Journal of Machine Learning Research 24 (2023) 1-42 Submitted 6/22; Revised 1/23; Published 3/23

Intrinsic Gaussian Process on Unknown Manifolds with Probabilistic Metrics

Mu Niu MU.NIU@GLASGOW.AC.UK
School of Mathematics and Statistics
University of Glasgow, UK

Zhenwen Dai ZHENWEND@SPOTIFY.COM
Spotify, London, UK

Pokman Cheung POKMAN@ALUMNI.STANFORD.EDU
London, UK

Yizhu Wang 2603214W@STUDENT.GLA.AC.UK
School of Mathematics and Statistics
University of Glasgow, UK

Editor: Mingyuan Zhou

Abstract

This article presents a novel approach to construct Intrinsic Gaussian Processes for regression on unknown manifolds with probabilistic metrics (GPUM) in point clouds. In many real world applications, one often encounters high dimensional data (e.g.'point cloud data') centered around some lower dimensional unknown manifolds. The geometry of manifold is in general different from the usual Euclidean geometry. Naively applying traditional smoothing methods such as Euclidean Gaussian Processes (GPs) to manifold-valued data and so ignoring the geometry of the space can potentially lead to highly misleading predictions and inferences. A manifold embedded in a high dimensional Euclidean space can be well described by a probabilistic mapping function and the corresponding latent space. We investigate the geometrical structure of the unknown manifolds using the Bayesian Gaussian Processes latent variable models(B-GPLVM) and Riemannian geometry. The distribution of the metric tensor is learned using B-GPLVM. The boundary of the resulting manifold is defined based on the uncertainty quantification of the mapping. We use the probabilistic metric tensor to simulate Brownian Motion paths on the unknown manifold. The heat kernel is estimated as the transition density of Brownian Motion and used as the covariance functions of GPUM . The applications of GPUM are illustrated in the simulation studies on the Swiss roll, high dimensional real datasets of WiFi signals and image data examples. Its performance is compared with the Graph Laplacian GP, Graph Matérn GP and Euclidean GP.

Keywords: Implicit manifold, Gaussian Process, Heat kernel, Brownian motion, Probabilistic generative model

1. Introduction

Gaussian Processes (GPs) have been widely used as data efficient modelling approaches that produce good uncertainty estimates. They also power many decision making approaches such as Bayesian optimisation, multi-armed bandits and experiment design. Characteris-

tics of the function prior such as differentiability, periodicity and symmetry can be easily controlled by constructing a specific covariance function. Most widely used GP covariance functions are defined over Euclidean space. However, in real world applications, data often lies on a manifold within the original space. Predictions are only valid on the manifold and should only be extrapolated from observations along the manifold. For example, traffic flows can only be measured over networks of roads, surface tension is only measured on the surface of a specific object and the joints of a robot arm can only be moved safely within a manifold of the joint space of all the joints. The traffic connectivity can also easily differ from road network due to road maintenance or traffic congestion. The inference of the manifold based on data brings the accuracy and flexibility. Directly applying a GP with a covariance function defined over Euclidean space would not be ideal as the extrapolation would not respect the geometry of the manifold.

Previously, several GP on manifold methods (Niu et al., 2019; Lin et al., 2019; Borovitskiy et al., 2020) have been proposed under the assumption that the geometry of the manifold is known. Niu et al. (2019) defined a heat kernel on a manifold and constructed an intrinsic Gaussian Process on the manifold. The intrinsic GP refers to a GP that employs the intrinsic Riemannian geometry of the manifold, including the boundary features and interior conditions. Lin et al. (2019) proposed extrinsic framework for GP modeling on manifolds, which relies on embedding of the manifold into a Euclidean space and then constructing extrinsic kernels for GPs on their images. Dunson et al. (2020) , Borovitskiy et al. (2021) and Bolin et al. (2022) focused on developing GPs on graphs and metric graphs formed by data observed on the manifold. Azangulov et al. (2022) developed stationary kernels and Gaussian processes on Lie Groups and their homogenous spaces.

In this work, we study GP modelling on manifolds of which the geometry is unknown. This is a more realistic setting for real world applications because measuring the exact geometry of data manifolds is often impossible or overly expensive. In particular, we focus on the scenario where only a sparse set of data points can be collected. As shown in our experiments, Graph Laplacian based methods perform poorly in this scenario due to the poor graph approximation. In contrast, we estimate the probabilistic parameterisation of the unknown manifolds using probabilistic latent variable models and propose a practical and general intrinsic GP on unknown manifolds (GPUM) methodology. This is the major novel contribution of the paper. Specifically, we investigate the geometrical structure of the unknown manifolds using Riemannian geometry. The distributions of the metric tensor and the boundaries of resulting manifolds are estimated using the Bayesian Gaussian process latent variable models (B-GPLVM). Brownian Motion(BM) sample paths on the unknown manifold are simulated using the probabilistic metric and respecting the boundaries. The covariance kernel on the unknown manifold is estimated by employing the equivalence relationship between the heat kernel and the transition density of BM on the unknown manifold. We prove this estimator is coordinate independent. Our method can incorporate the intrinsic geometry and the uncertainty of the unknown manifold for inference and respect the interior constraints. With numerical experiments on synthetic and real world datasets, we compared our method against Graph Laplacian based methods and GP regressions without manifold estimations. We demonstrated that our method outperforms all other methods on all of the datasets.

In the following sections, concepts of Riemannian geometry are introduced in Section 2. The metric learning algorithms are explained in Section 5. We prove the BM paths on manifolds simulated using different metrics have the same transition density in Section 6. The heat kernel estimates derived from the analytical metric and different metric learning methods such as Gaussian Processes Latent Variable Model (GPLVM) and B-GPLVM are compared in Section 6.2. Applications of GPUM on a synthetic dataset on Swiss roll, high dimensional real datasets of WiFi signals (Ferris et al., 2007) and COIL images (Nayar and Murase, 1996) are illustrated and compared to Graph Laplacian GPs (Dunson et al., 2020), Graph Matérn GPs (Borovitskiy et al., 2021) and Euclidean GPs in Section 7, 8 and 9.

2. Concepts of Riemannian Geometry and Theoretical Background

One way of representing a high dimensional dataset is to relate it to a lower dimensional set of latent variables through a set of mapping functions (potentially nonlinear). Tosi et al. (2014); Arvanitidis et al. (2019) investigated the geometrical structure of probabilistic generative dimensionality reduction models (or latent variable models) using Riemannian metrics and computed the geodesic distances on the manifolds learned from data. A manifold embedded in a high dimensional Euclidean space can be well described by a probabilistic mapping function and the corresponding latent space. If the dimension of the latent space is the same as the intrinsic dimension of the manifold, the latent space can be interpreted as the chart of the learned manifold. Intuitively, the chart provides a distorted view of the manifold. An illustration is shown in Fig.1(a). The x^1 and x^2 coordinates in the chart represent the radius and width of the Swiss roll in $\mathbb{R}^3$. The blue triangles in the chart can be mapped to the black points in the embedded space (Swiss roll in $\mathbb{R}^3$) through the mapping $\phi : R^q \to \mathbb{M} \subset R^p$. In other words, the Swiss roll can be parameterised by the chart. Measurements on the manifold can be computed in the chart locally, and integrated to provide global measures. This gives rise to the definition of a local inner product, known as a Riemannian metric tensor.

Let $\mathbb{M}$ be a q-dimensional complete and compact Riemannian manifold with the Riemannian metric $\mathbf{g}$, and $\partial\mathbb{M}$ its boundary. The Riemannian metric $\mathbf{g}$ can be represented as a symmetric, positive definite matrix-valued function, which defines a smoothly varying inner product in the tangent space of $\mathbb{M}$. Let $\mathcal{J}$ denote the Jacobian of ϕ. We have

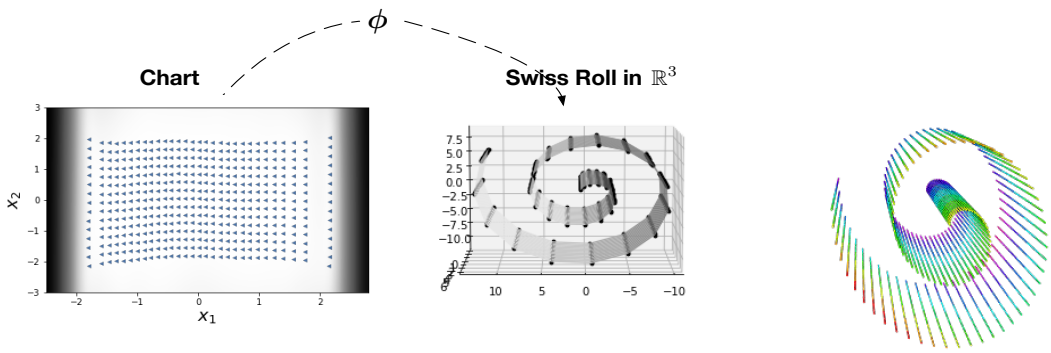
$$\mathbf{g} = \mathcal{J}^T \mathcal{J}, \quad \mathcal{J}_{i,j} = \frac{\partial \phi^i}{\partial x^j}. \tag{1}$$

The superscript indicates the j_{th} dimension of the chart and the i_{th} dimension of the observation space.

Moreover, based on its metric tensor, $\mathbb{M}$ has an associated Laplace-Beltrami operator, which is an intrinsically defined differential operator denoted Δ_s. In local coordinates, the Laplacian-Beltrami operator is

$$\Delta_s f = \frac{1}{\sqrt{G}} \frac{\partial}{\partial x^j} \left(\sqrt{G} \mathbf{g}^{ij} \frac{\partial f}{\partial x^i} \right), \tag{2}$$

where G is the determinant of the metric $\mathbf{g}$, $\mathbf{g}^{ij}$ is the (i, j) element of its inverse and f is a smooth function on $\mathbb{M}$. Take the special case where $\mathbb{M}$ is a Euclidean space $\mathbb{R}^q$, $\mathbf{g}$ becomes



(a) Chart of $\mathbb{M}$(Swiss roll) (b) Regression on $\mathbb{M}$

Figure 1: An illustration of a parameterisation of the manifold $\mathbb{M}$(Swiss roll) estimated from a data cloud in (a). ϕ maps the chart into $\mathbb{M} \subset \mathbb{R}^3$. The blue triangles in the chart are mapped to the black dots on $\mathbb{M} \subset \mathbb{R}^3$. The Riemannian geometry can help to learn the regression function in (b) (the color indicates function values) .

an identity matrix. The Laplace-Beltrami operator Δ_s becomes the Laplace operator Δ (the sum of second partial derivatives).

Consider the heat equation on $\mathbb{M}$, given by

$$\frac{\partial}{\partial t} K_{heat}(s_0, s, t) = \frac{1}{2} \Delta_s K_{heat}(s_0, s, t), \qquad s_0, s \in \mathbb{M},$$

where $s \in \mathbb{M}$, Δ_s is the Laplacian-Beltrami operator on $\mathbb{M}$, and $t \in \mathbb{R}^+$ is the diffusion time. A heat kernel of $\mathbb{M}$ is a smooth function $K(s_0, s, t)$ on $\mathbb{M} \times \mathbb{M} \times \mathbb{R}^+$ that satisfies the heat equation. It can be interpreted as the amount of heat that is transferred from s_0 to s in time t via diffusion.

The heat kernel satisfies the initial condition $\lim_{t \to 0} K_{heat}(s_0, s, t) = \delta(s_0, s)$ with δ the Dirac delta function. The heat kernel becomes unique when we impose a suitable condition along the boundary $\partial\mathbb{M}$, such as the Neumann boundary condition: $\partial K / \partial \mathbf{n} = 0$ along $\partial\mathbb{M}$ where $\mathbf{n}$ denotes a normal vector of $\partial\mathbb{M}$. The Neumann boundary condition can be understood as allowing no heat transfer across the boundary. If $\mathbb{M}$ is a Euclidean space $\mathbb{R}^q$, the heat kernel has a closed form corresponding to a time-varying Gaussian function:

$$K_{heat}(\mathbf{s}_0, \mathbf{s}, t) = \frac{1}{(2\pi t)^{q/2}} \exp\left\{ -\frac{||\mathbf{s}_0 - \mathbf{s}||^2}{2t} \right\}, \ \mathbf{s} \in \mathbb{R}^q.$$

The diffusion time t controls the rate of decay of the covariance. In the following, we will also write $K_{heat}(\mathbf{s}_0, \mathbf{s}, t)$ as $K^t_{heat}(\mathbf{s}_0, \mathbf{s})$.

For arbitrary Riemannian manifold, the construction of the heat kernel associated with the Laplace-Beltrami operator is not a trivial task and belongs to the fields of partial differential equations and differential geometry (Chavel, 1984). To circumvent solving the heat equation on manifolds, Niu et al. (2019) estimated the heat kernel as the BM transition density by simulating BM paths on a known manifold. In the case of point clouds, both the

metric tensor $\mathbf{g}$ and the boundary $\partial\mathbb{M}$ are unknown. In this work, we will use probabilistic latent variable models to learn the distribution of the metric $\mathbf{g}$ and define the boundary $\partial\mathbb{M}$ based on the uncertainty quantification of the mapping. The Laplace-Beltrami operator in (2) is the infinitesimal generator of BM on manifolds (Hsu, 1988). The BM on a Riemannian manifold in a local coordinate system is given as a system of stochastic differential equations in the Itô form (Hsu, 1988, 2008):

$$dx_i(t) = \frac{1}{2}G^{-1/2}\sum_{j=1}^{q}\frac{\partial}{\partial x_j}\left(\mathbf{g}^{ij}G^{1/2}\right)dt + \left(\mathbf{g}^{-1/2}dB(t)\right)_i, \tag{3}$$

where $\mathbf{g}$ is the metric tensor of $\mathbb{M}$, G is the determinant of $\mathbf{g}$ and $B(t)$ represents an independent BM in the Euclidean space.

3. Intrinsic Gaussian Processes on unknown manifolds

In this work, we focus on the following model,

$$y_i = f(s_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2) \tag{4}$$

where $f : \mathbb{M} \to \mathbb{R}$ is an unknown regression function. $y_i \in \mathbb{R}$ is a response variable. $s_i = (s_i^1, \ldots, s_i^p) \in \mathbb{M} \subset \mathbb{R}^p$ is an observed predictor on a complete and compact Riemannian manifold $\mathbb{M}$ embedded in $\mathbb{R}^p$. $\mathbb{M}$ can be parameterised by a q dimensional local coordinate system (chart), and $q < p$. For example, let $\mathbb{M}$ be a unit sphere and we have a point i on $\mathbb{M}$. $s_i \in \mathbb{R}^3$ is a three dimensional vector representing the Cartesian coordinates of i. The two dimensional chart of $\mathbb{M}$ is the longitude and latitude coordinate system. The point i can also be represented as a two dimensional vector x_i in the chart. There exists a mapping function ϕ from the Cartesian coordinates to the spherical coordinates. However in the case of point clouds where s_i is observed in $\mathbb{R}^p$, the parameterisation ϕ of $\mathbb{M}$ is unknown and x_i becomes a latent variable. We propose to infer how the output y varies with the input s, including predicting y values at new locations not represented in the training set.

A GP prior can be assigned to f with a covariance function. The choice of covariance kernel has a fundamental impact on the results. Most common choices of covariance kernels such as the squared exponential kernel and Matérn kernel depend critically on the Euclidean distance between s_i and s_j and ignore the intrinsic geometry of $\mathbb{M}$. By contrast, the heat kernel depends only on the Riemmanian metric and the intrinsic geometry of $\mathbb{M}$. It provides a natural generalisation of the RBF kernel on manifolds. The heat kernel represents the diffusion of the heat on a Riemannian manifold. However it is analytically intractable to directly evaluate (Hsu, 1988). Niu et al. (2019) proposed a computational framework to estimate the heat kernel on the manifold in which the analytical parameterisation is known. However, data represented as point clouds is often high dimensional and concentrated around some unknown lower dimensional structures, Niu et al. (2019) is not applicable due to the lack of analytical manifold parameterisation. In this paper, we address this problem by using the probabilistic generative dimension reduction models to learn the geometry of the implicit manifold and define the boundary. We propose to construct GPUM by using the heat kernel of the implicit manifold as the covariance kernel. With the help of Riemannian geometry, the Brownian motion sample paths can be simulated on $\mathbb{M}$ and the

heat kernel can be estimated as the transition density of the BM. With this construction we can learn the regression function on $\mathbb{M}$ as in Fig.1(b).

Let $\mathcal{D} = \{(s_i, y_i), i = 1, ..., n\}$ be the data, with $n \geq 1$ the number of labeled observations, $s_i \in \mathbb{M} \subset \mathbb{R}^p$ is the predictor and $y_i \in \mathbb{R}$ is the corresponding response. Suppose we are also given an unlabeled dataset $\mathcal{V} = \{s_i, i = n + 1..., n + v\}$ where $v \geq 1$. Consider the regression model in (4), we would like to make inferences about f with the labeled dataset $\mathcal{D}$ and predict y values for the unlabeled dataset $\mathcal{V}$. Under an GPUM prior for the unknown regression function as $f \sim GP(0, K^t_{heat}(.,.))$, we have

$$p(\mathbf{f}|s_1, s_2, ..., s_n) \sim \mathcal{N}(0, \Sigma_{\mathbf{ff}}),$$

where $\mathbf{f}$ is the discretization of f over $s_1, s_2, \cdot \cdot \cdot, s_n$ so that $f_i = f(s_i)$. $\Sigma_{\mathbf{ff}} \in \mathbb{R}^{n \times n}$ is the covariance matrix induced from the heat kernel. The (i, j) entry of $\Sigma_{\mathbf{ff}}$ corresponds to $\Sigma_{\mathbf{ff}_{i,j}} = \sigma^2_h K^t_{heat}(s_i, s_j)$. We introduce the rescaling hyperparameter σ^2_h to add extra flexibility to the heat kernel.

