times n} Y_{n\times 1} \tag{4}$$

$$K'_{(n+1)\times(n+1)} = K_{(n+1)\times(n+1)} - K^T_{n\times(n+1)} K^{-1}_{n\times n} K_{n\times(n+1)} \tag{5}$$

$$K_{n\times(n+1)} = K^T_{(n+1)\times n} = \begin{pmatrix} K(x_1, x_1) & \cdots & K(x_1, x_n), & K(x_1, x_{n+1}) \\ \vdots & & & \vdots \\ K(x_n, x_1) & \cdots & K(x_n, x_n), & K(x_n, x_{n+1}) \end{pmatrix} \tag{6}$$

$$K_{(n+1)\times(n+1)} = \begin{pmatrix} K(x_1, x_1) & \cdots & K(x_1, x_n), & K(x_1, x_{n+1}) \\ \vdots & & & \vdots \\ K(x_n, x_1) & \cdots & K(x_n, x_n), & K(x_n, x_{n+1}) \\ K(x_{n+1}, x_1) & \cdots & K(x_{n+1}, x_n), & K(x_{n+1}, x_{n+1}) \end{pmatrix} \tag{7}$$

When new samples are continuously given, the GP is recursively updated. Later, we adopt the framework of GPR in a salient point selection algorithm. Each salient point is taken as a sample. We update the saliency map after adding the previous selection into the prior and then select the next one until a certain number of salient points are collected.

3.3. Deep Gaussian Processes with Doubly Stochastic Variational Inference

A DGP is a deep belief network that hierarchically concatenates multiple Gaussian process latent variable models together (GP-LVMs) [13]. It mimics the composition of restricted Boltzmann machines (RBMs) in NNs. The sparse variational inference is usually used in GPR to estimate the posterior and avoid the cubic complexity [49]. Suppose M inducing points $Z = \{z_1, ..., z_M\}(M \ll N)$ are selected, the complexity is decreased to $\mathcal{O}(M^2N)$ in a single GPR. For a DGP, the doubly stochastic variational inference is often applied to estimate the posterior [46, 4]. Specifically, the sparse variational inference is used to simplify the correlations within layers and keep the correlations between layers unchanged. In a DGP with L layers, the prior is recursively defined on a series of vector-valued stochastic functions $F = \{F^1, ..., F^L\}$. The i^{th} row of F^l is denoted as f_i^l. Function values at inducing points Z are U. Each single function has an independent Gaussian prior and inducing points. A joint density of a DGP can be expressed as:

$$p(Y, \{F^l, U^l\}_{l=1}^L) = \underbrace{\prod_{i=1}^N p(y_i|f_i^L)}_{likelihood} \underbrace{\prod_{l=1}^L p(F^l|U^l; F^{l-1}, Z^{l-1})p(U^l; Z^{l-1})}_{prior} \tag{8}$$

According to the theories of variational inference, a factorized form of the posterior joint density is defined as [46]:

$$q(\{f^l, U^l\}_{l=1}^L) = \prod_{l=1}^L p(f^l|f^{l-1}, U^l, Z^l)q(U^l) \tag{9}$$

where $q(U^l)$ is a Gaussian with mean function m^l and covariance function S^l for layer l. Eq. (9) indicates that the prediction of the l^{th} layer, f^l, depends on the previous prediction f^{l-1} and the inducing points of the current layer. By marginalising the approximation $q(U^l)$ from each layer, the i^{th} factorized variational posterior of the final layer is the integral of all paths $(f_i^1, ..., f_i^L)$ through the Gaussian distributions defined by parameters m^l, and S^l:

$$q(f_i^L) = \int \prod_{l=1}^{L-1} q(f_i^l | m^l, S^l; f_i^{l-1}, Z^{l-1}) df_i^l \tag{10}$$

The objective function is the doubly stochastic evidence lower bound (ELBO):

$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_{q(f_i^L)}[logp(y_i|f_i^L)] - \sum_{l=1}^{L} \mathcal{KL}(q(U^l)||p(U^l)) \tag{11}$$

where $\mathcal{KL}$ is the Kullback–Leibler divergence. The ELBO has the complexity $\mathcal{O}(M^2N(D^1 + ... + D^L))$ to compute, D^l is the size of the l^{th} layer. The variational expection likelihood $\mathbb{E}$ in Eq. 11 is computed using the Monte Carlo approximation. Please refer to [46] for more details.

4. Methods

4.1. Geometry-Aware Convolutional Kernel

In this section, we start by briefly explaining what motivates us to choose the periodic potential diffusion process as the theoretical basis. Then, we provide two implementations of the kernel for different applications. Furthermore, we introduce our theoretical analysis about the property of the kernel. In the end, a hierarchical Bayesian model is defined.

