Solving SPDEs for Multi-Dimensional Shape Analysis

by

Yonghui Fan

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2021 by the
Graduate Supervisory Committee:

Yalin Wang, Chair
Natasha Leporé
Pavan Turaga
Yezhou Yang

ARIZONA STATE UNIVERSITY

August 2021

ABSTRACT

Statistical Shape Modeling is widely used to study the morphometrics of deformable objects in computer vision and biomedical studies. There are mainly two viewpoints to understand the shapes. On one hand, the outer surface of the shape can be taken as a two-dimensional embedding in space. On the other hand, the outer surface along with its enclosed internal volume can be taken as a three-dimensional embedding of interests. Most studies focus on the surface-based perspective by leveraging the intrinsic features on tangent plane. But a two-dimensional model may fail to fully represent the realistic properties of shapes with both intrinsic and extrinsic properties. In this thesis, several Stochastic Partial Differential Equations (SPDEs) are thoroughly investigated and several methods are originated from these SPDEs to try to solve the problem of both two-dimensional and three-dimensional shape analyses. The unique physical meanings of these SPDEs inspired the findings of features, shape descriptors, metrics, and kernels in this series of work. Initially, the data generation of high-dimensional shapes, here, the tetrahedral meshes, is introduced. The cerebral cortex is taken as the study target and an automatic pipeline of generating the gray matter tetrahedral mesh is introduced. Then, a discretized Laplace-Beltrami operator (LBO) and a Hamiltonian operator (HO) in tetrahedral domain with Finite Element Method (FEM) are derived. Two high-dimensional shape descriptors are defined based on the solution of the heat equation and the Schrödinger's equation. Considering the fact that high-dimensional shape models usually contain massive redundancies, and the demands on effective landmarks in many applications, a Gaussian process landmarking on tetrahedral meshes is further studied. A SIWKS-based metric space is used to define a geometry-aware Gaussian process. The study of periodic potential diffusion process further inspired the idea of a new kernel call the geometry-aware convolutional kernel.

A series of Bayesian learning methods are then introduced to tackle the problem of shape retrieval and classification. Experiments of each single item are demonstrated. From the popular SPDE such as the heat equation and the Schrödinger's equation to the general potential diffusion equation and the specific periodic potential diffusion equation, it clearly shows that classical SPDEs play an important role in discovering new features, metrics, shape descriptors and kernels. I hope this thesis could be an example of using interdiscipline knowledge to solve problems.

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Dr. Yalin Wang, for so much advice, encouragement, support, and understanding. I still remember the first time I contacted with him via email, the first time I came to Tempe from the freezing cold Chicago on January 1st, 2016, and the first time I met with Dr. Yalin in his office. During these five years, he taught me not only how to be a PhD student but also how to be a researcher. His attitude of doing research will accompany with me for the whole life.

Thanks to Liang Mi who gave me so many constructive suggestions since I started my PhD program. He is always patient to my silly questions and I really benefit a lot from talking with him.

Thanks to my collaborators, Dr. Natasha Leporé, Dr. Gang Wang, Dr. Qunxi Dong, Dr. Hanghang Tong, Jian Kang and Yuxiang Liu, who helped me a lot in analyzing the problem, preparing the experiments, and writing the paper. Thanks to Jianfeng Wu for the support in data processing in my last project. Thanks Yanshuai Tu and Duyan Ta for the countless useful discussions in my projects.

Szymon Stachniak and Mengxin Li made me feel at home at Microsoft Research. You were great mentors to me during my internship in Mixed Really. You brought me to the field of CV perception, VR/AR, and scene/object understanding. The amazing summer totally changed my future career direction.

Wenyuan Qi was my mentor during my internship in Canon Medical Research USA, and he demonstrated what a great researcher should be.