4. Related work

One of the paramount challenges in developing GP models on manifolds is the difficulty in specifying the covariance structure via constructing valid covariance kernels on manifolds. One might hope to achieve this by replacing Euclidean norms in the squared exponential kernel with geodesic distances. However Feragen et al. (2015) proved this is not generally a well-defined kernel. Lin et al. (2019) proposed extrinsic Gaussian Processes on manifolds by embedding the manifolds onto a higher dimensional Euclidean space. The squared exponential kernel is applied on the images of the manifold after embedding. However, such embeddings are not easy to obtain and only available for certain manifolds when the geometry is known. Yang and Dunson (2016) proposed a model bypassing the need to estimate the manifold, and can be implemented using standard algorithms for posterior computation in GPs. They show that by imposing a GP prior on the regression function with a covariance kernel defined directly on the ambient space (the embedding of the manifold in a high dimensional Euclidean space), the posterior distribution yields a posterior contraction rate depending on the intrinsic dimension of the manifold. They assume that the unknown lower dimensional space where the predictors center around are a class of submanifolds of Euclidean space. They focus on compact manifolds without boundary.

Niu et al. (2019) proposed to use heat kernels to construct the intrinsic Gaussian Processes on complex constrained domains. Heat kernel can be interpreted as the transition density of Brownian Motion on the manifold (Hsu, 1988). It is estimated by simulating BM paths on manifolds in which the analytical parameterisation is known in Niu et al. (2019). Alternatively if the eigen-paires of the Laplacian-Beltrami operator of the manifold are available, Borovitskiy et al. (2020) approximated the heat kernel with the sum of finite-many eigen-paires of the Laplacian-Beltrami. Both Borovitskiy et al. (2020) and Niu et al. (2019) are only applicable when the geometry of the manifold is known and the dimensions of the observation(or embedding) space are low such as $\mathbb{M} \subset \mathbb{R}^2$ or $\mathbb{M} \subset \mathbb{R}^3$ ($p = 2$ or 3).

Most recent research in Dunson et al. (2020) tackled this problem by approximating the heat kernel kernel of a compact Riemannian manifold with finitely-many eigenpairs of the Graph Laplacian using the labeled and unlabeled predictor values. We refer the Gaussian

processes constructed by this approximation as the Graph Laplacian Gaussian processes (GL-GPs). Let Δ be the Laplace-Beltrami operator of a manifold and $\{\lambda_i\}_{i=0}^{\infty}$ be the spectrum or eigenvalues of $-\Delta$. Denote φ_i the corresponding eigenfunction, for each $i \in N$, we have $\Delta\varphi_i = -\lambda_i\varphi_i$. If the eigen-decomposition of Δ is known, the corresponding heat kernel of the manifold has the following expression: $K_{heat}(s, s', t) = \sum_{i=0}^{\infty} e^{-\lambda_i t}\varphi_i(s)\varphi_i(s')$. If the geometry of the manifold and the corresponding Δ are unknown, Δ is approximated by the Graph Laplacian matrix $\boldsymbol{L}$ in Dunson et al. (2020). $\boldsymbol{L}$ is constructed from a point cloud whose adjacency matrix is computed by using a Gaussian function with pair-wised Euclidean distance. The heat kernel is approximated as the summation of finite eigenpairs of $\boldsymbol{L}$, $\sum_{i=0}^{n_G} e^{-\mu_i t} v_i v_i^T$, where μ_i and v_i are the i_{th} normalised eigenvalue and eigenvector of $\boldsymbol{L}$. n_G is the number of eigen pairs. The implementation of the GL-GP is provided in Appendix I.

Borovitskiy et al. (2021) leveraged the stochastic partial differential equation characterization of Matérn kernel to study their analog for undirected graphs and developed the Graph Matérn Gaussian processes (GM-GPs). The Graph Matérn kernel is constructed with the sum of finitely-many eigenpairs of the normalised Graph Laplacian $\boldsymbol{L}$. $\boldsymbol{L}$ is computed from the adjacency matrix of a predefined graph. We implemented the GM-GP by following the instructions in the Github repository in Borovitskiy et al. (2021). The key component of GL-GPs and GM-GPs is the Graph Laplacian. If the true graph connections are not known, the graph constructed based on local distances such as Delaunay triangulation can be error-prone when observations are sparse (Hjelle and Dæhlen, 2006). The graph based methods often result in poor approximation of the manifold when the number of the observations are low. GL-GPs and GM-GPs are applied to the simulation studies and real datasets and compared to GPUM in the later sections.

There are still some critical gaps in current practices. In particular, the lack of robust methods for carrying out intrinsic statistical inference and effective models for regressions with manifold-valued data embedded in a point cloud. In this work we focus on learning the manifold structure using probabilistic dimension reduction methods such as Bayesian GPLVM and constructing the GPUM by using the heat kernel of the learned manifold. Other related methods such as Auto-encoders (Kramer, 1991) and Variational Auto-encoders (VAE)(Kingma et al., 2019), can also be considered in this two stage approach. Auto-encoders provide a neural network based framework for learning deep latent variable models. However, the resulting mapping function is deterministic and the model does not have a built-in quantification of its uncertainty. There is lack of uncertainty quantification in the learned manifold. Alternatively, VAE address this concern directly through an explicit likelihood model and a variational approximation of the representation posterior. It learns a generative model by specifying a likelihood of observations conditioned on latent variables and a prior over the latent variables. Both the likelihood and the variational distributions have parameters predicted by neural networks that act similarly to the encoder–decoder pair of the classic autoencoder.

5. Learning metrics and boundaries

Let $\mathcal{S} = \{s_i | i = 1, \cdots, n + v\}$, $s_i \in \mathbb{R}^p$, include all predictors in the labeled dataset $\mathcal{D}$ and unlabeled dataset $\mathcal{V}$. We have $\mathcal{S} \in \mathbb{R}^{(n+v)\times p}$. Suppose we perform probabilistic nonlinear

dimensionality reduction by defining a latent variable model that introduces a set of latent (unobserved) variables $\mathcal{X} = \{x_i | i = 1, \cdots, n+v\}$, $x_i \in \mathbb{R}^q$, $\mathcal{X} \in \mathbb{R}^{(n+v)\times q}$ and $q < p$. $\mathcal{X}$ is related to $\mathcal{S}$ which is observed in a higher dimensional space. A prior distribution is placed on the latent space which induces a distribution over $\mathcal{S}$ under the assumption of the probabilistic mapping

$$s_i^j = \phi^j(x_i) + e_i^j, \tag{5}$$

where $x_i \in \mathbb{R}^q$ is the latent point associated with the i_{th} observation $s_i \in \mathbb{R}^p$ in the original observation space and $i \leq n$. j is the index of the features (dimensions) of s in the observation space and $j \leq p$. e_i^j is a Gaussian distributed noise term, $e_i^j \sim \mathcal{N}(0, \beta^2)$. An illustration is given in Fig. 1(a) where $q = 2$ and $p = 3$. If the mapping ϕ is linear and the prior $p(\mathcal{X})$ is Gaussian, this model is known as probabilistic principal component analysis (Tipping and Bishop, 1998). In this work, we do not restrict to this linear assumption and consider some nonlinear dimensionality reduction methods such as GPLVM and B-GPLVM.

When the ϕ function in Fig.1(a) is differentiable, it can be interpreted as the mapping between the latent space and the manifold $\mathbb{M}$. If the dimensions of $\mathbb{M}$ are known, by letting q equal the dimensions of $\mathbb{M}$, the latent space can be interpreted as the chart of $\mathbb{M}$. If the dimensions of $\mathbb{M}$ are unknown, q is estimated by using the so called Automatic Relevance Determination (ARD) in GPLVM and B-GPLVM (Zwiessele, 2017). ARD allows assigning scaling parameters for each dimension. These scaling parameters can be incorporated to kernels such as the RBF kernel as the inverse of the squared lengthscales.

In general a manifold may need more than one chart to be parameterised. In this work, we focus on examples of single chart. A rigorous study of a general multi-chart manifold problem is very challenging and beyond the scope of the current paper. In principle our method can be extended to multiple charts manifolds. However, there are some challenges such as systematically defining the number of the charts required to parameterise the manifold, learning the individual charts and estimating the corresponding kernel. The Riemannian metric of the given model can be computed as in (1). In the case of probabilistic LVMs, we place a Gaussian prior over the mapping function $\phi(x)|x$. The conditional probability over the Jacobian also follows a Gaussian distribution, this naturally induces a distribution over the metric tensor $\mathbf{g}$. We denote the distribution of the Jacobian as $\mathbf{J}$ given the set $\mathcal{S}$ and the mapping ϕ in (5). Assuming independent rows of $\mathbf{J}$ (Lawrence, 2005; Titsias and Lawrence, 2010),

$$p(\mathbf{J}|\mathcal{S}, \Phi) = \prod_{j=1}^{p} \mathcal{N}(\mu_{\mathbf{J}}^j, \Sigma_{\mathbf{J}}). \tag{6}$$

This independent row assumption is for the dimensions in the original observational space. The dimensions in the learned latent space are not independent. This assumption can be relaxed by using a multi-output GP which is more compuationaly expensive (Alvarez and Lawrence, 2011). The expressions of the mean $\mu_{\mathbf{J}}$ and variance $\Sigma_{\mathbf{J}}$ of the Jacobian are model specific and given in section 5.1 and 5.2. The resulting metric $\mathbf{g}$ follows a non-central Wishart distribution (Anderson, 1946)

$$\mathbf{g} \sim \mathcal{W}_q\left(p, \Sigma_{\mathbf{J}}, \mathbb{E}(\mathbf{J}^T)\mathbb{E}(\mathbf{J})\right). \tag{7}$$

From this distribution, the expected metric tensor can be computed as

$$\mathcal{G} = \mathbb{E}(\mathbf{g}) = \mathbb{E}(\mathbf{J}^T)\mathbb{E}(\mathbf{J}) + p\Sigma_{\mathbf{J}}. \tag{8}$$

We denote the expectation by $\mathcal{G}$. Note that the variance term $\Sigma_{\mathbf{J}}$ is included in $\mathcal{G}$. It implies that the metric tensor expands as the uncertainty over the mapping increases. Hence the BM simulation steps in the SDE in (21) will travel 'slower' in the region of the latent space where the uncertainty is high.

We also need the gradient of the expected metric to simulate BM as in section 6.1.

$$\frac{\partial \mathcal{G}}{\partial x^l} = \frac{\partial \mathbb{E}[\mathbf{g}]}{\partial x^l} = \frac{\partial \mathbb{E}[\mathbf{J}^T]}{\partial x^l}\mathbb{E}[\mathbf{J}] + \mathbb{E}[\mathbf{J}^T]\frac{\partial \mathbb{E}[\mathbf{J}]}{\partial x^l} + p\frac{\partial \Sigma_{\mathbf{J}}}{\partial x^l} \tag{9}$$

The estimates of the mapping $\phi(x)$ become highly unreliable when x is far from the data points. The corresponding metric tensor estimates and BM simulations can also be ill defined and violate the manifold geometry. To avoid these poorly estimated region due to lack of data, the boundary of the learned manifold can be defined by $\mathbb{V}ar(\phi(x)|x)$, the variance of the mapping at x. $\mathbb{V}ar(\phi(x)|x)$ is also a smooth function in B-GPLVM. In most cases, the $\partial\mathbb{M}$ defined here is also a $q-1$ smooth manifold. Any x outside of the boundary has $\mathbb{V}ar(\phi(x)|x) > \alpha$.

$$\partial\mathbb{M} = \{x \in \mathbb{R}^q \mid \mathbb{V}ar(\phi(x)|x) = \alpha\}. \tag{10}$$

To define the value of α, we first compute the maximum distance $\delta_{\mathcal{X}}$ between two neighbouring data points in the latent space. We then sample the 'shifted latent points' h from a set $\mathbb{H} = \{h | \|h - x_i\| = \delta_{\mathcal{X}}, x_i \in \mathcal{X}\}$, which contains all points in the latent space whose distances to the data points are $\delta_{\mathcal{X}}$. The maximum variance of $\phi(h)$ is used for the value of α. In practice, we create h by moving all data points with $\delta_{\mathcal{X}}$ in random directions and let α equal to the maximum variance of the mapping at these relocated points. The samples are chosen to make sure that $\alpha > \max\left(\mathbb{V}ar\left(\phi(x_i)|x_i\right)\right)$, for all $x_i \in \mathcal{X}$. An ablation study of the choice of α based on the Swiss roll experiment is given in Appendix D.

5.1 GPLVM metric

Gaussian Process Latent Variables Model (GPLVM; Lawrence, 2005) is a nonlinear probabilistic generative model. In this section, we will derive the distribution of the metric from GPLVM. A sample from GPLVM defines a generative mapping from $x \in \mathbb{R}^q$ in the latent space to $s \in \mathbb{M} \subset \mathbb{R}^p$ in the observation space. GPs define a prior over the mapping ϕ in (5). A zero mean prior is used as a default choice. If the domain knowledge of where the prior should be centred is available, such as the embedding of $\mathbb{R}^q$ into $\mathbb{R}^p$, it can also be encoded into the mean function. The individual dimensions of the p dimensional observation space are modeled independently in the GP prior sharing the same hyperparameters. Given the construction outlined above, the probability of the observed data $\mathcal{S} = \{s_i^j | i \in \{1, \cdots, n+v\}, j \in \{1, \cdots, p\}\}$ conditioned on all latent variables $\mathcal{X} = \{x_i^j | i \in \{1, \cdots, n+v\}, j \in \{1, \cdots, q\}\}$ is written as follows:

$$p(\mathcal{S}, \Phi|\mathcal{X}, \beta) = p(\mathcal{S}|\Phi, \beta)p(\Phi|\mathcal{X}) = \prod_{j=1}^{p} p(\mathbf{s}_:^j|\boldsymbol{\phi}_:^j, \beta)p(\boldsymbol{\phi}_:^j|\mathcal{X}), \tag{11}$$

where $\Phi = \{\boldsymbol{\phi}_i^j | i \in \{1, \cdots, n+v\}, j \in \{1, \cdots, p\}\}$, $\Phi \in \mathbb{R}^{(n+v)\times p}$ and $\boldsymbol{\phi}_i^j = \phi(x_i)^j$. $\mathbf{s}_:^j$ represents the j_{th} dimension of all points in $\mathcal{S}$ and $\mathcal{S} \in \mathbb{R}^{(n+v)\times p}$. The likelihood $p(\mathcal{S}|\mathcal{X})$ is computed by marginalising out Φ and optimising the latent variables $\mathcal{X}$

$$p(\mathcal{S}|\mathcal{X}) = \prod_{j=1}^{p} \mathcal{N}(\mathbf{s}_:^j, K_{\mathcal{X}\mathcal{X}} + \beta^2 I). \tag{12}$$

$K_{\mathcal{X}\mathcal{X}}$ is the $(n+v)\times(n+v)$ covariance matrix defined by the squared exponential kernel. Lawrence (2005) estimated all the latent variables $\mathcal{X}$ and the kernel hyper-parameters of GPLVM with the maximum likelihood estimate. If the covariance kernel is differentiable, the derivative of a GP is again a GP (Rasmussen and Williams, 2006; Adler, 2010). This property allows us to compute the derivative of GP. The Jacobian $\mathbf{J}$ of the GPLVM mapping can be computed as the partial derivative $\frac{\partial \phi_*}{\partial x^l}$ with respect to the l_{th} dimension for any point x_* in the latent space,

$$\mathbf{J}^T = \frac{\partial \phi_*}{\partial x} = \begin{bmatrix} \frac{\partial \phi(x_*)^1}{\partial x^1} & \cdots & \frac{\partial \phi(x_*)^j}{\partial x^1} & \cdots & \frac{\partial \phi(x_*)^p}{\partial x^1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial \phi(x_*)^1}{\partial x^q} & \cdots & \frac{\partial \phi(x_*)^j}{\partial x^q} & \cdots & \frac{\partial \phi(x_*)^p}{\partial x^q} \end{bmatrix}, \tag{13}$$

where $\frac{\partial \phi_*}{\partial x}$ is a $q \times p$ matrix. Considering the independence across the dimensions of the observation space, the joint distribution of the j_{th} dimension of the mapping ϕ and the j_{th} column of the Jacobian can be written as

$$\begin{bmatrix} \phi(\mathcal{X})^j \\ \frac{\partial \phi(x_*)^j}{\partial x} \end{bmatrix}, \sim \mathcal{N}\left(0, \begin{bmatrix} K_{\mathcal{X},\mathcal{X}} & \partial K_{\mathcal{X},*} \\ \partial K_{\mathcal{X},*}^T & \partial^2 K_{*,*} \end{bmatrix}\right). \tag{14}$$

The expressions of $K_{\mathcal{X},\mathcal{X}}$, $\partial K_{\mathcal{X},*}$, and $\partial^2 K_{*,*}$ are given in Appendix B. GPLVM provides an explicit mapping from the latent space to the observation space. This mapping defines the support of the observed data $\mathcal{S}$ as a q-dimensional manifold embedded in $\mathbb{R}^p$. The distribution of the Jacobian of GPLVM is the product of p independent Gaussian distributions (one for each dimension of the observation space). For a point x_* in the latent space, the distribution of the Jacobian takes the form

$$\begin{aligned} p(\mathbf{J}|\mathcal{X}, \mathcal{S}) &= \prod_{j=1}^{p} \mathcal{N}(\mu_{\mathbf{J}}^j, \Sigma_{\mathbf{J}}) \\ &= \prod_{j=1}^{p} \mathcal{N}(\partial K_{\mathcal{X},*}^T K_{\mathcal{X},\mathcal{X}}^{-1} \mathbf{s}_:^j \ , \partial^2 K_{*,*} - \partial K_{\mathcal{X},*}^T K_{\mathcal{X},\mathcal{X}}^{-1} \partial K_{\mathcal{X},*}). \end{aligned} \tag{15}$$

From this distribution, the expected metric tensor can be computed as in (8). The boundary $\partial\mathbb{M}$ is defined by $\mathbb{V}ar(\phi(x_*)|x_*)$, the variance of mapping in (10). For any x_* in the latent space

$$\mathbb{V}ar(\phi(x_*)|x_*) = K_{*,*} - K_{*,\mathcal{X}} K_{\mathcal{X},\mathcal{X}}^{-1} K_{\mathcal{X},*}. \tag{16}$$

5.2 Bayesian GPLVM metric

The maximum likelihood estimation of the latent inputs $\mathcal{X}$ in GPLVM often leads to overfitting due to its high dimensionality (Damianou et al., 2016). This overfitting can be avoided by applying a Bayesian treatment to the latent inputs. By introducing a prior distribution to the latent inputs, the marginal likelihood takes the form

$$p(\mathcal{S}) = \int p(\mathcal{S} \mid \mathcal{X}) p(\mathcal{X}) d\mathcal{X}.$$

The integral is intractable as the inputs $\mathcal{X}$ to the latent variable model go through a nonlinear calculation in the inverse of the covariance matrix. Bayesian Gaussian Process Latent Variable Model (B-GPLVM; Titsias and Lawrence, 2010) introduces a variational inference framework for training the latent variable model. It variationally integrates out the input variables and computes a lower bound on the exact marginal likelihood of the nonlinear latent variable model. The maximization of the variational lower bound provides a Bayesian training procedure that is robust to overfitting.