The idea of using periodic potential diffusion process comes from two observations:

The first observation is that the integral Laplace transform of function $f(t) = t^{d-1}e^{-\frac{1}{4}at}$ in $\mathbb{R}^d$ in Eq. (12) (which is a special upper incomplete gamma function, a is a constant) has several similar terms with the Matérn kernel in Eq. (13):

$$\int_0^\infty t^{d-1}e^{-\frac{1}{4}at}e^{-st}dt = 2\left[(\frac{1}{4}a)^{\frac{1}{2}}s^{-\frac{1}{2}}\right]^d \mathcal{K}_d(a^{\frac{1}{2}}s^{\frac{1}{2}}) \tag{12}$$

$$C(\tau) = \frac{\sigma^2}{\Gamma(d)2^{d-1}}(2\sqrt{d}\tau\kappa)^d\mathcal{K}_d(2\sqrt{d}\tau\kappa) \tag{13}$$

Both equations have modified Bessel function of the second kind $\mathcal{K}_d$, and the rest parts are functions with the same dimension order d. This indicates the possibility of deriving a kernel from Eq. (12).

The second observation is that the Green's function of Laplace-Beltrami operator Δ in the 3D diffusion problem belongs to the family of $t^{d-1}e^{-\frac{1}{4}at}, d = 3$. Bochner's theorem states that a stationary kernel K is positive definite in $\mathbb{R}^d$ if it is the Fourier transform of a positive bounded measure function [5]. Taking Eq. (3) as the real part of the Fourier transform (decomposing the exponential term with Euler's formula), it is already a stationary kernel function regarding the temporal variable t. If the spatial part is also proven to be positive semi-definite (PSD), then the periodic potential diffusion process T is a valid kernel function.

Summarizing these two observations and incorporating the background in Sec. 3.1, our goal is to derive a close-form expression from Eq. (3) and prove that this expression is PSD regarding its spatial variable v. Unfortunately, the integral in Eq. (3) has no explicit solution according to [47].

Alternatively, we apply the same approximation strategy in [20] to estimate the Eq. (3) as the combination of a cosine Fourier transform$\hat{f}_c(\omega)$ and a sine Fourier transform$\hat{f}_s(\omega)$:

$$\begin{aligned} T = \\ cos(\omega t)\int_0^t cos(\omega s)G(s)ds + sin(\omega t)\int_0^t sin(\omega s)G(s)ds \\ \approx cos(\omega t)\hat{f}_c(\omega) + sin(\omega t)\hat{f}_s(\omega) \end{aligned} \tag{14}$$

By solving this approximated form, we have the closed-form periodic potential diffusion process:

$$T = \frac{1}{4\pi}e^{-\|v\|\sqrt{\frac{1}{2}\omega}}cos(\|v\|\sqrt{\frac{1}{2}\omega} + \omega t) \tag{15}$$

For simplicity, we define a frequency term $\lambda = \sqrt{\frac{1}{2}\omega}$ and a phase term $\phi = \omega t$. λ and ω are hyper-parameters that are determined by regular parameter tuning methods. Reminding that the diffusion process is dynamic, we can express it as the accumulation of values at N time slots. Therefore, the final kernel definition is expressed as an additive kernel [14]:

$$K(\|v\|, \lambda, \phi) = \frac{1}{4\pi}\sum_{n=1}^{N} e^{-\lambda_n\|v\|}cos(\lambda_n\|v\| + \phi_n) \tag{16}$$

For regression tasks, the implementation in Eq. (16) is sufficient. But for better fitting with a hierarchical Bayesian model, we further introduce an implementation by taking the kernel as an ARD covariance function [23]. An individual length-scale parameter α is added for each input dimension which determines the relevancy of the input to the task:

$$K(\|v\|, \lambda, \phi, \alpha) = \frac{1}{4\pi}\sum_{n=1}^{N} e^{-\lambda_n\frac{\|v\|}{\alpha_n}}cos(\lambda_n\frac{\|v\|}{\alpha_n} + \phi_n) \tag{17}$$

GPs defined with Eq. (16) or Eq. (17) can be taken as the mixture of GPs. Previous studies show that the mixture of Gaussian has a universal approximation ability in fitting with continuous distributions [55, 40].

The next goal is to prove that Eq. (16) is PSD regarding its spatial variable. Eq. (16) is the composition of two functions: $e^{-\lambda\|v\|}$ and $cos(\lambda\|v\| + \phi)$. It is acknowledged that the exponential function and the cosine function are PSD regarding

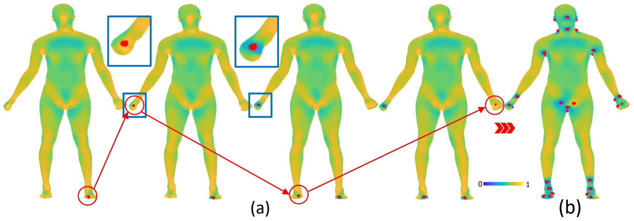


Figure 2. Demonstration of salient point selection on a human pose model. The meshes are rendered by normalized uncertainty scores at each iteration. (a) The progress of selecting the first four salient points. Salient points are remarked by red spheres. Noticing the color change before and after the selection in two zoom-in regions. (b) The first 30 salient points. The high uncertainty score regions are still centralized in ROIs after selecting the 30^{th} salient point.