I'd also thank Ruocheng Guo, Jundong Li, Xu Liu, Yao Zhou, Dawei Zhou, Zhe Xu, Si Zhang, Qinghai Zhou, Kaize Ding, Lin Chen, Le Cheng, Dongqi Fu, Boxin Du, Lu Duo, Yuhan Sun, Jia Yu, Yuzhen Ding, Yanzhe Xu, Manqing Mao, Yuhao Chen, Mohan Li, Sangdi Lin, Nooshin Shomal Zadeh, Zhiyuan Fang, Xin Ye, Mohammad

Farazi, Yanxi Chen, and Zhangsihao Yang for making my days in ASU pleasant and colorful. Thanks for all the endless discussions and interactions.

Thanks to Tianyuan Ni for supporting me and for her love. Thanks to all my family members for the support and love.

My graduate study was supported by the fundings of National Institutes of Health, and the Doctorate Fellowship of Arizona State University.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Statistical Shape Modeling is widely used to study the morphometrics of deformable objects in computer vision and biomedical imaging. On one hand, the outer surface of the shape can be taken as a two-dimensional embedding in space. On the other hand, we can take the outer surface along with its enclosed internal volume as a higher dimensional embedding of interests. Most current studies focus on the surface-based perspective by investigating intrinsic features on the tangent plane. But a two-dimensional model may fail to fully represent the realistic features of shapes with both intrinsic and extrinsic properties. In this thesis, a series of methods based on solving certain SPDEs are introduced to tackle the problem of analyzing both the two-dimensional and three-dimensional shapes. The following problems are to be solved:

1. Generating the tetrahedral mesh for shapes with complicated structures. Specifically, the gray matter (GM) tetrahedral mesh is to be generated automatically;
2. Deriving the discretized LBO and HO in tetrahedral domain. Spectral analysis is one of the widely used methods in shape analysis. The prerequisite is to solve the eigen-problem of the LBO or HO, but the discretized LBO and HO on tetrahedral meshes is rarely investigated;
3. Designing distinguishable shape descriptors for high-dimensional shape models;
4. Landmarking on high-dimensional shape models. More exactly, we intend to design unsupervised landmarking via Gaussian process (GP) regressions.

5. Defining geometry-aware GPs on multi-dimensional shapes and pushing the Bayesian network on manifolds to the manner of multi-layer deep architectures.

The morphometric study of the cerebral cortex is initially taken as the main application scenario. The cerebral cortex is a highly folded anatomical region in human brain. Our motivation comes from the existing abundant evidences that cerebral disorders can greatly influence brain morphological characteristics (Lin et al. 2017; Xiao et al. 2017; Watkins, Gadian, and Vargha-Khadem 1999; Mateos et al. 2019; Wolters et al. 2019). In particular, the GM morphometry has become a popular topic of Magnetic resonance imaging (MRI) research, as GM abnormalities have been found in many neurological diseases such as Alzheimer's disease (AD) (Mateos et al. 2019; Cuingnet et al. 2011), autism spectrum disorder (Di Martino et al. 2014; Xiao et al. 2017), Parkinson's disease (Wolters et al. 2019; Kluger et al. 2019), etc. This indicates that the characterization of GM morphometry may provide discriminative features or patterns to estimate the disease status and evolution (Lin et al. 2017; Labayru et al. 2019; Watkins, Gadian, and Vargha-Khadem 1999).The accurate modeling of the GM is the prerequisite for any further analyses. The challenge lies in its complex structure. The highly folded sulci and gyri make it complicated to generate geometric error-free high-dimensional models. Besides this, reliable and consistent morphometric measurements on high-dimensional manifold-valued data are still under being investigation.