The key to the tractable variational Bayes approach is the application of variational inference to an augmented GP formulation, known as sparse GP, where the GP prior on ϕ is augmented to include auxiliary variables. More specifically, we expand the conditional probabilistic model in (11) by including m extra samples (inducing points) to the GP latent mapping e.g. $u_i = \phi(x_{ui}) \in \mathbb{R}^p$ is such a sample (Lawrence, 2007). These inducing points are denoted by $\mathcal{U} = \{u_i^j | i \in \{1, \cdots, m\}, j \in \{1, \cdots, p\}\}$, $\mathcal{U} \in \mathbb{R}^{m \times p}$ and constitute latent function evaluations at a set of pseudo-inputs $\mathcal{X}_u = \{x_{ui}^j | i \in \{1, \ldots, m\}, j \in \{1, \ldots, q\}\}$, $\mathcal{X}_u \in \mathbb{R}^{m \times q}$. The inducing inputs $\mathcal{X}_u$ are variational parameters. Φ is defined as in (11). The augmented joint probability and the marginal likelihood can be written as (17)

$$p(\mathcal{S}, \Phi, \mathcal{U}, \mathcal{X} | \beta) = p(\mathcal{S}|\Phi) p(\Phi|\mathcal{U}, \mathcal{X}) p(\mathcal{U}) p(\mathcal{X}) = \prod_{j=1}^{p} p(\mathbf{s}_:^j | \boldsymbol{\phi}_:^j) p(\boldsymbol{\phi}_:^j | \mathbf{u}_:^j, \mathcal{X}) p(\mathbf{u}_:^j) p(\mathcal{X}), \quad (17)$$

$$p(\mathcal{S}) = \int \int \int p(\mathcal{S}, \Phi, \mathcal{U}, \mathcal{X}) \; d\mathcal{U} \; d\Phi \; d\mathcal{X}. \quad (18)$$

With variational inference, $\log(p(\mathcal{S}))$ can be lower bounded by applying Jensen's inequality (Damianou et al., 2016). The resulting lower bound can be computed analytically. The full technical details of the lower bound are given in (43) in Appendix C. The Jacobian has the same shape as in (13). For a point x_* in the latent space, the distribution of the Jacobian for B-GPLVM takes the form:

$$p(\mathbf{J}|\mathcal{X}, \mathcal{U}, \mathcal{S}) = \prod_{j=1}^{p} \mathcal{N}(\; \mu_{\mathbf{J}}^j, \Sigma_{\mathbf{J}}) \quad (19)$$

$$= \prod_{j=1}^{p} \mathcal{N}(\; \partial K_{\mathcal{X}_u,*}^T K_{\mathcal{X}_u,\mathcal{X}_u}^{-1} \mu_{qu}^j, \; \partial^2 K_{*,*} - \partial K_{\mathcal{X}_u,*}^T \Lambda \partial K_{\mathcal{X}_u,*})$$

$$\Lambda = K_{\mathcal{X}_u\mathcal{X}_u}^{-1} - K_{\mathcal{X}_u\mathcal{X}_u}^{-1} \Sigma_{qu} K_{\mathcal{X}_u\mathcal{X}_u}^{-1}$$

$$\mathbb{V}ar(\phi(x_*)|x_*) = K_{*,*} - K_{*,\mathcal{X}_u} \Lambda K_{\mathcal{X}_u,*} \quad (20)$$

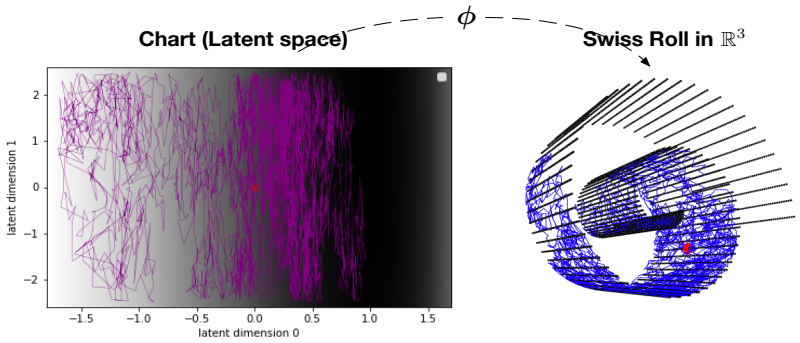


Figure 2: A BM sample path (blue line, right panel) on $\mathbb{M}$ (Swiss roll in $\mathbb{R}^3$) and its equivalent stochastic process (purple line, left panel) in the chart (or latent space) in $\mathbb{R}^2$. $\phi : \mathbb{R}^2 \to \mathbb{M} \subset \mathbb{R}^3$ is a parametrization of $\mathbb{M}$. The purple line in the chart is simulated using (21). Its mapping in $\mathbb{R}^3$ is the blue line on the Swiss roll. The red dot is the starting location of the BM trajectory. The horizontal axis of the latent space represents the radius of Swiss roll. The vertical axis is for the width. The gray color in the latent space indicates the magnification factor. When the latent space is mapped to $\mathbb{M} \subset \mathbb{R}^3$, the darker region will be stretched more. The latent space is learned using a dataset of 450 points in section 7.

where μ_{qu}^j and Σ_{qu} are the mean and variance of the variational distribution of the inducing points $\mathcal{U}$. The expressions of μ_{qu} and Σ_{qu} are given in (44) in Appendix C. The expectation of the metric tensor can be computed as in (8). The boundary $\partial\mathbb{M}$ of the implicit manifold can be defined by computing the variance of the mapping as in (20).

6. Estimate heat kernel as BM transition density

In this section, we will estimate the BM transition density by simulating BM sample paths on the implicit manifold $\mathbb{M}$.

6.1 Simulating Brownian motion on implicit manifolds

From section 5, we learned the probabilistic parameterisation ϕ of $\mathbb{M}$, the distribution of the associated metric tensor $\mathbf{g}$ and the boundary $\partial\mathbb{M}$. In order to estimate the BM transition density on $\mathbb{M}$, we first need to simulate BM trajectories. Simulating the sample paths of BM on $\mathbb{M} \subset \mathbb{R}^p$ is equivalent to simulating the stochastic processes in the latent space (or chart) in $\mathbb{R}^q$, $q < p$. An illustrated example is shown in Fig. 2.

BM on a Riemannian manifold in a local coordinate system is given as a system of stochastic differential equations (SDE) in Itô form (Hsu, 1988). We use the expected metric $\mathcal{G} = \mathbb{E}[\mathbf{g}]$ to construct the SDEs

$$dx^i(t) = \frac{1}{2} G^{-1/2} \sum_{j=1}^{q} \frac{\partial}{\partial x^j} \left(\mathcal{G}^{ij} G^{1/2} \right) dt + \left(\mathcal{G}^{-1/2} dB(t) \right)_i \tag{21}$$

where x^i represents the i_{th} dimension of the latent space (chart). $\mathcal{G}$ is defined in eqn (8), G is the determinant of $\mathcal{G}$ and $B(t)$ represents an independent BM in Euclidean space. The

discrete form of (21) is derived in (22).

$$x^i(t) = x^i(t-1) + \frac{1}{2}\sum_{j=1}^{q}\left(-\mathcal{G}^{-1}\frac{\partial \mathcal{G}}{\partial x^j}\mathcal{G}^{-1}\right)_{ij}\Delta t + \frac{1}{4}\sum_{j=1}^{q}(\mathcal{G}^{-1})_{ij}tr\left(\mathcal{G}^{-1}\frac{\partial \mathcal{G}}{\partial x^j}\right)\Delta t + \left(\mathcal{G}^{-1/2}dB(t)\right)_i$$
$$= \mu(x^i(t-1), \Delta t)_i + \left(\sqrt{\Delta t}\mathcal{G}^{-1/2}z^q\right)_i \tag{22}$$

where Δt is the diffusion time of the BM simulation step and z^q is a q-dimensional standard normal random variable. The discrete form of the SDE also defines the proposal mechanism of the BM

$$q\left(x(t)|x(t-1)\right) = \mathbb{N}\left(x(t)|\mu\left(x\left(t-1\right), \Delta t\right), \Delta t\mathcal{G}^{-1}\right). \tag{23}$$

Theorem 1 *The stochastic process defined in* (21) *is coordinate independent. With a given δt, simulations in any choice of local coordinates (or metric) as above are equivalent to the same step in $\mathbb{M}$.*

The proof of Theorem 1 is given in Appendix A. This implies that the BM sample paths simulated from different choices of metric $\mathcal{G}$ should have the same properties. The BM steps are sampled from the proposal distribution in (23), which is defined by the metric tensor. The boundary $\partial\mathbb{M}$ of the manifold is also quantified by the uncertainty of the mapping. We apply the Neumann boundary condition as in Section 3. As a result the simulated sample paths stay within the boundary. An example of BM trajectory on Swiss roll is given in Fig. 2. The latent space and the associated metrics are learned from B-GPLVM. The gray color in the latent space indicates the square root of the determinant of the metric. It is also known as the magnification factor (Bishop et al., 1997; Zwiessele, 2017; Tosi, 2014).

$$\mathcal{MF} = \sqrt{det(\mathcal{G})}$$

The geometric interpretation of the magnification factor is how much a small piece of the latent space in $\mathbb{R}^q$ will be stretched or compressed when it is mapped to $\mathbb{M} \subset \mathbb{R}^p$. For example in Fig.2 left panel, the horizontal axis of the latent space can be interpreted as the scaled radius of the Swiss roll, the vertical axis as the width of the Swiss roll. When the radius is bigger, the corresponding area in the latent space is darker and the magnification factor is larger. When the latent space is mapped back to the manifold, the darker region will be stretched more. The purple trajectory of the stochastic process in the latent space (Fig. 2 left panel) is denser in the darker area and more spread out in the bright area. The stochastic process travels with smaller steps when the magnification factor is large and vice versa. Note that, due to the BM being coordinate independent as in Theorem 1, the BM trajectory is evenly spread out in the manifold of the Swiss roll in $\mathbb{R}^3$ (see Fig. 2 right panel).

6.2 Estimate the transition density of BM

Considering the BM $\{\mathbf{S}(t)|t > 0\}$ on $\mathbb{M} \subset \mathbb{R}^p$. The BM starts from $\mathbf{S}(0) = s_0$ at time 0. We simulate N_{BM} sample paths. Given a point $s \in \mathbb{M}$, we define a small neighbourhood of s as

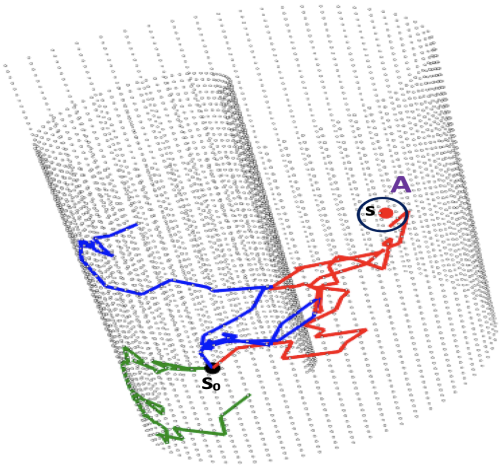


Figure 3: Three BM paths (red,blue,green) are simulated on the Swiss Roll, with the starting point at s_0 (black ball). s (red ball) is the target point. $\boldsymbol{A}$ is the neighbourhood of S. Only the red path reach $\boldsymbol{A}$ at time t. The transition probability $p\{S(t) \in \boldsymbol{A} | S(0) = s_0\}$ is 1/3.

$\mathbf{A}_s \in \mathbb{M}$. For any $t > 0$, the probability of $\mathbf{S}(t)$ reaching $\mathbf{A}_s$ at time t, $p(\mathbf{S}(t) \in \mathbf{A}_s | \mathbf{S}(0) = s_0)$, can be approximated by

$$p(\mathbf{S}(t) \in \mathbf{A}_s \mid \mathbf{S}(0) = s_0) \approx \frac{N_{A_s}}{N_{BM}} \tag{24}$$

where N_{A_s} is the number of sample paths reaching $\mathbf{A}_s$ at time t. An illustrative diagram is shown in Fig. 3. This BM transition probability is defined as the integral of the BM transition density over $\mathbf{A}_s$ which is also the heat kernel K^t_{heat}. Since we do not have the analytical expression of the transition probability, the transition density cannot be derived by taking the derivative of $p(\mathbf{S}(t) \in \mathbf{A}_s | \mathbf{S}(0) = s_0)$. Instead, K^t_{heat} can be numerically approximated as $\hat{K}^t_{heat}$

$$K^t_{heat}(s_0, s) \approx \frac{p(\mathbf{S}(t) \in \mathbf{A}_s \mid \mathbf{S}(0) = s_0)}{V(\mathbf{A}_s)} \approx \frac{1}{V(\mathbf{A}_s)} \cdot \frac{N_{A_s}}{N_{BM}} = \hat{K}^t_{heat} \tag{25}$$

where $V(\mathbf{A}_s)$ is the Riemannian volume of $\mathbf{A}_s$. $V(\mathbf{A}_s)$ is parameterised by its radius ω. Niu et al. (2019) has provided a indication of the optimal order of magnitude of ω by minimizing the error of the kernel estimator. When $V(A_s)$ is large, the error of estimating the transition probability becomes smaller. But the error of approximating the transition density becomes bigger. The former is called Monte Carlo error and the later is called numerical error in Niu et. al (2019). Detailed discussions about the error of the estimator and an ablation study of the choice of ω based on the Swiss roll experiment are given in Appendix E. The results show that the performance of GPUM is not very sensitive to the choice of ω.

The Neumann boundary condition corresponds to BM reflecting at the boundary. This can be approximated by pausing time and resampling the next step until it falls into the interior of $\mathbb{M}$. This estimator is asymptotically unbiased and consistent (Niu et al., 2019).

Note that t is the BM diffusion time. If t is large, the BM paths have a higher probability to reach the neighbourhood of the target point leading to higher covariance and vice versa. The transition density can be estimated using Algorithm 1.

Corollary 2 *The transition density estimate in* (25) *is coordinate independent.*

By Theorem 1, it is straightforward to have Corollary 2. The transition density estimate in (25) of the stochastic process defined in (21) is also coordinate independent. Based on Corollary 2, we can evaluate the heat kernel of the implicit manifold estimated by different LVMs, independent of the specific parameterisations of their latent spaces. If the estimated manifold is close to the true manifold, we expect the resulting BM transition density estimates to be similar to the ones estimated using the analytical parameterisation.

Here we take the Swiss roll as an example. Assuming the geometry of the Swiss roll is known, we can follow Niu et al. (2019)'s approach and evaluate the heat kernel $K^t_{heat}(s_0, s)$ by simulating BM paths with the analytical metric tensor. The derivations of the analytical metric and parameterisation of the Swiss roll are shown in Appendix D. Let s_0 with $radius = 6$ and $width = 3$ be the starting point of the BM. $N_{BM} = 20000$ BM paths are simulated. The BM transition density is evaluated using (25) at twenty nine target points $\{s_j \in \mathbb{M} \subset \mathbb{R}^3 | j \in \{1, \cdots, 29\}\}$ in the observation space. These target points are centred on s_0 and equally spaced. The diffusion time is fixed at 50. The results are plotted as the red solid line in Fig. 4. The horizontal axis is the radius of the Swiss roll and the vertical axis is the transition density. The red density plot is asymmetric. When the radius is large, the transition density estimate decreases more quickly.

If the geometry of the Swiss roll is unknown, Bayesian GPLVM and GPLVM can be applied to learn the metrics from 250 grid points on the Swiss roll. The derivations of B-GPLVM metrics and GPLVM metrics are given in section 5.2 and 5.1. The BM trajectories are simulated using the estimated metric tensors. The heat kernel estimates using B-GPLVM metrics are plotted as the green dashed line in Fig.4. It is clear that the B-GPLVM results are very close to the red solid line. The estimates using GPLVM metrics are plotted in brown dashed line. It is also close to the solid red line, but not as good as the B-GPLVM estimates. GPLVM performs point estimate of x in the latent space while B-GPLVM estimates a Bayesian posterior of x. As a result, B-GPLVM is more robust in terms of estimating the latent variables (Damianou et al., 2016).

The heat kernel estimates from the Graph Laplacian (GL) approach (Dunson et al. (2020)) in different data regime are also provided in Fig. 4. When the number of points on the Swiss roll is 250, the GL kernel estimates are plotted as the blue dashed line. It is far from the solid red line and do not match the overall pattern of the analytical kernel estimates. When the number of grid points is increased to 1000 and 10000, the GL kernel estimates are plotted as the purple dashed line and black dash dotted line. The estimates are closer to the solid red line as the number of grid points increases. Comparing to the GL approach, the kernel estimates using B-GPLVM metrics achieve the best performance with much fewer points on the Swiss roll. B-GPLVM is used in the simulation study and real data applications in later sections.