variable v. According to the composition property of a PSD function, the result of positive real function/constant times a PSD function is still PSD [3]. So, Eq. (16) is spatially PSD. Eq. (16) can be taken as a compositional kernel. Eq. (1) indicates a family of kernels. This opinion is generalized as:

Theorem 1. *A real-valued function $T(v, t)$ on $\mathbb{R}^d$ is a spatial-temporal kernel function if it is a linear/non-linear diffusion process: $\frac{\partial T}{\partial t} = \alpha \Delta T + P(t)\delta(v)$, where α is a positive constant, $P(t)$ is a periodic function, $\delta(v)$ is the Dirac delta function, and Δ is the Laplace operator.*

Proof. The proof is provided in Supplementary. □

The kernel implemented in Eq. (16) or Eq. (17) is named as the geometry-aware convolutional (GAC) kernel. The GAC kernel satisfies the two characteristics discussed in Introduction, which are summarized as two lemmas:

Lemma 1. *The GAC Kernel embeds the mean curvature flow in $\mathbb{R}^3$, which enables it to be geometry-aware.*

Lemma 2. *The GAC Kernel embeds a convolution filtering within the kernel structure, called intra-kernel convolution.*

Proof. The proofs are provided in the Supplementary. □

We validate the proposed GAC kernel in two different studies. The first one is to use it in a regular GPR model. An unsupervised salient point selection algorithm will be introduced. This implementation fully utilizes geometry-awareness property. The second one is to adopt it in a Bayesian network layer. Hierarchical deep Bayesian learning models will be discussed. The feature aggregation will benefit from the nice intra-kernel convolution property.

4.2. Unsupervised Salient Point Selection

GP regression (GPR) is widely used in spatial inference known as the "Kriging" method [45, 21]. Being inspired by the landmarking in face recognition [58], we use a GPR-based unsupervised salient point selection to process the irregular inputs. This method is essentially a manifold learning technique [30, 34, 44]. Therefore, applying our method also achieves nonlinear data dimensionality reduction. One advantage of GPR is the availability of uncertainty estimation [45]. We leverage this advantage and define the saliency score as the variance-based uncertainty [61]. By iteratively selecting new salient points and adding previously selected points to the prior, we successively collect a set of salient points. A geometry-aware kernel guarantees the salient points are significant to represent the original massive data. This strategy has been successfully applied in [34, 21, 19].

Suppose a set of κ salient points is denoted as $\tilde{v} = \{\tilde{v}^1, ..., \tilde{v}^\kappa\}$. Define $\lambda = \sqrt{0.2\pi n}, n = [1, ..., N_{fre}]$. ϕ equals to a N_{fre}-length vector by dividing $[0, \frac{\pi}{2}]$ into N_{fre} equal line-spaces. A multi-frequency multi-phase GAC kernel (MMK) is defined in a weighted squared form: $MMK = K \times W \times K$ (K is symmetric). The weight W is a diagonal matrix with the sum of absolute values of each row in GAC kernel K as the diagonal entries: $W(v) = \sum |K(v, \cdot)|$. The saliency score $\Sigma_{\mathcal{M}}$ of v_i during selecting the $(\kappa + 1)^{th}$ salient point is defined as:

$$\Sigma_{\mathcal{M}}^{\kappa+1}(v_i) = K(v_i, v_i) - K(v_i, \tilde{v}^\kappa) K^{-1}_{\tilde{v}^\kappa, \tilde{v}^\kappa} K^T(v_i, \tilde{v}^\kappa) \tag{18}$$

$$K(v_i, \tilde{v}^\kappa) = \begin{pmatrix} K(v_i, \tilde{v}^1) \\ \vdots \\ K(v_i, \tilde{v}^\kappa) \end{pmatrix}_{\kappa \times 1} \tag{19}$$

$$K_{\tilde{v}^\kappa, \tilde{v}^\kappa} = \begin{pmatrix} K(\tilde{v}^1, \tilde{v}^1) & \cdots & K(\tilde{v}^1, \tilde{v}^\kappa) \\ \vdots & & \vdots \\ K(\tilde{v}^\kappa, \tilde{v}^1) & \cdots & K(\tilde{v}^\kappa, \tilde{v}^\kappa) \end{pmatrix}_{\kappa \times \kappa} \tag{20}$$

Only the point with the highest uncertainty score is selected as the salient point: $\tilde{v} := argmax_v \Sigma$. The first salient point is the vertex with the maximum variance in MMK. From Eq. (18)-(20) we can see that a newly-selected salient point will be added to the prior knowledge and the next saliency score is determined by the previous selections. The whole process follows a GPR framework. The algorithm is summarized in Algorithm 1. Figure 1(b) shows examples of selecting salient points on point clouds. Figure. 4 demonstrates salient points on triangle meshes. Further evaluation results are available in Experiments.