High-dimensional modeling increases the precision and performance of cortical morphometry analysis by densely sampling on the grey matters. But we also notice that the model usually contains massive redundant information which heavily increases the computational cost. A model with N points and M length point-wise features (here $N \gg M$) has a $\mathbb{R}^{N\times M}$ feature space. For guaranteeing sufficient sampling power without suffering from the high dimensionality issue in further computations, we would like to extract a subspace N' from the N points, here $N' \ll N$, while keeping M unchanged. This process of determining a significant subspace is called *landmarking*, and a selected representative point is called a *landmark*. Noting that the saliency detection and salient point selection are used as the same concepts in some literature. In this proposal, we interchangeably use both types. Clinical data analysis also supports that some diseases have more significantly effects on certain cortical regions (Thompson et al. 2003). In computer vision fields, such techniques are also widely applied to scene understanding, shape registration and face recognition etc. (Liang and Paisley 2015). Gaussian process regression has been successfully applied for extracting anatomically meaningful landmarks (Gao et al. 2018; Guhaniyogi and Dunson 2016; Liang and Paisley 2015; Rasmussen 2004; Fan et al. 2021; Fan and Wang 2020). This inspired the idea of defining an appropriate GP on GM tetrahedral models towards a more efficient and accurate cortical morphometry analysis. The most important property of a GP on manifold-valued data is the preservation of relevant morphometric properties or geometric features (Gao et al. 2018). The characterization of prior knowledge should encode these features to ensure that the conditional posterior prediction truly reflects their morphometric significance (Rasmussen 2004; Liang and Paisley 2015). However, the current methods may not fit for high-dimensional shape models because they consider less in extracting geometric features on high-dimensional manifold. As a result, a less insignificant embedding may be selected. We intend to investigate a suitable GP model on high-dimensional data.

In this PhD thesis, a series of methods are proposed towards a practical multi-dimensional shape analysis. Our contributions are summarized as following:

- An automatic pipeline to generate GM tetrahedral meshes;

- A Double Laplace-Beltrami Operator Discretization scheme (DLBOD) as a set of FEM discretization methods for the LBO eigen-problem on the tetrahedral meshes;
- Two tetrahedron-based shape descriptors, the tetrahedron-based Heat Flux Signature (tHFS) and the scale-invariant Wave Kernel Signature (WKS), as spectral analysis tools;
- Two Gaussian Processes as Bayesian inference methods to implement unsupervised landmarking on multi-dimensional manifold-valued models.
- A hierarchical Bayesian network strategy to do shape analysis for multiple types of manifolds.

Chapter 2

RELATED WORK

Speaking of shape morphometric analysis, existing methods may be generally categorized into voxel and surface based morphometry. The voxel based morphometry (VBM) is the most traditional, fundamental and direct method. An example of the voxel based morphometry (VBM) is illustrated in Fig. 2(a). For voxel based methods, shape information exists in the intensities on spatially organized grids. Such morphometric methods have the advantages of convenient data preparation and low computational cost (Ashburner and Friston 2000; Baron et al. 2001; Draganski et al. 2004; Davatzikos 2004; Bookstein 2001; Whitwell 2009; Bas-Hoogendam et al. 2017; Tuokkola et al. 2019). Deformation-based morphometry (DBM) and Tensor-based morphometry (TBM) are two representatives. The former computes the anatomical differences on deformation fields and the latter identifies structural differences from the gradients of deformation fields (Penny et al. 2011). Pattern based morphometry (PBM) is also a widely used voxel based method in neuroimaging study. It computes subject differences by extracting global patterns through dictionary learning algorithms (Gaonkar, Pohl, and Davatzikos 2011; Verma et al. 2005). The voxel is the original data type of brain scanning, therefore above methods are naturally efficient in data preparation and computation. However, such methods are limited by the fixed size of one voxel, which prohibits a higher resolution analyses. Meanwhile, neither pixel nor voxel is clinically meaningful unit and many approaches dependent on high quality preprocessings, such as the registration and normalization. The surface based morphometry (SBM) usually uses a simplicial complex, for instance a set of vertices,