From (25) we can see the construction of GPUM and the heat kernel requires simulating BM sample paths at each data point. Although the BM simulations are trivially parallelizable, the computational cost can be high when the number of data points is large. Niu et al.

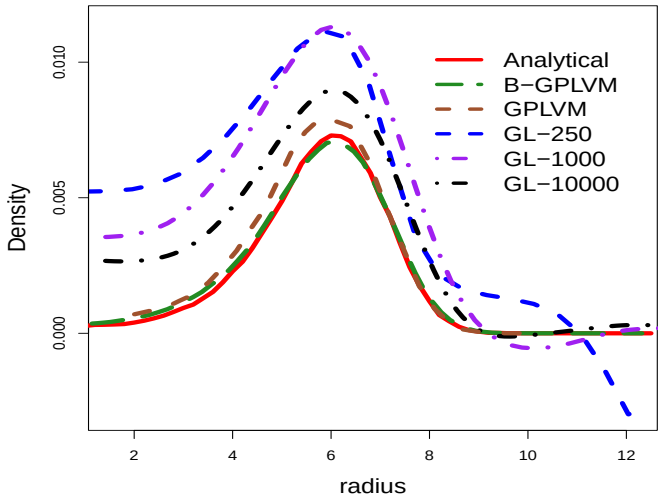


Figure 4: Comparison of heat kernel estimates using the analytical metric, B-GPLVM metric, GPLVM metric and Graph Laplacian. The red solid line represents heat kernel estimates using the analytical metric. The green dashed line represents estimates using B-GPLVM metric. The brown dash dotted line represents estimates using GPLVM metric. The heat kernel estimates from the Graph Laplacian approach are plotted as the blue dashed line. We increase the number of the grid points on the Swiss roll from 250 to 1000 and 10000. The GL estimates are plotted as the purple dotted line and black dashed line.

(2019) proposed the sparse intrinsic GP on known manifold by introducing some inducing points. The number of inducing points is much smaller than the number of data points. BM paths only need to be simulated starting at the inducing points instead of every data point. The inducing point approximation summarizes the training data into a small set of inducing points, so that inference could be done more efficiently (Quiñonero-Candela and Rasmussen, 2005). Similar approach can be applied in the GPUM when the manifold is unknown. Small number of inducing points can be introduced in the learned latent space. The focus of this paper has been on developing the GPUM. The development of the sparse GPUM is for the future research. Another example of estimating the heat kernel of the cylinder is given in Appendix H.

6.3 Optimising the kernel hyperparameters

Given a diffusion time t, we can generate the covariance matrix $\Sigma^t_{\mathbf{ff}}$ for training data $\mathcal{D}$ using Algorithm 1. $\Sigma^t_{\mathbf{ff}}$ can be obtained as follows: with the i_{th} data point as the starting point, N_{BM} trajectories are simulated to generate the i_{th} row of $\Sigma^t_{\mathbf{ff}}$. For each element of $\Sigma^t_{\mathbf{ff}}$, $\hat{K}^t(s_i, s_j)$ is then estimated using (25). The hyperparameters can be obtained by maximizing the log of the marginal likelihood (over f) in (26). The maximum BM diffusion time is set as $N_t \times \Delta t$. Δt is the BM simulation time step as defined in (23). N_t is

Algorithm 1: Simulating BM sample paths on $\mathbb{M}$ for estimating K^t_{heat}

Learn the metric $\mathcal{G}$ from the point cloud $\mathcal{S} = \{s_i | i = 1, \ldots, n+v\}$. {use eqn (8) }
1.1 Generate BM trajectories on implicit manifold.
for $i = 1, \ldots, n$ { n is the size of data points } **do**
  for $j = 1, \ldots, N_{BM}$ {N_{BM} is No. of trajectories } **do**
    for $l = 1, \ldots, N_t$ {N_t steps Brownian motion, $N_t \times \Delta t \to$ max diffusion time} **do**
      do {keep proposing x until it falls inside of the boundary }
      $q\left(x_{i,j}(l) | x_{i,j}(l-1)\right) = \mathbb{N}\left(x_{i,j}(l) | \mu\left(x_{i,j}\left(l-1\right), \Delta t\right), \Delta t \mathcal{G}^{-1}\right)$ { use eqn (23) }
      While $\mathbb{V}ar(\phi(x_{i,j})) > \alpha$, $x_{i,j}$ is outside of the boundary $\partial\mathbb{M}$. { use eqn (20) }
    end for
  end for
end for
return $\boldsymbol{x}$
1.2 Given a discrete choice of the diffusion time $t \in \{\Delta t, 2\Delta t, \cdots, N_t \Delta t\}$, the covariance matrix Σ^t is estimated based on the BM simulation from Algorithm 1.1.
for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, n$ **do**
    $N_{\mathbf{A}_j}$ = which(x(t) $\in \mathbf{A}_j$) { counting how many BM paths reach $\mathbf{A}_j$ }
    $K^t_{heat}(s_i, s_j) = \frac{N_{\mathbf{A}_j}}{N_{BM} * V(\mathbf{A}_j)}$ { use eqn (25) }
    $\Sigma^t_{ij} = \sigma_h^2 K^t_{heat}(s_i, s_j)$
  end for
end for
return Σ^t

the number of simulation steps. N_t covariance matrices $\Sigma_{\mathbf{ff}}^{1\ldots N_t}$ can be generated based on the BM simulations. Optimisation of the diffusion time t can be done by selecting the corresponding $\Sigma^t_{\mathbf{ff}}$ which maximizes the log marginal likelihood. Estimation of σ_h follows standard optimisation routines, such as quasi-Newton.

$$\begin{aligned}\log p(\boldsymbol{y}|s) &= \log \int p(\boldsymbol{y}|\mathbf{f}) p(\mathbf{f}|s) d\mathbf{f} \\ &= -\frac{1}{2}\boldsymbol{y}^T (\Sigma^t_{\mathbf{ff}} + \sigma^2_{noise} I)^{-1} \boldsymbol{y} - \frac{1}{2}\log|\Sigma^t_{\mathbf{ff}} + \sigma^2_{noise} I| - \frac{n}{2}\log 2\pi. \end{aligned} \tag{26}$$

Let $\mathbf{f}_*$ be a vector of values of $f(\cdot)$ at the unlabeled points in $\mathcal{V}$. Under the regression model in (4) and the GPUM prior, we have the joint distribution of $\mathbf{y}$ and $\mathbf{f}_*$:

$$p(\mathbf{y}, \mathbf{f}_*) = \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{\mathbf{ff}} + \sigma^2_{noise} I_n & \Sigma_{\mathbf{ff}_*} \\ \Sigma_{\mathbf{f}_*\mathbf{f}} & \Sigma_{\mathbf{f}_*\mathbf{f}_*} \end{bmatrix}\right), \tag{27}$$

where $\Sigma_{\mathbf{f}_*\mathbf{f}}$ is the covariance matrix for training(labeled) and unlabeled data points. The predictive distribution is derived by marginalising out $\mathbf{f}$:

$$p(\mathbf{f}_*|\boldsymbol{y}) = \int p(\mathbf{f}_*|\mathbf{f}, \boldsymbol{y}) d\mathbf{f} = \mathcal{N}\left(\Sigma_{\mathbf{f}_*\mathbf{f}}\left(\Sigma_{\mathbf{ff}} + \sigma^2_{noise} I\right)^{-1} \boldsymbol{y},\ \ \Sigma_{\mathbf{f}_*\mathbf{f}_*} - \left(\Sigma_{\mathbf{ff}} + \sigma^2_{noise} I\right)^{-1} \Sigma_{\mathbf{ff}_*}\right). \tag{28}$$

7. Simulation study on Swiss roll

In this section, we carry out a simulation study for a regression model with synthetic data on the Swiss roll which is a two dimensional manifold depicted by a point cloud in $\mathbb{R}^3$. The

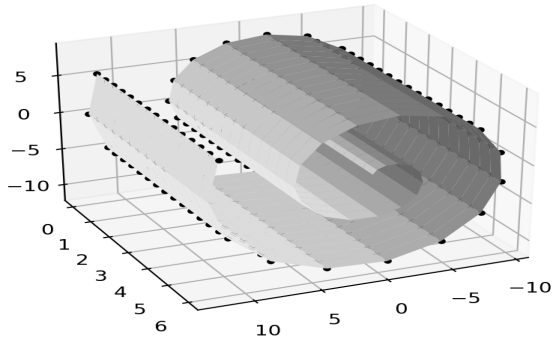
point cloud is plotted in Fig. 5(a). It is comprised of the set of labeled points $n = 24$ and the set of unlabeled points $v = 450$. Both labeled and unlabeled observed points are used in B-GPLVM to learn the latent space. The point cloud is unfolded into a flat surface as in Fig. 5(b). The unlabeled points are plotted as blue triangles and the labeled points are in red in the latent space. The variance of the mapping is plotted as the gray background of the latent space. It is clear from Fig. 5(b) that the background color gets darker in the region further away from the blue triangles. This indicates the variance of the mapping gets bigger in the region which is far from the observations.

Based on the definition in (10), the boundary $\partial\mathbb{M}$ is plotted in Fig. 5(d). The black regions on the left and right sides of Fig. 5(d) are outside of $\partial\mathbb{M}$. The white area in the middle is within $\partial\mathbb{M}$. The magnification factor is plotted as the background color in Fig. 5(c). Similar to Fig. 2, the horizontal axis can be interpreted as the scaled radius of the Swiss roll. When the radius is bigger, the corresponding magnification factor is larger (the color is darker). However, as there are fewer data points available at the tail of the Swiss roll, the estimation of the implicit manifold from B-GPLVM becomes less accurate. This can be observed on the right end of Fig. 5(c) where the last column of the observed latent points is further away from the rest. Once the metric $\mathcal{G}$ and the boundary $\partial\mathbb{M}$ are learned from B-GPLVM, we can estimate the heat kernel by simulating BM paths on the implicit manifold. An example of a BM sample path on Swiss roll is shown in Fig. 2.

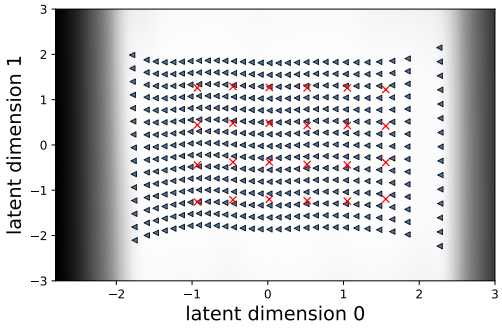
For the labeled points, the response variables are

$$y_i = f(s_i^1, s_i^2, s_i^3) + \epsilon_i, \quad i = 1 \dots n, \quad s_i \in \mathbb{R}^3$$

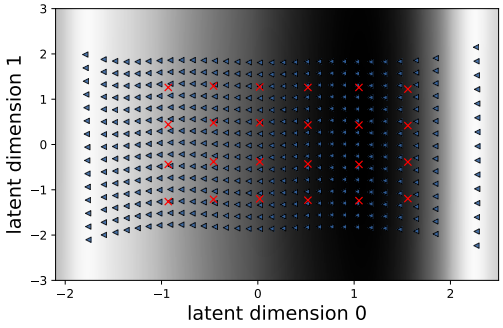
where f is the unknown regression function, s_i is the coordinate of the observed point in $\mathbb{M} \subset \mathbb{R}^3$. For better visualisation, the true function is plotted in the unfolded Swiss roll in Fig.6(a) using a two dimensional analytical parameterisation. The coordinates are radius and width. 24 labeled observations are marked as black crosses. The true function values are indicated by the color codes and contours. The regression function varies slowly when the radius is small and rapidly when the radius is big.



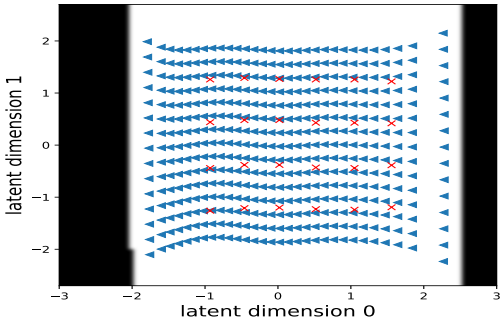
(a) Swiss roll point cloud in $\mathbb{R}^3$



(b) Latent space and variance of mapping



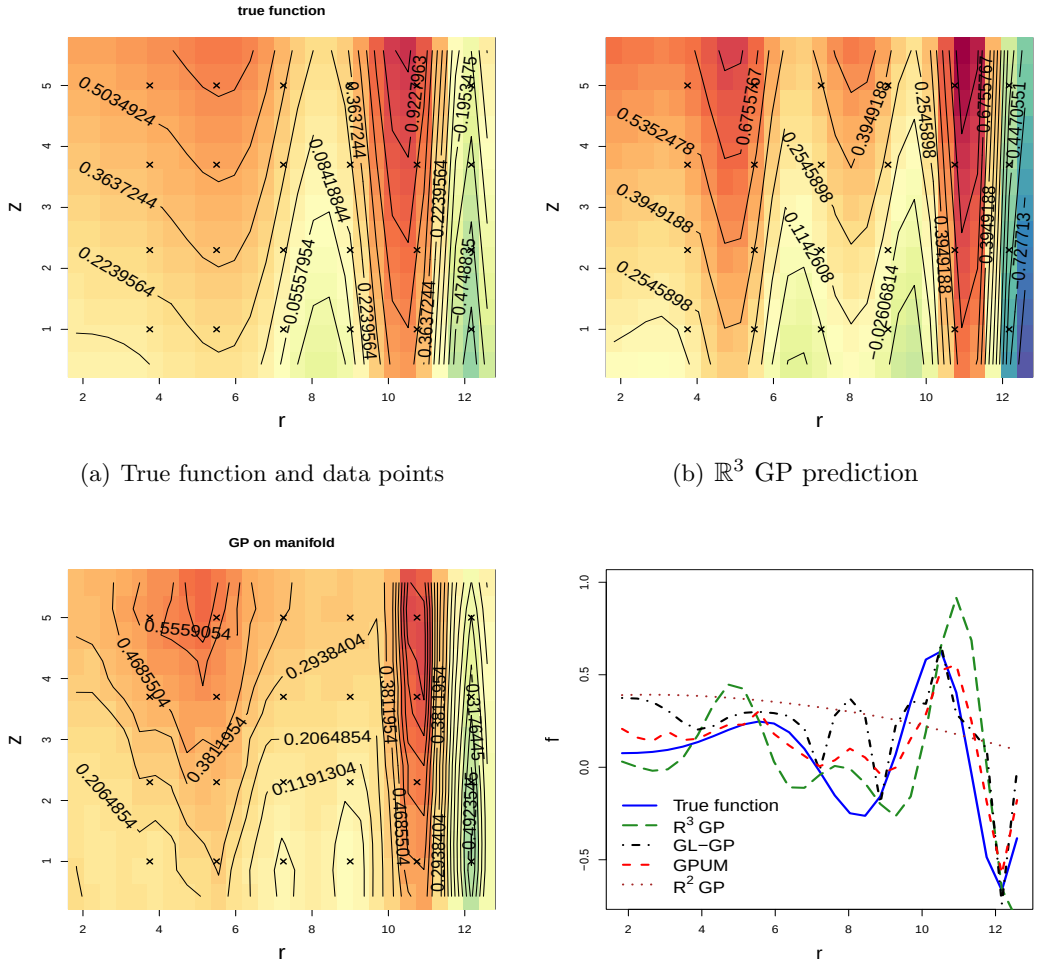
(c) Latent space and magnification factor



(d) Boundary

Figure 5: (a) The Swiss roll represented as a point cloud in $\mathbb{R}^3$. (b) The latent space constructed from Bayesian GPLVM. The blue triangles are unlabeled points and red crosses are labeled points. The background color represents the variance of the mapping, a dark background represents high uncertainty. (c) The same latent space visualization with the background color representing the magnification factor, a dark background represents high magnification factor. (d) The same latent space visualization highlighting the boundary. The dark region is outside of the boundary and the white region is inside of the boundary.

We first apply the Euclidean $\mathbb{R}^3$ GP (the standard GP as in Rasmussen and Williams (2006) Chapter 4) constructed by the squared exponential kernel with $\mathbb{R}^3$ Euclidean distance. With this kernel setting, the $\mathbb{R}^3$ GP completely ignores the interior structure of the manifold and lets the inner layer and outer layer of the Swiss roll interact. The predictive means on the unlabeled points are shown in Fig. 6(b). Compared to the true function in Fig.6(a), the overall shape of $\mathbb{R}^3$ GP prediction contours is more wiggly. The color coding of Fig.6(b) is also very different in the regions where the radius (horizontal axis) is 6, 8 and 10. In the second model, the $\mathbb{R}^2$ Euclidean distance in the latent space is used in the squared exponential kernel to construct the $\mathbb{R}^2$ GP. The geometric properties such as the metric and magnification factors are ignored. The color coding of the predictive means for $\mathbb{R}^2$ GP is shown in Fig. 11(a) in Appendix D. Since the regression function is nonstationary in the latent space, the $\mathbb{R}^2$ GP is underfitting.



(a) True function and data points

(b) $\mathbb{R}^3$ GP prediction

(c) GPUM prediction

(d) Comparison of predictions when fixing z

Figure 6: (a) The true function in the unfolded Swiss roll. The labeled points are marked with black crosses. (b) The prediction of $\mathbb{R}^3$ GP. $\mathbb{R}^3$ GP is constructed using the Euclidean distance in $\mathbb{R}^3$. (c) The prediction of GPUM . (d) Comparing the predictions of all methods on a spiral in the swiss roll by fixing the z coordinate at 4 and changing the radius from 2 to 12.5. The vertical axis represent the value of the prediction. The horizontal axis represents the radius.