4.3. Hierarchical Bayesian Model on Manifolds

We define a GAC-GP layer with the GAC kernel and follow the framework of DGPs to construct a hierarchical Bayesian learning model by stacking up multiple GP layers. Thanks to the intra-kernel convolution property, the GAC-GP layer has a good feature aggregation ability. In a pure hierarchical Bayesian learning model on manifolds, a computational pipeline is shown in Figure. 3. The first step is to

Algorithm 1. Unsupervised Salient Point Selection

```
procedure GPR($\mathcal{M}, \kappa$)▷ Manifold $\mathcal{M}$, $\kappa$ salient points
    $N \leftarrow$ KNN or Fast Marching ▷ calculate $N$ nearest neighbors of each point
    $MMK = K \times W \times K \leftarrow$ Kernel construction
    $\tilde{V} \leftarrow \varnothing$          ▷ initialize landmarks set $\tilde{V}$ as empty
    while $k \leq \kappa$ do
        if k=1 then
            $\Sigma_{\mathcal{M}}(v_i) \leftarrow max(diag(MMK))$
        else
            $\Sigma_{\mathcal{M}}(v_i)$     ▷ calculate the uncertainty score
        end if
        $\tilde{v}^k \leftarrow argmax\Sigma_{\mathcal{M}}$
        $k \leftarrow k + 1$
    end while
    return $\tilde{V} = \{\tilde{v}^1, ..., \tilde{v}^\kappa\}$
end procedure
```

process the input. Assume κ salient points are selected by Algorithm 1. The feature on each salient point $f_{\tilde{v}}$ is a vector of length l. For shape i, we link all features of salient points in the order of their selections: $f_i = \{f_{\tilde{v}^1}...f_{\tilde{v}^\kappa}\}$. Noting that all shapes here belong to the same dataset and all salient points are selected with the same parameter setting in Algorithm 1. Otherwise, point-to-point registration is needed to concatenate features. Suppose H shapes are used, the input X is an $H \times (\kappa \times l)$ matrix. We compose a sequence of layers that map the input x_i to its label y_i in a hierarchical Bayesian model for classifications:

$$\underbrace{x_i = f^0}_{1\times(\kappa\times l)} \overset{\mathcal{GP}_0}{\rightarrow} \underbrace{f^1}_{1\times S^1} \rightarrow \cdots \overset{\mathcal{GP}_{L-1}}{\rightarrow} \underbrace{f^L}_{1\times C} \overset{softmax}{\rightarrow} \underbrace{y_i}_{C_i} \qquad (21)$$

The output of hidden layer l is a vector of the size $1 \times S^l$, where S^l is the layer size. This is similar to the relationship of the input channel and output channel in a NN. When the batch processing is applied, the output of each hidden layer has dimension $B \times S^l$, B is the batch size. A final layer is appended with a softmax multi-class likelihood. The output vector has the dimension $1 \times C$, where C is the number of classes. Each entry stands for the probability belonging to a certain class. Arbitrary numbers of GP layers can be added as hidden layers. We use the doubly stochastic variational inference approach to estimate the posterior [46]. The optimization process is to maximize the ELBO in Eq. (11). The K-means method is used to choose inducing points.

Because the input of the Bayesian model is the point-wise features on manifolds, and NNs are strong in feature learning, we are inspired to further explore the potential of NN+Bayesian methods. Such a mixed model can take advantage of both the feature learning ability of NNs and the feature aggregation ability of Bayesian models. We generally follow the pipeline in deep kernel learning [60]. The input format is determined by the NN part. The output of NNs is the shape feature. The feature is then fed into a Bayesian model, and the following processing is the same as the pure Bayesian method. The negative marginal log-likelihood (MLL) is used as the loss function.

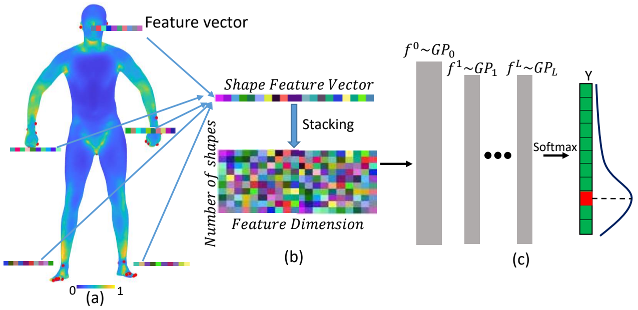


Figure 3. Pipeline of human pose retrieval. (a) Point-wise feature computation and salient point selection. The mesh is rendered by normalized mean curvatures. (b) Shape feature preparation. (c) Hierarchical Bayesian learning model for feature aggregation and inference. A softmax likelihood function is used at the last layer.

5. Experiments

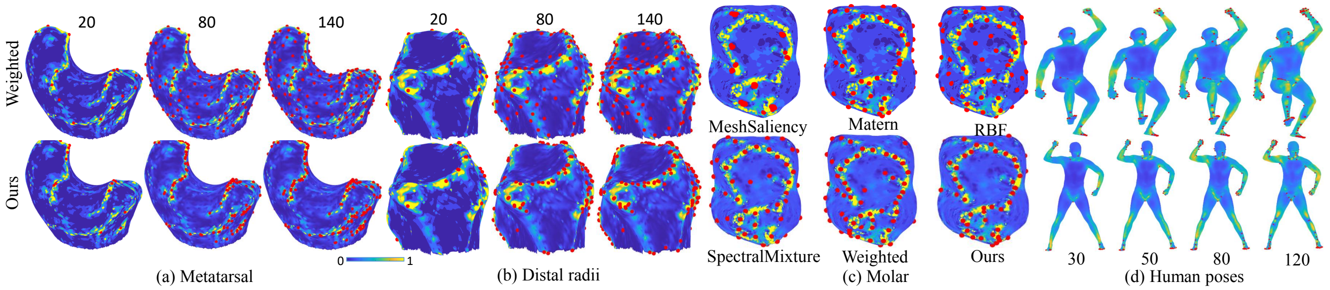
We evaluate our methods with three experiments. Our method is noted as GAC-GP. Applications are implemented in Pytorch and GPytorch with GPU acceleration [23].

5.1. Unsupervised Salient Point Selection

In the first experiment, the task is to select salient points on manifolds. The purpose is to evaluate the geometry-awareness of the GAC-GP and its stability in continuous regressions. Additionally, we compare the computational efficiency by recording the average running time. When defining MMK, we use the fast marching algorithm to compute geodesic distances and only select $k = 200$ nearest neighboring points for each vertex. N_{fre} is set as 5.

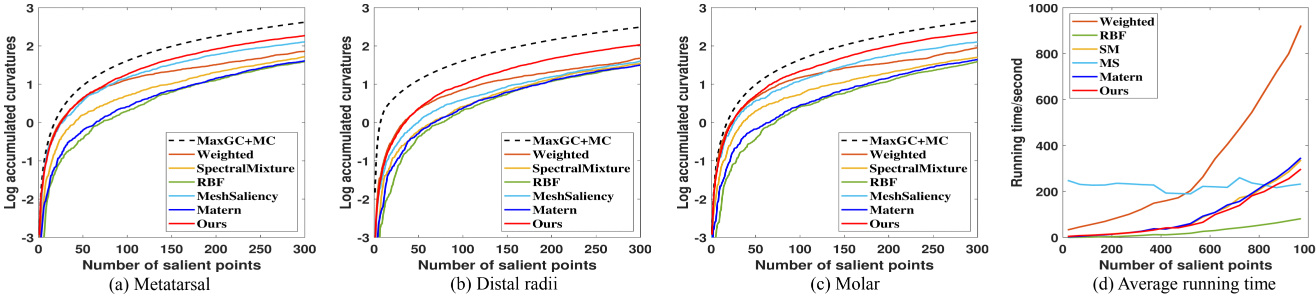
Three datasets are used [7]: (i) "Mandibular molars", or "molar", contains 116 teeth shapes; (ii) "First metatarsals", or "metatarsal", contains 57 shapes; and (iii) Distal radii contains 45 shapes. All shapes are triangle meshes with around 5000 vertices. Examples of each dataset are illustrated in Figure 4(a)-(c). The ROIs of such data are usually the marginal ridges, teeth crowns, and outline contours where the geometric features are rich [7, 11]. Therefore, the meshes in Figure. 4(a)-(c), are rendered by the normalized mean curvature as the ground-truth [26, 25]. The salient points are expected to evenly distribute in yellow regions.

Comparison methods include: (1) RBF kernel GP (RBF, RBF-GP) [45]. RBF is a classical choice; (2) spectral mixture kernel GP (SM, SpectralMixture, SMK-GP) [59]. SMK-GP had good performances in many vision tasks. 10 mixtures are used; (3) Matérn 3/2 kernel GP (Matérn, Matérn-



(a) Metatarsal (b) Distal radii (c) Molar (d) Human poses

Figure 4. Visualization of salient points. All meshes are rendered by the normalized mean curvature. The ROI is estimated to be the high-curvature yellow region. The salient points are marked by the red spheres. (a) and (b) illustrate 20, 80, 140 salient points on metatarsal and distal radii data. The upper row is the results of W-GP [22], the bottom row is ours. (c) illustrates 50 salient points selected by comparison methods and ours on a molar model. (d) shows 30-120 salient points on two human poses. The saliency transition is visible.



(a) Metatarsal (b) Distal radii (c) Molar (d) Average running time

Figure 5. (a)-(c) The log accumulated AC curve of each dataset. (d) Average running time of selecting one salient point.

GP) [52]. Matérn $1/2$ and $5/2$ are excluded because of worse results; (4) mesh saliency (MeshSaliency, MS) [31]. It is a classical saliency detection method; (5) Weighted GP (Weighted, W-GP) in [22]. W-GP is a state-of-the-art GP method on manifolds. The same parameter settings are used.

Figure 4(a)-(c) demonstrate salient point selection results. In (a) and (b), we focus on comparing ours with the W-GP because it is the only GP kernel method that stresses the geometry-awareness, and it outperforms all other comparison methods. We show 20, 80, and 140 salient points. When selecting a small number of salient points, both methods present reasonable results. But the accuracy of W-GP gradually drops with the iterations increasing while GAC-GP shows a much better stability of geometry-awareness. For other comparison methods, we give examples of 50 salient points on one Molar shape in Figure 4(c). Figure. 1(b) shows some examples of salient points on point clouds.