The GPUM predictive mean is shown in Fig. 6(c). The overall pattern of the GPUM prediction is similar to the true function. The shape of the contours and the color coding of Fig.6(c) are consistent with Fig.6(a). The prediction of the GL-GP is shown Fig.11(b) in Appendix D. Since the GL kernel estimates in Fig. 4 are far from the analytical kernel, the GL-GP prediction is also poor. A one dimensional comparison is generated by plotting the predictions of all methods at $z = 4$ and varying the radius from 2 to 12.5 in Fig.6(d). It is clear the $\mathbb{R}^2$ prediction(brown dotted line) is under fitting. The $\mathbb{R}^3$ prediction(green dashed line) is oscillating in the opposite direction of the ground truth (blue solid line).

Table 1: Comparison of the root mean squared errors of five methods on Swiss roll. Values in parentheses show the standard deviation.

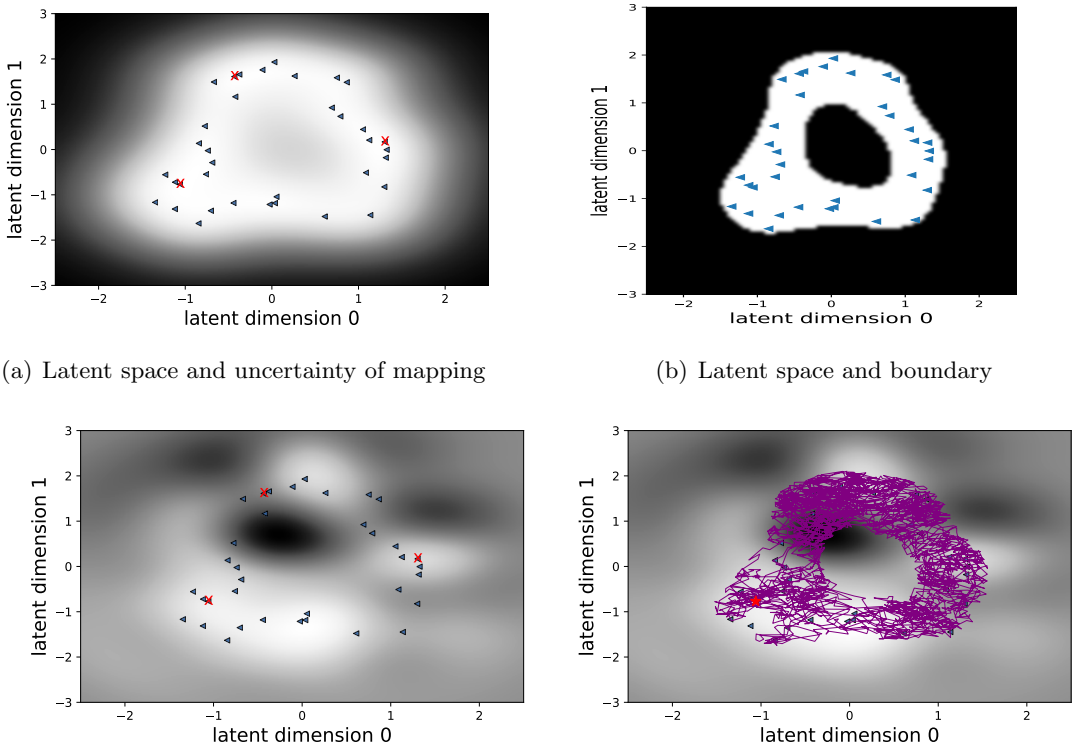
	$\mathbb{R}^3GP$	$\mathbb{R}^2GP$	GPUM	GL-GP	GM-GP
RMSE $v = 250$	0.284(0.006)	0.293(0.005)	0.163(0.020)	0.243(0.003)	0.231(0.001)
RMSE $v = 450$	0.298(0.007)	0.290(0.005)	0.162(0.003)	0.220(0.002)	0.207(0.002)
RMSE $v = 800$	0.287(0.006)	0.282(0.005)	0.164(0.002)	0.216(0.001)	0.206(0.001)

The GL-GP prediction (black dashed line) is also oscillating around the blue solid line. The GPUM prediction (red dashed line) achieves the best performance and follows the overall pattern of the ground truth.

In order to evaluate the performance of different methods in different data regimes, we consider three scenarios with different numbers of unlabeled grid points on the Swiss roll from $v = 250$ to $v = 450$ and $v = 800$. These unlabeled grid points are used to estimate the manifold for GPUM , GM-GP, GL-GP and are also used as the test set of the regression accuracy. We constructed twenty sets of training points by randomly selecting $n = 23$ labeled points from the labeled set. For each training set, GPUM , GM-GP, GL-GP, $\mathbb{R}^2$GP and $\mathbb{R}^3$GP have been applied to make predictions at the points in the test set. The root mean square errors (RMSE) are calculated between the true function values and the predictive means of all five models. The mean and standard deviation of the root mean square errors are reported in Table 1. GPUM's results are consistent across all three scenarios and significantly better than all other methods. The GL-GP and GM-GP have similar performance. Both models perform better when the number of the grid points is large. They are significantly better than the Euclidean GPs. Comparing to the graph based approaches, GPUM achieves the minimum mean RMSE with fewer points on the manifold.

8. Location estimation from WiFi signal

Indoor location estimation has tremendous value but standard location estimation techniques such as Global Positioning System (GPS) do not work indoors. Instead, indoor wireless signals from devices such as WiFi access points can be exploited for location estimation. In this section, we consider the problem of indoor 2D location estimation from WiFi access point signal strengths. We use the WiFi data collected by Ferris et al. (2007), in which a series of WiFi signal strength traces are collected by a mobile device which travels in a one floor university building. The 2D location coordinates of the mobile device are also recorded by a click-to-map based annotation program. The total number of WiFi access points in this dataset is 30. As it is often expensive to collect labeled data, we mimic a low data indoor location estimation scenario by assuming that the true locations of only three points are known and we aim at predicting the indoor locations of the remaining points based on the WiFi signals.



(a) Latent space and uncertainty of mapping

(b) Latent space and boundary

(c) Latent space and magnification factor

(d) A BM trajectory

Figure 7: (a) The latent space constructed from Bayesian GPLVM. The blue triangles are unlabeled points and the red crosses are labeled points. The background color represents the variance of the mapping, a dark background represents high uncertainty. (b) The same latent space visualization highlighting the boundary. The dark region is outside of the boundary and the white region is inside of the boundary. (c) The same latent space visualization with the background color representing the magnification factor, a dark background represents a high magnification factor. (d) A BM trajectory is shown in purple. The red star is the starting location.

This location estimation problem can be treated as a regression problem, in which the location coordinate y of the mobile device can be modelled as a function of the high dimensional WiFi signal

$$y_i = f(s_i) + \epsilon_i, \quad i = 1 \dots n, \quad s_i \in \mathbb{R}^{30}$$

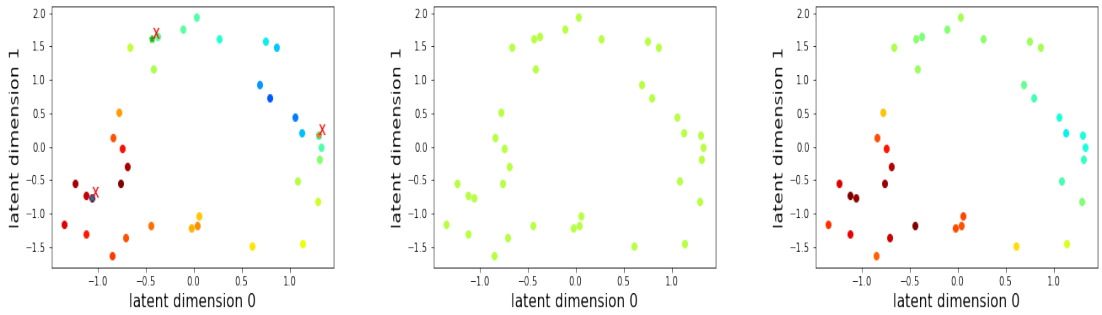
where f is the unknown regression function and WiFi signal s_i is represented by a 30 dimensional vector. Here we consider the WiFi signal measurements at $n+v = 36$ locations. Only $n = 3$ of the locations are labeled with one dimensional location coordinates of the mobile device. To avoid selecting the three points clustered together, we randomly pick the points from three different regions respectively, where the union of the regions cover the whole dataset. Different methods are applied to estimate the coordinates of the mobile device in the testing sets (the unlabeled $v = 33$ locations). Twenty training and testing sets are generated from this random selection.

Table 2: Comparison of the root mean squared errors of five methods on WiFi signal data. Values in parentheses show the standard deviation.

	$\mathbb{R}^{30}$GP	$\mathbb{R}^{2}$GP	GPUM	GL-GP	GM-GP
mean RMSE	5.57(1.43)	4.83(1.83)	4.11(0.88)	5.6(1.15)	6.04 (0.57)

Unlike the simulation study of the Swiss roll, we cannot plot the high dimensional point cloud of the WiFi signals. The implicit manifold is also unknown. We first estimate a $q = 2$ dimensional latent space (the chart of the underlying manifold) using B-GPLVM. The value of q is determined by the ARD contributions which measure how much each dimension is contributing to the latent space. The input dimensions are sorted based on the relevance assigned by the scaling in B-GPLVM. The plot of ARD contributions is shown in Appendix F. The latent space is plotted with the variance of the mapping as the gray background in Fig. 7(a). The 36 WiFi signal strength measurements are represented by the blue triangles. The training set (the labeled points) is marked by the red crosses. The dark color value is for high uncertainty. Since the mobile device moves in a loop closure, the latent points forms a closed loop in the latent space. The boundary of the implicit manifold ∂M is shown in Fig. 7(b) which is defined by (10). The magnification factor is plotted as the gray background in Fig.7(c). The dark color value is for high magnification factor. A sample path of BM in the implicit manifold is plotted in Fig. 7(d). With the Neumann boundary condition the BM path can only exist within the boundary.

The coordinates of the mobile device are estimated by five different methods. The ground truth values are marked by different colors in the latent space in Fig. 8(a). If we ignore the interior structure of the manifold, a $\mathbb{R}^{30}$ GP using the squared exponential kernel with $\mathbb{R}^{30}$ Euclidean distance of the WiFi signals is applied. The predictive means of the $\mathbb{R}^{30}$ GP are plotted in Fig.8(b). It is clear the $\mathbb{R}^{30}$ GP prediction is very poor. The color coding of the prediction is different from the ground truth. The GPUM predictive mean is shown in Fig. 8(c). The overall pattern of the GPUM prediction is similar to the ground truth. In the third case, the $\mathbb{R}^2$ Euclidean distance in the latent space is used with the squared exponential kernel to construct the $\mathbb{R}^2$ GP. The prediction results are shown in Fig.13 in Appendix F. The Graph Laplacian based approaches such as GL-GP and GM-GP have also been applied. Since the number of observations is relatively low ($n + v = 36$), the graph based methods result in poor approximation of the implicit manifold. The GL-GP's performance is similar to the $\mathbb{R}^{30}$GP. The mean and standard deviation of the root mean square errors of the twenty testing sets are also calculated in Table 2 for all five models. GPUM significantly outperforms the other four models and achieves the minimum mean RMSE.



(a) Ground truth (b) $\mathbb{R}^{30}$ GP prediction (c) GPUM prediction

Figure 8: Comparison of GPUM and Euclidean GPs in WiFi signal example. (a) The ground truth is plotted with color at the observed points in the latent space. (b) $\mathbb{R}^{30}$ GP prediction. $\mathbb{R}^{30}$ GP is constructed using the Euclidean distance of WiFi signals in $\mathbb{R}^{30}$. (c) GPUM prediction.

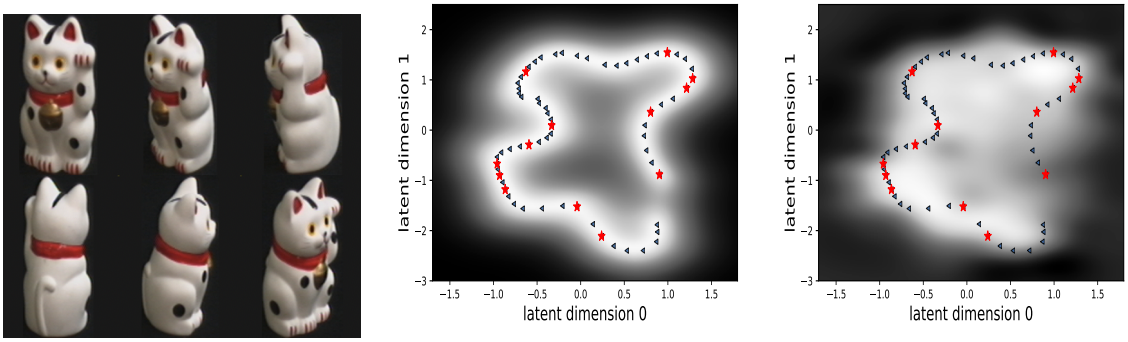
9. Camera angle estimation from images

In this section, we consider the problem of estimating the camera angle from images. We consider the setting in which an object is placed on a turntable and a set of images are taken at different angles with respect to the camera. We aim at recovering the camera angles associated with individual images by knowing the true camera angles for some of the images. We use the images from the COIL data set (Nayar and Murase, 1996). Here we consider 66 images of object-14 (a toy cat). The camera angles range from 15 degrees to 340 degrees. The raw images are converted to grayscale and downscaled to 32×32. The raw pixels of each image are flattened into a 1024 dimensional vector. The dimension of the original image space is $p = 1024$. Six image examples are given in Fig. 9(a). We estimate a $q = 2$ dimensional latent space using Bayesian GPLVM. The two dimensional latent space is plotted with the variance of the mapping as the gray background in Fig. 9(b). A dark background represents a high uncertainty in the corresponding region. The unlabeled data points are marked as blue triangles and the labeled data points as red stars. The overall shape of the latent points in Fig.9(b) looks like a ring with a gap in the lower right. The magnification factor is plotted as the gray background in Fig. 9(c). A dark background represents a high magnification factor. The boundary of the implicit manifold $\partial\mathbb{M}$ is shown in Fig.14 in Appendix G.

The scaled camera angle y is modelled as a function of the image.

$$y_i = f(s_i) + \epsilon_i, \quad i = 1 \dots n, \quad s_i \in \mathbb{R}^{1024}.$$

where f is the unknown regression function, and s_i is represented by a 1024 dimensional vector. n=13 images are randomly selected and used as the training set (labeled data). They are plotted as the red stars in the latent space in Fig.9(b). The remaining 53 images are used as the testing set (unlabeled data) which are plotted as the blue triangles in the latent space in Fig. 9(b). The true angle values are marked by different colors on the observed points in the latent space in Fig. 10(a). Different methods have been applied



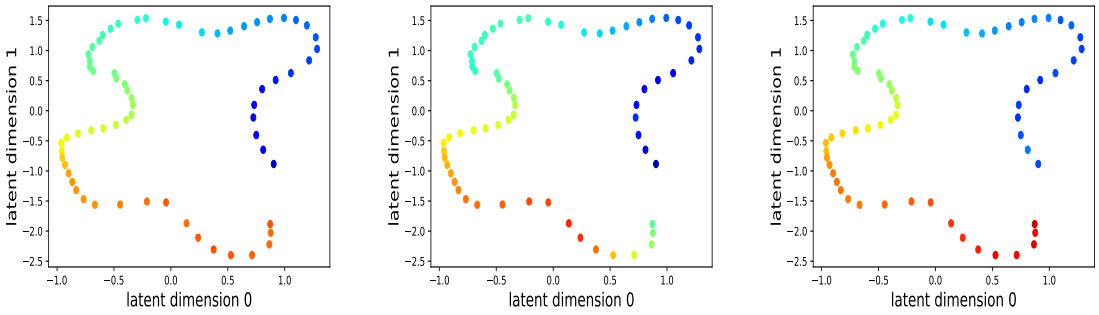
(a) COIL images (b) Latent space and variance of mapping (c) Latent space and magnification factor

Figure 9: (a) Examples of six COIL object images. (b) The latent space constructed from Bayesian GPLVM. The blue triangles are unlabeled points and red stars are labeled points. The background color represents the variance of the mapping. The dark color represents high uncertainty. (c) The same latent space visualization with the background color representing the magnification factor. The dark color represents a high magnification factor.

Table 3: Comparison of the root mean squared errors of five methods on COIL images. Values in parentheses show the standard deviation.

	$\mathbb{R}^{1024}$GP	$\mathbb{R}^2$GP	GPUM	GL-GP	GM-GP
MEAN RMSE	0.097(0.040)	0.094(0.042)	0.060(0.038)	0.116(0.027)	0.143(0.026)

to estimate the scaled camera angles. The $\mathbb{R}^{1024}$GP using the squared exponential kernel with $\mathbb{R}^{1024}$ Euclidean distance is applied to the image data first. The predictive means of the $\mathbb{R}^{1024}$GP are plotted in Fig.10(b). The color coding of the prediction at the lower right of the plot is bright green which is different from the truth. Ignoring the boundary and the magnification factor in the latent space, the $\mathbb{R}^2$GP using the squared exponential kernel with $\mathbb{R}^2$ Euclidean distance in the latent space is applied to the image data. The predictive means are shown in Fig.15 in Appendix G. The predictive means of GPUM are plotted in Fig.10(c). The overall pattern of the GPUM prediction is very similar to the true function. Since the boundary of the implicit manifold is defined in the lower right region of the latent space, the GPUM does not smooth across the boundary and gives a better prediction. Ten training and testing sets are generated from the random selection. The mean and standard deviation of the root mean square errors are calculated for the ten testing sets for all methods in Table 3. The GPUM achieves the smallest mean RMSE. It is significantly better than the GL-GP and GM-GP. The difference between the GPUM and the $\mathbb{R}^{1024}$GP is not significant. The boxplot of the root mean squared errors are shown in Fig.16 in Appendix G.