Numerically, we define an Accumulated Curvature (AC) value to measure the selection performance: $AC_N = log \sum_{k=1}^{\kappa}(|GC_k| + |MC_k|)$, where GC and MC are normalized Gaussian curvature and mean curvature, and κ is the total number of salient points. Drawing the AC values with the increased number of salient points forms an AC curve. Within a dataset, we compute the log average of AC values to plot an average AC curve. This curve reflects the geometry-awareness of different selection methods. When only selecting points with the largest accumulated curvatures at each iteration and draw the AC curve with these maximum values, we get a MaxGC+MC AC curve. This curve stands for the upper bound of saliency selections. The higher and closer to this MaxGC+MC curve an AC curve is, the better the geometry-awareness ability a corresponding method has. The results are plotted in Figure 5(a)-(c). The MaxGC+MC AC curve is drawn in dashes. Generally, AC curves of GAC-GP are above all other methods. Both the empirical visualizations and the numerical measurements verify that the GAC-GP is capable of making geometry-aware inference on manifolds. More importantly, its geometry-awareness is still consistent and stable after continuous regressions.

The computational efficiency is measured by averaging the running time of selecting one salient point, as shown in Figure 5(d). GPR-based methods gradually slow down due to the increased prior knowledge as shown in Eq. (18)-(20). The GAC-GP enjoys strong computational efficiency considering its superior performance because usually only a small number of salient points are needed.

5.2. Human Pose Retrieval

In the second experiment, the task is to classify different human poses modeled by triangle meshes. The first purpose is to further evaluate the salient point selections by fixing the Bayesian learning architecture. The second purpose is to fix the inputs and evaluate different hierarchical Bayesian learning architectures. The pipeline in Figure. 3 is used.

Table 1. Results of human pose retrieval with Bayesian models defined by different kernels and numbers of salient points.

SIWKS	RBF-GP	W-GP	Matern-GP	SMK-GP	MS	GAC-GP
50	0.850	0.898	0.885	0.898	0.866	**0.915**
100	0.859	0.901	0.886	0.908	0.872	**0.921**
250	0.862	0.905	0.890	0.912	0.899	**0.925**

Table 2. Human pose retrieval with different Bayesian learning architectures.

Method	Accuracy
GCGP*	91.2%
1RBF	91.1%
1GAC	92.5%
1GAC(10)+1RBF	92.7%
1GAC(10)+1GAC	**93.4%**

*(Using self-reproduced code.)

We choose the scaled-invariant wave kernel signature (SIWKS) [33, 1] as the feature. When computing the SIWKS, 30 smallest eigenvalues are used. The other parameter settings are the same as those in [1]. Features of 50, 100, and 250 salient points are used. In customizing the Bayesian model, we use the multitask variational strategy and the softmax likelihood in GPytorch. The number of inducing points is 50. We use Adam as the optimizer with an initial learning rate of 0.001. After the first 200 epochs, the learning rate changes to 10^{-4}. We trained for 2000 epochs. The cost function is the variational ELBO mentioned in Sec 3.3. The comparison methods are the same as the prior experiment.

The SHREC14 non-rigid 3D human model is used [41]. It contains 400 triangle meshes of 40 human subjects with 10 poses. Each mesh has about 15000 vertices. The dataset is randomly split into: 90% for training, 5% for validation, and 5% for testing. For the first purpose, we fix the Bayesian model to be a one-layer GAC-GP and feed in features from different methods. Table 1 shows the results. Classification with the features of all points has an accuracy of 0.910. Taking this value as a reference, we can draw conclusions that (1) our strategy of selecting salient points works for distinguishing different shapes. When enough salient points are selected, it is possible to use a small subset to represent the original data; (2) the geometry-aware selection of GAC-GP is more distinguishable than other comparison methods. For the second purpose, we fix the inputs to be GAC-GP salient features and evaluate different Bayesian learning architectures. Here we use GCGP [57] as a comparison method. The results are shown in Table. 2. Noting that the code of GCGP is not available and we use our implementations, so we put a star mark on GCGP's result. We can see that the accuracy is generally increased after adding a GAC layer, supporting a strong feature aggregation property. Meanwhile, a hierarchical concatenation of GAC layers shows a better accuracy than the single layer structure.

Table 3. Multi-class classifications on ModelNet40.