(a) True function (b) $\mathbb{R}^{1024}$ GP prediction (c) GPUM on manifold prediction

Figure 10: Comparison of GPUM and Euclidean GPs in COIL images example. (a) The true function is plotted with color at the observed points in the latent space. (b) $\mathbb{R}^{1024}$ GP prediction. $\mathbb{R}^{1024}$ GP is constructed using the Euclidean distance of data points in $\mathbb{R}^{1024}$. (c) GPUM prediction.

10. Discussion

Our work provides a novel framework for regression on implicit manifolds embedded in high dimensional point clouds. The geometry of the implicit manifold is learned by probabilistic latent variable models. This gives a distribution over a smoothly changing local metric at each point in the latent space. The boundary of the implicit manifold is defined according to the uncertainty of the mapping. The expression for the expected local metric is derived and used to simulate BM sample paths on manifolds. We have proved the BM transition density estimation is coordinate independent in section 6.2. This allows us to compare the transition density estimates from different parameterisations of the same manifold. The BM simulation using the B-GPLVM metric gives similar transition density estimation results as the BM simulation using the analytical metric. GPUM is constructed by employing the equivalence relationship between heat kernels and the transition density of BM on manifolds. This allows the GPUM to incorporate the intrinsic geometry of the implicit manifold for inference while respecting the interior constraints and boundary. The experiment results in section 7, 8 and 9 indicate that GPUM achieves significant improvements over Euclidean GPs and Graph Laplacian based GPs. Although the simulation of BM paths can be easily parallelised, Algorithm 1 can still be computational expensive if the sample size is large. The number of the sample paths can be reduced by leveraging the idea of pseudo data, which has been widely studied in the literature of sparse GPs (Quiñonero-Candela and Rasmussen, 2005; Niu et al., 2019).

Acknowledgments

M. Niu acknowledges the support of EPSRC grants EP/W021595/1 and EP/X5257161/1.

Appendix A. Proof of Theorem 1 and Corollary 2

The SDE in (22) can be derived by expressing the heat equation in local coordinates as a Fokker-Planck equation, which in particular implies its coordinate independence. In this appendix, however, we will provide a direct verification that (22) is invariant under coordinate changes.

Some notations we need in the proof are summarised here. $\mathbb{M}$ is a manifold of dimension q with a Riemannian metric $g_{\mathbb{M}}$. $\theta : \mathbb{U} \to \mathbb{M}$ and $\bar{\theta} : \bar{\mathbb{U}} \to \mathbb{M}$ are two local coordinate charts of $\mathbb{M}$ where $\mathbb{U}$ and $\bar{\mathbb{U}}$ are open subsets of $\mathbb{R}^q$.

- $\psi = \bar{\theta}^{-1} \circ \theta : \mathbb{U} \to \bar{\mathbb{U}}$ is the change of coordinate. $\psi^{-1} = \theta^{-1} \circ \bar{\theta} : \bar{\mathbb{U}} \to \mathbb{U}$ is the inverse change of coordinate. (assume $\theta(\mathbb{U}) = \bar{\theta}(\bar{\mathbb{U}})$ without loss of generality.)
- $\left(x^1, \ldots, x^q\right)$ denotes the standard coordinates in $\mathbb{U} \in \mathbb{R}^q$. $\left(\bar{x}^1, \ldots, \bar{x}^q\right)$ denotes the standard coordinates in $\bar{\mathbb{U}} \in \mathbb{R}^q$.
- g denotes matrix representation of $g_{\mathbb{M}}$ in $\mathbb{U}$ (via θ) i.e. $g_{ij} = g_{\mathbb{M}}\left(\theta_* \frac{\partial}{\partial x^i}, \theta_* \frac{\partial}{\partial x^j}\right)$. g^{ij} denotes the element of g^{-1}. $G = det(g)$. Similarly for $\bar{g}$, $\bar{g}_{ij}$, $\bar{g}^{ij}$, $\bar{G}$.
- $D\psi$ is the matrix derivative of $\psi = \left(\psi^1\left(x^1, \ldots x^q\right), \ldots, \psi^q\left(x^1, \ldots, x^q\right)\right)$, i.e. $(D\psi)_i^j = \frac{\partial \psi^j}{\partial x^i}$. Using the chain rule $\Rightarrow D\psi^{-1} \cdot D\psi = I_q$.
- $\frac{\partial}{\partial x^i} = \sum_{j=1}^q \frac{\partial \psi^j}{\partial x^i} \frac{\partial}{\partial \bar{x}^j} = \sum_{j=1}^q (D\psi)_i^j \frac{\partial}{\partial \bar{x}^j} \quad \Rightarrow \quad g_{ij} = (D\psi)_i^k \bar{g}_{k\ell} (D\psi)_j^\ell \quad \Rightarrow g^{-1} = \left(D\psi^{-1}\right)^\top \bar{g}^{-1} D\psi^{-1}$, $G = (\det \mathcal{D}\psi)^2 \bar{G}$.

Proof Consider a stochastic process in $\mathbb{U} \in \mathbb{R}^q$ defined by:

$$dx^i = \frac{1}{2} \frac{1}{\sqrt{G}} \sum_{j=1}^q \frac{\partial}{\partial x^j}\left(\sqrt{G} g^{ij}\right) dt + \left(g^{-\frac{1}{2}} dB\right)^i, \quad i = 1, \ldots, q. \tag{29}$$

Note that

$$\begin{aligned} dx^i dx^j &= \left(g^{-\frac{1}{2}} dB\right)^i \left(g^{-\frac{1}{2}} dB\right)^j = \left(\sum_{k=1}^q \left(g^{-\frac{1}{2}}\right)^{ik} dB^k\right) \cdot \left(\sum_{l=1}^q \left(g^{-\frac{1}{2}}\right)^{jl} dB^l\right) \\ &= g^{ij} dt \quad i, j = 1, \ldots, q. \end{aligned}$$

ψ maps the above process in $\mathbb{U}$ to a process in $\bar{\mathbb{U}}$ defined by:

$$\begin{aligned} d\bar{x}^i &= d\psi^i\left(x^1, \ldots, x^q\right) \\ &= \sum_{j=1}^q \frac{\partial \psi^i}{\partial x^j} dx^j + \frac{1}{2} \sum_{j=1}^q \sum_{k=1}^q \frac{\partial^2 \psi^i}{\partial x^j \partial x^k} dx^j dx^k \\ &= \frac{1}{2} \sum_{j=1}^q \sum_{k=1}^q \left[(D\psi)_j^i \cdot \frac{1}{\sqrt{G}} \frac{\partial}{\partial x^k}\left(\sqrt{G} g^{jk}\right) + \frac{\partial^2 \psi^i}{\partial x^j \partial x^k} g^{jk}\right] dt + \sum_{j=1}^q (D\psi)_j^i \left(g^{-\frac{1}{2}} dB\right)^j. \end{aligned} \tag{30}$$

The dt term in (30) can be written as follows

$$\begin{aligned}
&\frac{1}{2}\sum_{j=1}^{q}\sum_{k=1}^{q}\left[(D\psi)_j^i\cdot\frac{1}{detD\psi}\cdot\frac{1}{\sqrt{\bar{G}}}\cdot\sum_{\ell=1}^{q}(D\psi)_k^l\cdot\frac{\partial}{\partial\bar{x}^\ell}\left(detD\psi\cdot\sqrt{\bar{G}}\sum_{r=1}^{q}\sum_{s=1}^{q}\left(D\psi^{-1}\right)_r^j\bar{g}^{rs}(D\psi^{-1})_s^k\right)\right.\\
&\left.+\left(\sum_{l=1}^{q}(D\psi)_j^l\frac{\partial}{\partial\bar{x}^\ell}(D\psi)_k\right)\left(\cdot\sum_{r=1}^{q}\sum_{s=1}^{q}\left(D\psi^{-1}\right)_r^j\bar{g}^{rs}\left(D\psi^{-1}\right)_s^k\right)\right]dt\\
=&\frac{1}{2}\sum_{l=1}^{q}\frac{1}{\sqrt{\bar{G}}}\frac{\partial}{\partial\bar{x}^\ell}\left(\sqrt{\bar{G}}\bar{g}^{il}\right)dt+\frac{1}{2}\sum_{s=1}^{n}\mathrm{Tr}\left[D\psi\cdot\frac{\partial}{\partial\bar{x}^s}\left(D\psi^{-1}\right)\right]\cdot\bar{g}^{is}dt\\
&+\frac{1}{2}\sum_{j=1}^{q}\sum_{l=1}^{q}\sum_{r=1}^{q}\frac{\partial}{\partial\bar{x}^\ell}\left[(D\psi)_j^i\left(D\psi^{-1}\right)_r^j\right]\cdot\bar{g}^{rl}dt+\frac{1}{2}\sum_{l=1}^{n}\mathrm{Tr}\left[\frac{\partial}{\partial\bar{x}^\ell}(D\psi)\cdot\left(D\psi^{-1}\right)\right]\bar{g}^{i\ell}dt\\
=&\frac{1}{2}\sum_{l=1}^{q}\frac{1}{\sqrt{\bar{G}}}\frac{\partial}{\partial\bar{x}^\ell}\left(\sqrt{\bar{G}}\bar{g}^{il}\right)dt+\frac{1}{2}\sum_{s=1}^{q}\mathrm{Tr}\left[\frac{\partial}{\partial\bar{x}^s}\left(D\psi\cdot D\psi^{-1}\right)\right]\cdot\bar{g}^{is}dt+0\\
=&\frac{1}{2}\sum_{\ell=1}^{q}\frac{1}{\sqrt{\bar{G}}}\frac{\partial}{\partial\bar{x}^\ell}\left(\sqrt{\bar{G}}\bar{g}^{i\ell}\right)dt+0.
\end{aligned}\tag{31}$$

The dB term in (30) can be written in vector form as

$$(D\psi)^\top\cdot g^{-\frac{1}{2}}\cdot dB.\tag{32}$$

Since

$$(D\psi)^\top\cdot g^{-\frac{1}{2}}\cdot dB\quad\left((D\psi)^\top g^{-\frac{1}{2}}dB\right)^\top=\bar{g}^{-1}dt,$$

we have

$$(D\psi)^\top\cdot g^{-\frac{1}{2}}\cdot dB\sim\mathcal{N}\left(0,\bar{g}^{-1}dt\right),\tag{33}$$

i.e. the same distribution as $\bar{g}^{-1/2}dB$.

Conclusion: ψ maps the process in $\mathbb{U}\subset\mathbb{R}^n$ defined by

$$dx^i=\frac{1}{2}\frac{1}{\sqrt{G}}\sum_{j=1}^{q}\frac{\partial}{\partial x^j}\left(\sqrt{G}g^{ij}\right)dt+\left(g^{-\frac{1}{2}}dB\right)^i,\quad i=1,\ldots,q,\tag{34}$$

to the process in $\bar{\mathbb{U}}\subset\mathbb{R}^n$ defined by

$$d\bar{x}^i=\frac{1}{2}\frac{1}{\sqrt{\bar{G}}}\sum_{j=1}^{q}\frac{\partial}{\partial\bar{x}^j}\left(\sqrt{\bar{G}}\bar{g}^{ij}\right)dt+\left(\bar{g}^{-\frac{1}{2}}dB\right)^i,\quad i=1,\ldots,q.\tag{35}$$

This verifies that the process defined by the above formula is coordinate independent.

As a result, with a given dt, simulating one step using any choice of local coordinates as above is equivalent as a step in $\mathbb{M}$.

■

Appendix B. GPLVM metric

Following the description in section 3.1, the joint distribution of the j_{th} dimension of the mapping ϕ and the j_{th} column of the Jacobian can be written as

$$\begin{bmatrix} \phi(\mathcal{X})^j \\ \frac{\partial \phi(x_*)^j}{\partial x} \end{bmatrix}, \sim \mathcal{N}\left(0, \begin{bmatrix} K_{\mathcal{X},\mathcal{X}} & \partial K_{\mathcal{X},*} \\ \partial K_{\mathcal{X},*}^T & \partial^2 K_{*,*} \end{bmatrix}\right). \tag{36}$$

We choose the covariance kernel k as RBF kernel, the (i,j) element of $K_{\mathcal{X},\mathcal{X}}$ is

$$k(x_i, x_j) = \gamma \exp(-\rho ||x_i - x_j||^2).$$

The derivatives of K are:

$$(\partial K_{\mathcal{X},*})_i^l = \frac{\partial k_{x_i,x_*}}{\partial x_*^l} = \rho(x_i^l - x_*^l)k(x_i, x_*), \tag{37}$$

$$(\partial^2 K_{\mathcal{X},*})_i^{r,l} = \frac{\partial^2 k_{x_i,x_*}}{\partial x_i^r \partial x_*^l} = \begin{cases} -4\rho^2(x_i^r - x_*^r)(x_i^l - x_*^l)k(x_i, x_*), & \text{if } r \neq l, \\ 2\rho(1 - 2\rho(x_i^r - x_*^r)^2)k(x_i, x_*), & \text{if } r = l, \end{cases}$$

$$(\partial^2 K_{*,*})^{r,l} = \frac{\partial^2 k_{x_*,x_*}}{\partial x_*^r \partial x_*^l} = \begin{cases} 0, & \text{if } r \neq l, \\ 2\rho k_{x_*,x_*} = 2\rho\gamma, & \text{if } r = l. \end{cases}$$

We also need the gradient of the expected metric to simulate BM trajectories. It requires computing $\partial E[\mathbf{J}^T]^j / \partial x_*^l$ and $\partial \Sigma_{\mathbf{J}} / \partial x_*^l$.

$$\frac{\partial E[\mathbf{J}^T]^j}{\partial x_*^l} = \frac{\partial \mu_{\mathbf{J}}^j}{\partial x_*^l} = \frac{\partial (\partial K_{\mathcal{X},*}^T)}{\partial x_*^l} K_{\mathcal{X},\mathcal{X}}^{-1} \mathbf{s}_{:,j}, \tag{38}$$

$$\frac{\partial \Sigma_{\mathbf{J}}}{\partial x_*^l} = -(\frac{\partial (\partial K_{\mathcal{X},*})}{\partial x_*^l})^T K^{-1} \partial K_{\mathcal{X},*} - \partial K_{\mathcal{X},*}^T K^{-1} \frac{\partial (\partial K_{\mathcal{X},*})}{\partial x_*^l}, \tag{39}$$

$$\frac{\partial^2 k(x_i, x_*)}{\partial x_*^l \partial x_*^l} = 2\rho k(x_i, x_*)(2\rho(x_i^l - x_*^l)^2 - 1),$$

$$\frac{\partial^2 k(x_i, x_*)}{\partial x_*^l \partial x_*^r} = 4\rho^2(x_i^l - x_*^l)(x_i^r - x_*^r)k(x_i, x_*).$$

Appendix C. Bayesian GPLVM metric

Following the description in section 3.2, the marginal likelihood can be derived from the the augmented joint probability as

$$p(\mathcal{S}) = \int\int\int \prod_{j=1}^{p} p(\mathbf{s}_{:}^{j}|\boldsymbol{\phi}_{:}^{j}) p(\boldsymbol{\phi}_{:}^{j}|\mathbf{u}_{:}^{j}, \mathcal{X}) p(\mathbf{u}_{:}^{j}) p(\mathcal{X})\ d\mathcal{U}\ d\Phi\ d\mathcal{X}, \tag{40}$$

where $\mathcal{S} = \{s_i | i = 1, \cdots, n+v\}$, $s_i \in \mathbb{R}^p$, is the set of the observed data points in the original space. $\mathcal{X} = \{x_i | i = 1, \cdots, n+v\}$, $x_i \in \mathbb{R}^q$ is the set of latent (unobserved) variables. The mapping is denoted as $\Phi = \{\boldsymbol{\phi}_i | i = 1, \cdots, n+v\}$, $\Phi \in \mathbb{R}^{(n+v)\times p}$ and $\boldsymbol{\phi}_i^j = \phi(x_i)^j$. The inducing points are denoted by $\mathcal{U} = \{u_i | i = 1, \cdots, m\}, \mathcal{U} \in \mathbb{R}^{m\times p}$ and $u_i = \phi(x_{ui}) \in \mathbb{R}^p$. The inducing points are evaluated at the pseudo-inputs $\mathcal{X}_u = \{x_{ui} | i = 1, \ldots, m\}$, $\mathcal{X}_u \in \mathbb{R}^{m\times q}$, in the latent space.