Method	Error rates
PCNN	86.1%
PointNet++	90.7%
PointNet++ +Normal	91.9%
PCNN+1GAC	**87.2%**
PointNet++ +1GAC	**91.8%**
PointNet++ +Normal+1GAC	92.1%
PointNet++ +Normal+2GAC	92.8%
PointNet++ +Normal+3GAC	**93.1%**

5.3. Point Cloud Classification

In the third experiment, the task is to classify different point cloud models. Our purpose is to demonstrate the work of integrating NNs with Bayesian learning. Here, we use the hierarchical feature learning architecture in PointNet++ [43] to learn the pointwise feature. We perform multi-class classification on ModelNet40 which contains 12311 3D CAD models of 40 categories. Each point cloud has 10000 points. We use 9843 models for training and 2468 models for testing. In the feature aggregation part, we use one single GAC-GP layer (ten mixtures). 64 inducing points are used. The optimizer is Adam and the initial learning rate is 0.04. The comparison methods include PointNet++, PointNet++ with normal information, and the Pointwise Convolutional NNs (PCNN) [29]. Table 3 shows that (1) the mechanism of NN+Bayesian can be jointly trained for tasks on manifolds; (2) models with Bayesian aggregation layers generally outperforms the classical multiple fully connected layers in our tests. We notice that the performance gain of using single GAC layer shrinks after adding normal information. Our hypothesis is that the features become more complicated, and the inference capability of single GAC layer is not powerful enough to well aggregate the new features. By adding 2&3 GAC layers, the improvements increase to 0.9% and 1.2%, respectively. The overall results show that architectures with GAC layers universally perform better than their original versions, which proves that such a co-design benefits the performance. A reasonable outlook is to investigate more effective architectures that integrate both methods for end-to-end tasks on manifolds.

6. Conclusion

In this work, we propose the GAC kernel that carries properties of geometry-awareness and intra-kernel convolution. Our methods show strong feature aggregation capability in various tasks on manifolds. We hope our work may inspire future Bayesian and NN+Bayesian studies on manifolds.

Acknowledgements This work was funded by grants R01EY032125 and R21AG065942.

References

[1] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 1626–1633. IEEE, 2011.

[2] James V Beck, Kevin D Cole, A Haji-Sheikh, and Bahman Litkouhl. *Heat conduction using Green's function*. Taylor & Francis, 1992.

[3] Rajendra Bhatia. Positive definite matrices, princeton ser. *Appl. Math., Princeton University Press, Princeton, NJ*, 2007.

[4] Kenneth Blomqvist, Samuel Kaski, and Markus Heinonen. Deep convolutional gaussian processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 582–597. Springer, 2019.

[5] Salomon Bochner. *Harmonic analysis and the theory of probability*. Courier Corporation, 2005.

[6] Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Matérn gaussian processes on riemannian manifolds. *Advances in Neural Information Processing Systems*, 2020.

[7] Doug M Boyer, Yaron Lipman, Elizabeth St Clair, Jesus Puente, Biren A Patel, Thomas Funkhouser, Jukka Jernvall, and Ingrid Daubechies. Algorithms to automatically quantify the geometric similarity of anatomical surfaces. *Proceedings of the National Academy of Sciences*, 108(45):18221–18226, 2011.

[8] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[9] Ismaël Castillo, Gérard Kerkyacharian, and Dominique Picard. Thomas bayes' walk on manifolds. *Probability Theory and Related Fields*, 158(3-4):665–710, 2014.

[10] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In Vittorio Scarano, Rosario De Chiara, and Ugo Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008.

[11] Sébastien Couette and Jess White. 3d geometric morphometrics and missing-data. can extant taxa give clues for the analysis of fossil primates? *Comptes Rendus Palevol*, 9(6-7):423–433, 2010.

[12] Noel Cressie and Christopher K Wikle. *Statistics for spatio-temporal data*. John Wiley & Sons, 2015.

[13] Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

[14] Nicolas Durrande, David Ginsbourger, and Olivier Roustant. Additive covariance kernels for high-dimensional gaussian process modeling. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 21, pages 481–499, 2012.

[15] Vincent Dutordoir, Mark Wilk, Artem Artemev, and James Hensman. Bayesian image classification with deep convolutional gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 1529–1539. PMLR, 2020.

[16] Klaus Ecker. Heat equations in geometry and topology. *Jahresber. Deutsch. Math.-Verein*, 110(3):117–141, 2008.

[17] Gert Ehrlich and Kaj Stolt. Surface diffusion. *Annual Review of Physical Chemistry*, 31(1):603–637, 1980.

[18] Ludwig Fahrmeir, Thomas Kneib, Stefan Lang, and Brian Marx. *Regression: models, methods and applications*. Springer Science & Business Media, 2013.

[19] Yonghui Fan, Natasha Lepore, and Yalin Wang. Morphometric gaussian process for landmarking on grey matter tetrahedral models. In *15th International Symposium on Medical Information Processing and Analysis*, volume 11330, page 113300H. International Society for Optics and Photonics, 2020.

[20] Yonghui Fan and Yalin Wang. Convolutional bayesian models for anatomical landmarking on multi-dimensional shapes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 786–796. Springer, 2020.

[21] Tingran Gao, Shahar Z Kovalsky, Doug M Boyer, and Ingrid Daubechies. Gaussian process landmarking for three-dimensional geometric morphometrics. *SIAM Journal on Mathematics of Data Science*, 1(1):237–267, 2019.

[22] Tingran Gao, Shahar Z Kovalsky, and Ingrid Daubechies. Gaussian process landmarking on manifolds. *SIAM Journal on Mathematics of Data Science*, 1(1):208–236, 2019.