$p\left(\boldsymbol{\phi}^j \mid \mathbf{u}^j, \mathcal{X}, \mathcal{X}_u\right)$ and $p(\mathbf{u}^j)$ are defined as

$$\begin{aligned} p\left(\boldsymbol{\phi}^j \mid \mathbf{u}^j, \mathcal{X}, \mathcal{X}_u\right) &= \mathcal{N}\left(\boldsymbol{\phi}^j \mid K_{\mathcal{X},\mathcal{X}_u} K_{\mathcal{X}_u,\mathcal{X}_u}^{-1} \mathbf{u}^j, K_{\mathcal{X},\mathcal{X}} - K_{\mathcal{X},\mathcal{X}_u} K_{\mathcal{X}_u,\mathcal{X}_u}^{-1} K_{\mathcal{X}_u,X}\right), \\ p(\mathbf{u}^j) &= \mathcal{N}(\mathbf{u}^j | 0, K_{\mathcal{X}_u,\mathcal{X}_u}). \end{aligned}$$

We can now apply variational inference to approximate the true posterior, $p(\Phi, \mathcal{U}, \mathcal{X}|\mathcal{S}) = p(\Phi|\mathcal{U}, \mathcal{S}, \mathcal{X}) p(\mathcal{U}|\mathcal{S}, \mathcal{X}) p(\mathcal{X}|\mathcal{S})$ with a variational distribution of the form

$$q(\Phi, \mathcal{U}, \mathcal{X}) = p(\Phi|\mathcal{U}, \mathcal{X}) q(\mathcal{U}) q(\mathcal{X}) = \left(\prod_{j=1}^{p} p(\boldsymbol{\phi}_{:}^{j}|\mathbf{u}_{:}^{j}, \mathcal{X}) q(\mathbf{u}_{:}^{j})\right) q(\mathcal{X}).$$

The distribution $q(\mathcal{X})$ is chosen to be Gaussian with variational parameters for mean and variance. Using this variational distribution and the Jensen's inequality, we can derive the variational lower bound $\mathcal{F}$ of $\log p(\mathcal{S})$ the log marginal likelihood. $p(\mathcal{X})$ is chosen as Gaussian prior with identity covariance. The particular choice for the variational distribution allows us to analytically compute a lower bound.

$$\begin{aligned} \mathcal{F}(q(\mathcal{X})q(\mathcal{U})) &= \int q(\Phi, \mathcal{U}, \mathcal{X}) \log \frac{p(\mathcal{S}, \Phi, \mathcal{U}, \mathcal{X})}{q(\Phi, \mathcal{U}, \mathcal{X})} d\mathcal{X} d\Phi \partial\mathcal{U} \\ &= \hat{\mathcal{F}}(q(\mathcal{X}), q(\mathcal{U})) - \mathrm{KL}(q(\mathcal{X})||p(\mathcal{X})). \end{aligned} \tag{41}$$

Clearly, the second KL term in (41) can be easily calculated since both $p(\mathcal{X})$ and $q(\mathcal{X})$ are Gaussian. The first term in (41) can be written as

$$\begin{aligned} \hat{\mathcal{F}}(q(\mathcal{X}), q(\mathcal{U})) &= \sum_{j=1}^{p} \hat{\mathcal{F}}^j(q(\mathcal{X}), q(\mathcal{U})), \\ \hat{\mathcal{F}}^j(q(\mathcal{X}), q(\mathcal{U})) &= \int q(\mathbf{u}_{:}^{j}) \log \frac{e^{\langle \log \mathcal{N}(\mathbf{s}_{:}^{j} | K_{\mathcal{X},\mathcal{X}_u} K_{\mathcal{X}_u,\mathcal{X}_u}^{-1} \mathbf{u}_{:}^{j}, \beta^{-1} I_{n+v}) \rangle_{q(\mathcal{X})}} p(\mathbf{u}_{:}^{j})}{q(\mathbf{u}_{:}^{j})} d\mathbf{u}_{:}^{j} + \mathcal{T}, \end{aligned} \tag{42}$$

where $\mathcal{T} = \frac{\beta}{2} \operatorname{Tr}\left(\langle K_{\mathcal{X}\mathcal{X}} \rangle_{q(X)}\right) - \frac{\beta}{2} \operatorname{Tr}\left(K_{\mathcal{X}_u\mathcal{X}_u}^{-1} \langle K_{\mathcal{X}_u\mathcal{X}} K_{\mathcal{X}\mathcal{X}_u} \rangle_{q(\mathcal{X})}\right)\right)$. $\langle\cdot\rangle_{q(\mathcal{X})}$ denotes expectation under the distribution $q(\mathcal{X})$. The expression in above equation is KL-like quantity.

And $q(\mathbf{u}_:^j)$ is optimally set to be proportional to the numerator inside the logarithm of the above equation which is also a Gaussian distribution. The final expression for $\hat{\mathcal{F}}$ becomes

$$\begin{aligned}\hat{\mathcal{F}}(q(\mathcal{X})) = \sum_{j=1}^{p}\Bigg(\log\left(\int e^{\left\langle \log \mathcal{N}(\mathbf{s}_:^j|K_{\mathcal{X},\mathcal{X}_u}K_{\mathcal{X}_u,\mathcal{X}_u}^{-1}\mathbf{u}_:^j,\beta^{-1}I_{n+v})\right\rangle_{q(X)}}p(\mathbf{u}_:^j)d\mathbf{u}_:^j\right)\\ -\tfrac{\beta}{2}\operatorname{Tr}\left(\langle K_{\mathcal{X}\mathcal{X}}\rangle_{q(X)}\right)+\tfrac{\beta}{2}\operatorname{Tr}\left(K_{\mathcal{X}_u\mathcal{X}_u}^{-1}\langle K_{\mathcal{X}_u\mathcal{X}}K_{\mathcal{X}\mathcal{X}_u}\rangle_{q(\mathcal{X})}\right)\Bigg).\end{aligned} \tag{43}$$

This quantity can be computed in closed from since the computation of

$$\begin{aligned}\psi_0 &= \operatorname{Tr}\left(\langle K_{\mathcal{X}\mathcal{X}}\rangle_{q(\mathcal{X})}\right),\\ \psi_1 &= \langle K_{\mathcal{X}\mathcal{X}_u}\rangle_{q(\mathcal{X})},\\ \psi_2 &= \langle K_{\mathcal{X}_u\mathcal{X}}K_{\mathcal{X}\mathcal{X}_u}\rangle_{q(\mathcal{X})},\end{aligned}$$

are analytically computable for the squared exponential kernel. These quantities are referred to as Ψ statistics in Titsias and Lawrence (2010). The distribution of the inducing vairables are

$$\begin{aligned}q(\mathbf{u}_:^j) &= \mathcal{N}(\mu_{qu}^j,\Sigma_{qu}),\\ \mu_{qu}^j &= K_{\mathcal{X}_u\mathcal{X}_u}(\beta K_{\mathcal{X}_u\mathcal{X}_u}+\psi_2)^{-1}\psi_1^{-1}\mathbf{s}_:^j,\\ \Sigma_{qu} &= \beta K_{\mathcal{X}_u\mathcal{X}_u}(\beta K_{\mathcal{X}_u\mathcal{X}_u}+\psi_2)^{-1}K_{\mathcal{X}_u\mathcal{X}_u}.\end{aligned} \tag{44}$$

The bound can be jointly maximized over the variational parameters and the model hyperparameters by standard optimisation method such as quasi newton. The conditional probability over the Jacobian follows a Gaussian distribution.

$$\begin{aligned}p(\mathbf{J}|\mathcal{X},\mathcal{S}) &= \prod_{j=1}^{p}\mathcal{N}\left(\partial K_{\mathcal{X}_u,*}^T K_{\mathcal{X}_u,\mathcal{X}_u}^{-1}\mu_{qu}^j,\ \partial^2 K_{*,*}-\partial K_{\mathcal{X}_u,*}^T\Lambda\partial K_{\mathcal{X}_u,*}\right),\\ \Lambda &= K_{\mathcal{X}_u\mathcal{X}_u}^{-1}-K_{\mathcal{X}_u\mathcal{X}_u}^{-1}\Sigma_{qu}K_{\mathcal{X}_u\mathcal{X}_u}^{-1},\\ \frac{\partial E[\mathbf{J}^T]^j}{\partial x_*^l} &= \frac{\partial\mu_{\mathbf{J}}^j}{\partial x_*^l} = \frac{\partial(\partial K_{\mathcal{X}_u,*}^T)}{\partial x_*^l}K_{\mathcal{X}_u,\mathcal{X}_u}^{-1}\mu_{qu}^j, \qquad (45)\\ \frac{\partial\Sigma_{\mathbf{J}}}{\partial x_*^l} &= -(\frac{\partial(\partial K_{\mathcal{X}_u,*})}{\partial x_*^l})^T\Lambda\partial K_{\mathcal{X}_u,*}-\partial K_{\mathcal{X}_u,*}^T\Lambda\frac{\partial(\partial K_{\mathcal{X}_u,*})}{\partial x_*^l}. \qquad (46)\end{aligned}$$

Appendix D. Swiss roll parameterisation

The three-dimensional coordinates of the Swiss Roll can be parametrised by the radius r and the width z. Consider the Swiss roll parametrised by

$$\mathbf{x}(r, z) = (r\cos r,\ r\sin r,\ z).$$

To find its metric tensor, we first compute the partial derivatives

$$\mathbf{x}_r = (\cos r - r\sin r,\ \sin r + r\cos r,\ 0), \quad \mathbf{x}_z = (0, 0, 1).$$

The metric tensor is given by

$$(\mathbf{x}_r \cdot \mathbf{x}_r)dr^2 + 2(\mathbf{x}_r \cdot \mathbf{x}_z)dr\,dz + (\mathbf{x}_z \cdot \mathbf{x}_z)dz^2 = (1 + r^2)dr^2 + dz^2.$$

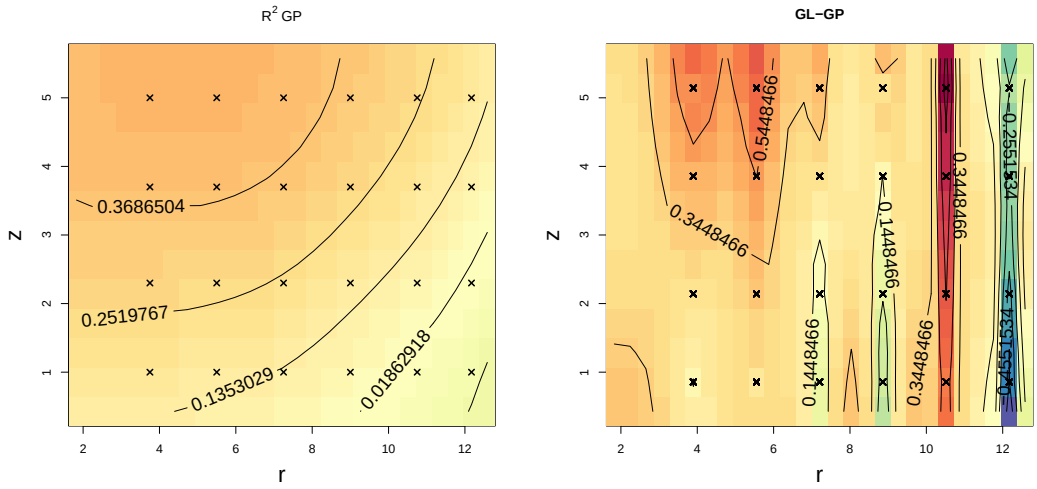
or in matrix form

$$g = \begin{bmatrix} 1+r^2 & 0 \\ 0 & 1 \end{bmatrix}, \qquad g^{-1} = \begin{bmatrix} \frac{1}{1+r^2} & 0 \\ 0 & 1 \end{bmatrix}, \qquad \frac{\partial g}{\partial r} = \begin{bmatrix} 2r & 0 \\ 0 & 0 \end{bmatrix}.$$

The determinant of the metric tensor in this case would be $1 + r^2$, as r grows the determinant is getting bigger. This indicates the exaggeration from the low dimensional latent space to the high dimensional original space is getting bigger.

The BM on the Swiss Roll can be written as

$$\begin{aligned} dr(t) &= -\frac{1}{2}\frac{r}{(1+r^2)^2}dt + (1+r^2)^{-1/2}dB_r(t), \\ dz(t) &= dB_z(t). \end{aligned} \tag{47}$$



(a) $\mathbb{R}^2$ GP (b) GL-GP

Figure 11: Prediction of $\mathbb{R}^2$GP and GL-GP

Table 4: Comparison of the root mean squared errors of three different boundary conditions on Swiss roll. Values in parentheses show the standard deviation.

	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.4$
mean RMSE	0.160(0.005)	0.163(0.003)	0.160(0.004)

To empirically evaluate the impact of boundary conditions on the performance of GPUM, we have experimented with different threshold settings in Swiss roll example. We tested three different threshold values: the original value of 0.2, and values of 0.1 and 0.4 obtained by halving and doubling the original threshold, respectively. We generated kernel estimates using BM simulations based on these three different boundary conditions and computed the regression RMSE for randomly generated datasets. The results are shown in Table 4. There are no significant differences between the results obtained using these different threshold values.

Appendix E. Estimator error

The error of the BM based estimator $\hat{K}^t$ has been discussed in Niu et al. (2019). The error consists of two parts: the numerical error and the Monte Carlo error. The Monte Carlo error arises due to the approximation of the transition probability by counting the paths that reach the neighborhood. On the other hand, the numerical error is caused by the approximation of the transition density, which involves dividing the transition probability by the volume of the neighborhood. A loss function of ω is defined by minimising the sum of two errors as described above. Specifically, for an arbitrary manifold M, one has

$$\mathcal{L}(w) = O(w^2) + O(w^{-d/2}).$$

where d is the dimension of M. These results show that there exists an optimal value of ω which can minimise the error of the estimator. But it does not provide guidance on how to choose a precise value of ω. In our numerical experiments, we notice that the fitting of GPUM is not sensitive to the choice of ω. An ablation study of the choice of ω based on the Swiss roll experiment can be found in Table 5. We conducted experiments with four different values of ω (1, 1.5, 2, and 2.5). The corresponding kernel estimates are generated from BM simulations based on these four settings. The regression RMSEs are computed for randomly generated datasets. These results show that the performance of GPUM is not very sensitive to the choice of ω.

Table 5: Comparison of the root mean squared errors of four different A_s on Swiss roll. Values in parentheses show the standard deviation.

	$\omega = 1$	$\omega = 1.5$	$\omega = 2$	$\omega = 2.5$
mean RMSE	0.176(0.009)	0.165(0.013)	0.161(0.006)	0.167(0.004)

Appendix F. WiFi signal regression

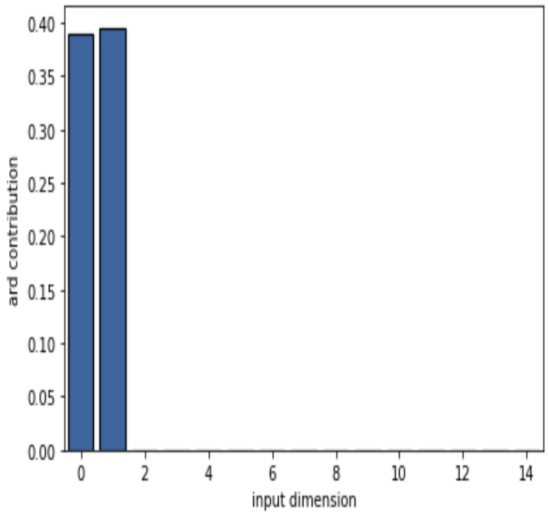


Figure 12: Automatic Relevance Determination (ARD) contributions of the WiFi signal datasets. The horizontal axis is the index of dimension. The vertical axis is the ARD contribution(or relevance) which are computed as the inverse of the squared lengthscales in B-GPLVM. If the ARD contribution is low, the corresponding dimension is less important.

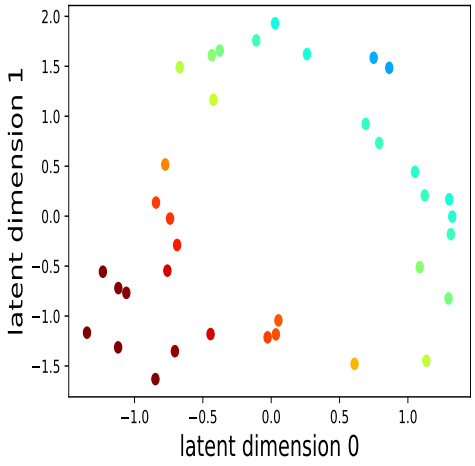
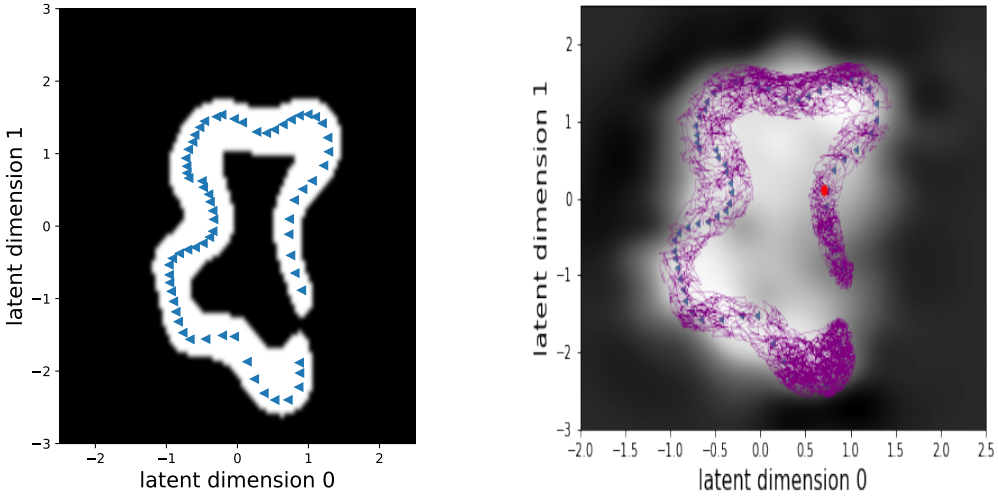


Figure 13: $\mathbb{R}^2$ GP prediction. The $\mathbb{R}^2$ GP is constructed using the euclidean distance of data points in the latent space of WiFi signals. It ignores the boundary and the magnification factor.

Appendix G. Coil image latent space



(a) COIL boundary in latent space

(b) BM trajectory

Figure 14: A BM sample path is simulated in the implicit manifold of COIL images. (a) The visualisation of the boundary in the latent space. The white region is within $\partial\mathbb{M}$. The blue triangles are the observed latent points. (b) A BM trajectory is plotted as the purple line. The red ball is the starting location. The gray background represents the magnification factor.

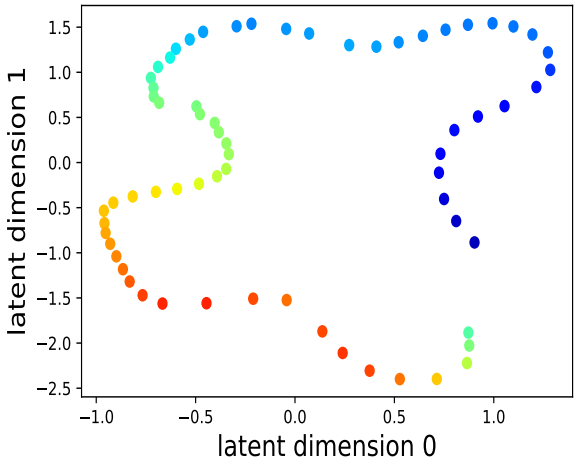


Figure 15: $\mathbb{R}^2$ GP prediction. $\mathbb{R}^2$ GP is constructed using the euclidean distance of data points in the latent space and ignoring the impact of the boundary and the magnification factor.

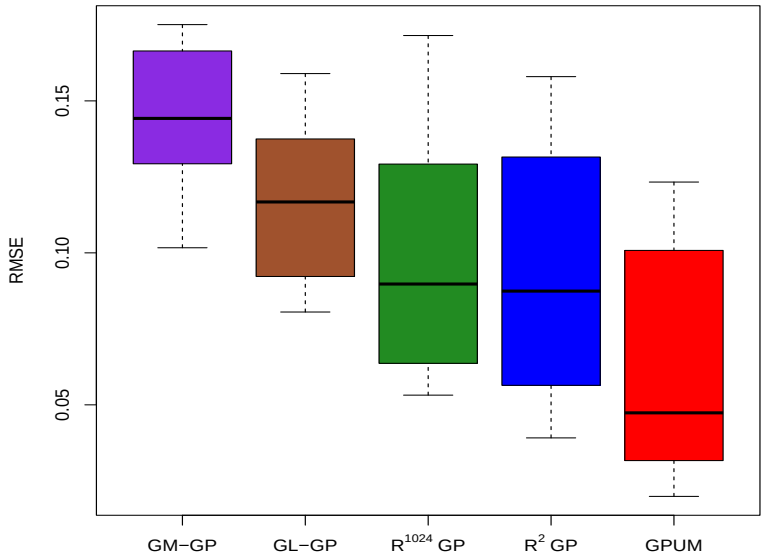
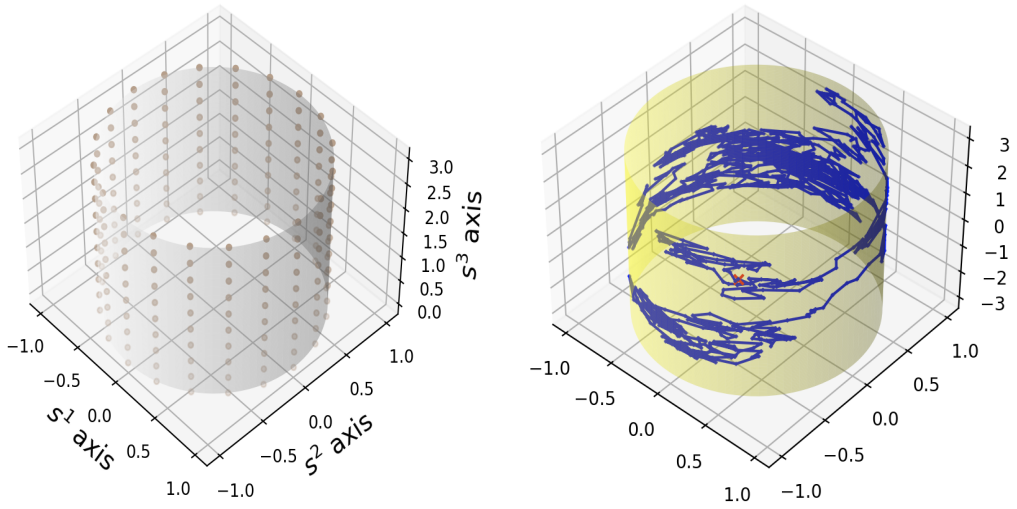


Figure 16: Boxplot of the RMSE for all methods applied in COIL images example.

Appendix H. Estimate the heat kernel of the Cylinder

The cylinder in Fig.17(a) is described by 200 points in $\mathbb{R}^3$. These points represent the union of two pre-defined datasets, $\mathcal{V}_1$ and $\mathcal{V}_2$, which overlap. Both sets constain 150 points. B-GPLVM has been applied to learn the latent spaces $(\phi_1, \mathcal{X}_1)$ and $(\phi_2, \mathcal{X}_2)$.The Brownian Motion on the cylinder is equivalent to the stochastic process in the local coordinates. The BM trajectories can be simulated using the estimated metric tensors. An example is given in Fig.17(b). Since the intersection of the two datasets is not empty, the overlapping region in the two latent spaces can also be identified. Following the method proposed in section 6.1, we can estimate the transition density of BM on the cylinder using the learned charts. Let s_0 with $s_0^1 = 0.56$, $s_0^2 = -0.83$ and $s_0^3 = 1.5$ be the starting point of the BM. $N_{BM} = 50000$ BM paths are simulated. The BM transition density is evaluated using (24) at forty target points $\{s_j \in \mathbb{M} \subset \mathbb{R}^3 | j \in 1, \cdots, 40\}$ in the observational space. These target points are equally spaced on the cylinder with fixed s^3 coordinates at 1.5. The diffusion time is fixed at 2. The analytical form of the heat kernel on the cylinder is known and can be computed as the product of the heat kernel of $\mathbb{R}^1$ and the kernel of circle S^1. It is defined as

$$K_{cylinder}(s_0, s, t) = \frac{1}{\sqrt{4\pi t}} e^{\frac{-(s_0^3 - s^3)^2}{4t}} \frac{1}{\sqrt{4\pi t}} \sum_{i=-\infty}^{\infty} e^{-\frac{[\arccos(s_0^1 s^1 + s_0^2 s^2) + 2\pi i]^2}{4t}}, \quad s \in \mathbb{R}^3. \tag{48}$$



(a) The cylinder dataset in $\mathbb{R}^3$.

(b) A BM path on the cylinder.

Figure 17: The example of the cylinder.

The true kernel values are plotted as the red solid line in Fig.18. The horizontal axis is the angular distance on the cylinder and the vertical axis is the kernel density. The heat kernel estimates using B-GPLVM metrics are plotted as the green dashed line. It is clear that the B-GPLVM results are very close to the true kernel values and match the red solid line very well. The heat kernel estimates from the Graph Laplacian (GL) approach (Dunson et al. (2020)) in different data regimes are also provided in Fig.18. When the number of

grid points on the cylinder is 200, the GL kernel estimates are plotted as the blue dashed line. It is far from the solid red line. When the number of grid points is increased to 1000 and 10000, the GL kernel estimates are plotted as the purple dashed line and black dashed line. They are getting closer to the solid red line. Comparing to the GL approach, the kernel estimates using B-GPLVM metrics achieves the better performance with much fewer points on the cylinder.

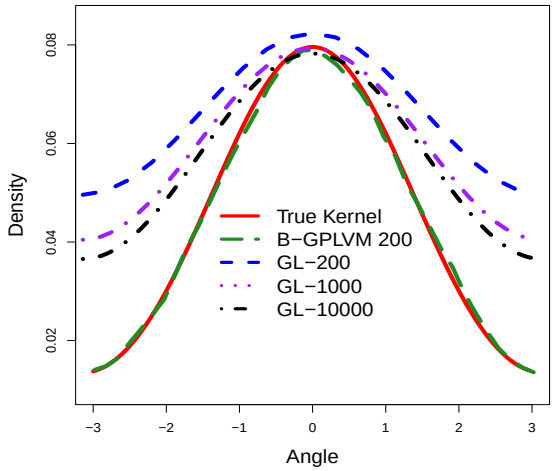


Figure 18: Comparison of heat kernel estimates using the B-GPLVM metric and Graph Laplacian. The red solid line represents the true heat kernel. The green dashed line represents estimates using B-GPLVM metric. The metric is learned from 200 grid points on the cylinder. The heat kernel estimates from the Graph Laplacian approach using the same 200 grid points are plotted as the blue dashed line. We increase the number of the points to 1000 and 10000. The GL estimates are plotted as the purple dashed line and black dashed line.

Appendix I. Graph Laplacian and Graph based Gaussian Process

Suppose $\mathbb{M}$ is a d-dimensional smooth closed and connected Riemannian manifold embedded in R^p through $f : \mathbb{M} \to \mathbb{R}^p$. Let $-\Delta$ be the Laplace-Beltrami operator of $\mathbb{M}$. Let $\{\lambda_i\}_{i=0}^{\infty}$ be the spectrum of $-\Delta$. We have eigenvalue: $0 = \lambda_0 < \lambda_1 < \ldots$. Denote ϕ_i the corresponding eigenfunction. The heat kernel has the expression:

$$H(x, x', t) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i(x) \phi_i(x').$$

Supposing we are able to recover the eigenfunctions and eigenvalues of the Laplace-Beltrami operator through the Graph Laplacian, we can recover the heat kernel via

$$H(x, x', t) = \sum_{i=0}^{\infty} e^{-\mu_{i,\epsilon}} \tilde{v}_{i,\epsilon} \tilde{v}_{i,\epsilon}^T,$$

where μ and $\tilde{v}$ are the eigenvalue and eigenvector the Graph Laplacian. Given a data set $X := \{x_1, x_2, \cdots, x_n, \cdots, x_{n+v}\}, x_i \subset R^p$, where n is the number of the labeled observation and v is the number of unlabeled grid points, we construct a kernel normalized Graph Laplacian(GL) over $x_1, x_2, ..., x_{n+v}$ following Dunson et al. (2020)'s approach.

We define a Gaussian-like kernel function:

$$k_\varepsilon(x, x') = \exp(-\frac{||x - x'||^2_{R^p}}{4\varepsilon^2}),$$

where $\varepsilon > 0$, ε is the bandwidth.

The $(n + v) \times (n + v)$ affinity matrix W is constructed using the normalised kernel $(\alpha - normalization)$, where $q_\varepsilon(x) := \sum_{i=1}^{n} k_\varepsilon(x, x_i)$:

$$W_{ij} := \frac{k_\varepsilon(x_i, x_j)}{q_\varepsilon(x_i)q_\varepsilon(x_j)} = \frac{k_\varepsilon(x_i, x_j)}{\sum_{i=1}^{n+v} k_\varepsilon(x, x_i) \sum_{j=1}^{n+v} k_\varepsilon(x, x_j)}.$$

An $(n+v) \times (n+v)$ diagonal matrix D is constructed by setting the diagonal elements as:

$$D_{ii} = \sum_{j=1}^{n+v} W_{ij}.$$

The row stochastic transition matrix A is defined as:

$$A = D^{-1}W.$$

The Graph Laplacian (GL) matrix is:

$$L := \frac{A - I}{\varepsilon^2}.$$

$\tilde{A}$ can be computed as:

$$\tilde{A} = D^{-1/2}WD^{-1/2}.$$

$\tilde{A}$ is diagonalizable, and the eigenvalue of $\tilde{A}$ is the same as A.

Given the GL matrix constructed as above, denote $\mu_{i,\epsilon}$ the i-th eigenvalue of $-L$ with the associated eigenvector $\tilde{v}_{i,\epsilon}$ normalized in l^2 norm.

We do the following normalization of the eigenvector $\tilde{v}_{i,n,\epsilon}$ in the l^2 norm. Let $N(i) = |B^{R^p}_\epsilon \cap (f(x_i))\{f(x_1) \dots f(x_n)\}|$ be the number of points on ϵ ball in the ambient space. We have the l^2 norm of $\tilde{v}$:

$$||\tilde{v}||_{l^2} = \sqrt{\frac{|S^{d-1}|\epsilon^d}{d} \sum_{i=1}^{n} \frac{\tilde{v}^2(i)}{N(i)}}.$$

We get the eigenvector after normalizing as :

$$v_{i,n,\epsilon} = \frac{\tilde{v}_{i,n,\epsilon}}{||\tilde{v}||_{l^2}}.$$

The heat kernel can be approximated as GL_{kernel}:

$$GL_{kernel} = \sum_{i=0}^{K-1} e^{-\mu_{i,\epsilon}} v_{i,\epsilon} v^T_{i,\epsilon}.$$

where K is the order of eigen-paires. The construction of the kernel can be summarised in Algorithm 2.

Algorithm 2: GL Algorithm.

Algorithm inputs include t, ϵ, K

Step (1): Construct the $(n+v) \times (n+v)$ matrix W and D as shown in Appendix I with bandwidth ϵ and points cloud $\{x_1, \ldots, x_{n+v}\}$. We can get:

$$\tilde{A} = D^{-1/2} W D^{-1/2}.$$

Step (2): Find the first $K-1$ eigenpairs of $\tilde{A}$:

$$\{\alpha_{i,\epsilon}, U_{i,\epsilon}\}_{i=1}^{K-1}.$$

Step (3): Suppose $\tilde{v}_{i,\epsilon}$ is the normalized vector of $D^{-1/2}U_{i,\epsilon}$ in the l^2 norm, and we have:

$$\mu_{i,\epsilon} := \frac{1-\alpha_{i,\epsilon}}{\epsilon^2}.$$

Let $N(i) = |B_\epsilon^{R^p}(f(x_i))\{f(x_1)\ldots f(x_n)\}|$ be the number of points on ϵ ball in the ambient space,We have the l^2 norm of $\tilde{v}$:

$$\tilde{v}_{l^2} = \sqrt{\frac{|S^{d-1}|\epsilon^d}{d} \sum_{i=1}^{n} \frac{\tilde{v}^2(i)}{N(i)}}.$$

For $i = 1, 2, \ldots, K-1$, we have: $v_{i,\epsilon} = \frac{\tilde{v}_{i,\epsilon}}{\tilde{v}_{l^2}}$

Construct $H_{\epsilon,t}^K$ as

$$H_{\epsilon,t}^K = \sum_{i=0}^{K-1} e^{-\mu_{i,\epsilon}} v_{i,\epsilon} v_{i,\epsilon}^T.$$

References

Robert J Adler. The geometry of random fields. SIAM, 2010.

Mauricio A Alvarez and Neil Lawrence. Computationally efficient convolved multiple output Gaussian processes. The Journal of Machine Learning Research, 12:1459–1500, 2011.

Theodore W Anderson. The non-central wishart distribution and certain problems of multivariate statistics. The Annals of Mathematical Statistics, pages 409–431, 1946.

Georgios Arvanitidis, Soren Hauberg, Philipp Hennig, and Michael Schober. Fast and robust shortest paths on manifolds learned from data. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 1506–1515. PMLR, 2019.

Iskander Azangulov, Andrei Smolensky, Alexander Terenin, and Viacheslav Borovitskiy. Stationary kernels and Gaussian processes on lie groups and their homogeneous spaces i: the compact case. arXiv preprint arXiv:2208.14960, 2022.

Christopher M Bishop, Markus Svensén, and Christopher KI Williams. Magnification factors for the GTM algorithm. IET, 1997.

David Bolin, Alexandre B Simas, and Jonas Wallin. Gaussian Whittle-Matérn fields on metric graphs. arXiv preprint arXiv:2205.06163, 2022.

Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, et al. Matérn Gaussian processes on riemannian manifolds. Advances in Neural Information Processing Systems, 33: 12426–12437, 2020.

Viacheslav Borovitskiy, Iskander Azangulov, Alexander Terenin, Peter Mostowsky, Marc Deisenroth, and Nicolas Durrande. Matérn Gaussian processes on graphs. In International Conference on Artificial Intelligence and Statistics, pages 2593–2601. PMLR, 2021.

Isaac Chavel. Eigenvalues in Riemannian geometry. Academic press, 1984.

Andreas C Damianou, Michalis K Titsias, and Neil Lawrence. Variational inference for latent variables and uncertain inputs in Gaussian processes. The Journal of Machine Learning Research, 17(1):1425–1486, 2016.

David Dunson, Hau-Tieng Wu, and Nan Wu. Diffusion based Gaussian processes on restricted domains. arXiv preprint arXiv:2010.07242, 2020.

Aasa Feragen, Francois Lauze, and Soren Hauberg. Geodesic exponential kernels: When curvature and linearity conflict. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3032–3042, 2015.

Brian Ferris, Dieter Fox, and Neil Lawrence. Wifi-SLAM using Gaussian process latent variable models. In IJCAI, volume 7, pages 2480–2485, 2007.

Øyvind Hjelle and Morten Dæhlen. Triangulations and applications. Springer Science & Business Media, 2006.

Elton P Hsu. A brief introduction to Brownian motion on a Riemannian manifold. Lecture Notes, 2008.

Pei Hsu. Brownian motion and Riemannian geometry. Contemporary Mathematics, 73: 95–104, 1988.

Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. Foundations and Trends® in Machine Learning, 12(4):307–392, 2019.

Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. AIChE journal, 37(2):233–243, 1991.

Neil Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. Journal of machine learning research, 6(Nov):1783–1816, 2005.

Neil Lawrence. Learning for larger datasets with the Gaussian process latent variable model. In Artificial intelligence and statistics, pages 243–250, 2007.

Lizhen Lin, Niu Mu, Pokman Cheung, and David Dunson. Extrinsic Gaussian processes for regression and classification on manifolds. Bayesian Analysis, 14(3):887–906, 2019.

Nayar and H. Murase. Columbia object image library: Coil-100. Technical Report CUCS-006-96, Department of Computer Science, Columbia University, February 1996.

Mu Niu, Pokman Cheung, Lizhen Lin, Zhenwen Dai, Neil Lawrence, and David Dunson. Intrinsic Gaussian processes on complex constrained domains. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 81(3):603–627, 2019.

Joaquin Quiñonero-Candela and Carl E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. Journal of Machine Learning Research, 6, 2005.

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

Michael E Tipping and Christopher M Bishop. Mixtures of probabilistic principal component analysers. Neural Computation, 1998.

Michalis Titsias and Neil Lawrence. Bayesian Gaussian process latent variable model. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pages 844–851, 2010.

Alessandra Tosi. Visualization and interpretability in probabilistic dimensionality reduction models. PhD thesis, Universitat Politècnica de Catalunya, 2014.

Alessandra Tosi, Søren Hauberg, Alfredo Vellido, and Neil Lawrence. Metrics for probabilistic geometries. arXiv preprint arXiv:1411.7432, 2014.

Yun Yang and David Dunson. Bayesian manifold regression. The Annals of Statistics, 44 (2):876–905, 2016.

Max Zwiessele. Bringing models to the domain: Deploying gaussian processes in the biological sciences. PhD thesis, University of Sheffield, 2017.