[23] Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

[24] Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. Deep convolutional networks as shallow gaussian processes. *The International Conference on Learning Representations (ICLR)*, 2019.

[25] Gaël Guennebaud, Marcel Germann, and Markus Gross. Dynamic sampling and rendering of algebraic point set surfaces. In *Computer Graphics Forum*, volume 27, pages 653–662. Wiley Online Library, 2008.

[26] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces. In *ACM Transactions on Graphics (TOG)*, volume 26, page 23. ACM, 2007.

[27] Peter Guttorp and Tilmann Gneiting. Studies in the history of probability and statistics xlix on the matern correlation family. *Biometrika*, 93(4):989–995, 2006.

[28] James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. Mcmc for variationally sparse gaussian processes. In *Advances in Neural Information Processing Systems*, pages 1648–1656, 2015.

[29] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–993, 2018.

[30] Samuel Kadoury. Manifold learning in medical imaging. In *Manifolds*. IntechOpen, 2018.

[31] Chang Ha Lee, Amitabh Varshney, and David W Jacobs. Mesh saliency. *ACM transactions on graphics (TOG)*, 24(3):659–666, 2005.

[32] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *International Conference on Learning Representations(ICLR)*, 2018.

[33] Haisheng Li, Li Sun, Xiaoqun Wu, and Qiang Cai. Scale-invariant wave kernel signature for non-rigid 3d shape retrieval. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 448–454. IEEE, 2018.

[34] Dawen Liang and John Paisley. Landmarking manifolds with Gaussian processes. In *International Conference on Machine Learning*, pages 466–474, 2015.

[35] Lizhen Lin, Niu Mu, Pokman Cheung, David Dunson, et al. Extrinsic gaussian processes for regression and classification on manifolds. *Bayesian Analysis*, 2018.

[36] Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

[37] Anton Mallasto and Aasa Feragen. Wrapped gaussian process regression on riemannian manifolds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5580–5588, 2018.

[38] Radford M Neal. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pages 29–53. Springer, 1996.

[39] Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[40] Gabriel Parra and Felipe Tobar. Spectral mixture kernels for multi-output gaussian processes. In *Advances in Neural Information Processing Systems*, pages 6681–6690, 2017.

[41] David Pickup, X Sun, Paul L Rosin, RR Martin, Z Cheng, Z Lian, M Aono, A Ben Hamza, A Bronstein, M Bronstein, et al. Shrec'14 track: Shape retrieval of non-rigid 3d human models. In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval*, volume 1, page 6. Eurographics Association, 2014.

[42] Victor Adrian Prisacariu and Ian Reid. Nonlinear shape manifolds as shape priors in level set segmentation and tracking. In *CVPR 2011*, pages 2185–2192. IEEE, 2011.

[43] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[44] Manazhy Rashmi and Praveen Sankaran. Optimal landmark point selection using clustering for manifold modeling and data classification. *Journal of Classification*, pages 1–19, 2019.

[45] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

[46] Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4588–4599, 2017.

[47] Simo Särkkä. Linear operators and stochastic partial differential equations in gaussian process regression. In *International Conference on Artificial Neural Networks*, pages 151–158. Springer, 2011.

[48] Michael Sherman. *Spatial statistics and spatio-temporal data: covariance functions and directional properties*. John Wiley & Sons, 2011.

[49] Rishit Sheth, Yuyang Wang, and Roni Khardon. Sparse variational inference for generalized gp models. In *International Conference on Machine Learning*, pages 1302–1311, 2015.

[50] Kaleem Siddiqi, Juan Zhang, Diego Macrini, Ali Shokoufandeh, Sylvain Bouix, and Sven Dickinson. Retrieving articulated 3-d models using medial surfaces. *Machine vision and applications*, 19(4):261–275, 2008.

[51] Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank gaussian process regression. *Statistics and Computing*, 30(2):419–446, 2020.

[52] Michael L Stein. A kernel approximation to the kriging predictor of a spatial process. *Annals of the Institute of Statistical Mathematics*, 43(1):61–75, 1991.

[53] Michael L Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.

[54] Walter A Strauss. *Partielle Differentialgleichungen: eine Einführung*. Springer-Verlag, 2013.

[55] Felipe Tobar, Thang D Bui, and Richard E Turner. Learning stationary time series using gaussian processes with nonparametric kernels. In *Advances in Neural Information Processing Systems*, pages 3501–3509, 2015.

[56] Mark Van der Wilk, Carl Edward Rasmussen, and James Hensman. Convolutional gaussian processes. In *Advances in Neural Information Processing Systems*, pages 2849–2858, 2017.

[57] Ian Walker and Ben Glocker. Graph convolutional gaussian processes. *arXiv preprint arXiv:1905.05739*, 2019.

[58] Yue Wang and Yang Song. Facial keypoints detection, 2014.

[59] Andrew Gordon Wilson. *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

[60] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378, 2016.

[61] James V Zidek, Constance van Eeden, et al. Uncertainty, entropy, variance and the effect of partial information. *Lecture Notes-Monograph Series*, 42:155–167, 2003.