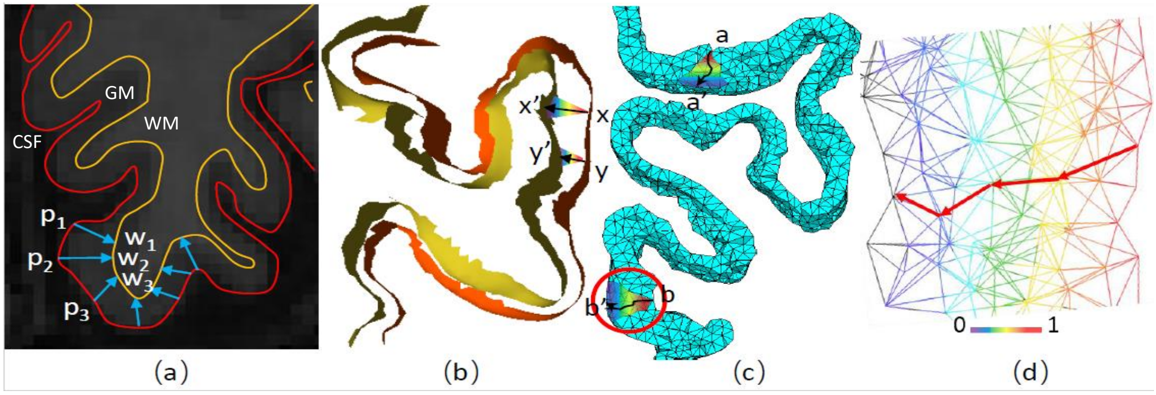


Figure 1. Cortical thickness analysis approaches. (a) Voxel-based methods. The red curve indicates the pial surface, while the yellow curve represents the grey (GM) - white matter (WM) one. P_1, P_2, P_3 are registered to W_1, W_2, W_3, respectively. The line distances of P_1-W_1, P_2-W_2 and P_3-W_3 are thicknesses. (b) Surface-based morphometry. The orange belt is the pial surface, and the yellow belt the GM-WM interface. Vertices x and y are mapped to x' and y'. Line distances x-x' and y-y' are thicknesses. (c) Tetrahedron-based volumetric morphometry. Vertex a is on pial surface and vertex b is on the GM-WM interface. Heat flow trajectories are marked as black curves from a to a' and b to b'. The lengths of trajectories are thicknesses defined on a and b. (d) Zoomed in temperature distribution. One heat flow line is shown in red and circled in red in (c).

faces and edges in a triangle mesh, to shape cerebral structures from structural MRI data. The gray matter surface model contains two boundary surfaces extracted from the segmented cortex image, as shown in Fig. 2(b):

(1) the pial surface which is the boundary between gray matter and cerebrospinal fluid (CSF), marked in orange; and (2) the white surface, which is the boundary between the GM and the white matter (WM), marked in yellow (Fischl and Dale 2000). SBM is considered to be a more precise method because of the nodal three-dimensional coordinate records and the possibility of floating point computing (J. Shi et al. 2013; Wang et al. 2012; R. Shi et al. 2017; Ge et al. 2016). Therefore, it is the current mainstream in meeting higher application demands such as the precise calculation of cortical area and curvatures, cortical parcellation, sub-cortical segmentation and

subject registrations (Xia et al. 2020; Liu et al. 2019; J. Shi et al. 2015). While SBM is sufficient to deal with the intrinsic geometry, they lack enough structural support to computing extrinsic properties. In many applications of gray matter morphometry, such as the cortical thickness estimation, the extrinsic properties are of key importance.

To combine the advantages of above two approaches while remedying their shortcomings, a volumetric modeling method seems to be a good choice, as shown in Figure. 2(c). But generating such a high-resolution geometric error-free gray matter volumetric model is still an open problem. Existing methods require global smoothing, which greatly modify the geometric properties of the original data. Smoothing is often not sufficient to solve the problem, therefore the successful meshing rate is low (G. Wang and Y. Wang 2017; Fan et al. 2018). Meanwhile, discretization methods matching with the volumetric models are also needed. Previous attempts require global smoothing, but such treatment may greatly modify the geometric properties of the original data and smoothing is often not sufficient to solve the problem. Therefore, the successful meshing rate is low and many manual adjustments are needed (G. Wang and Y. Wang 2017). Regarding to the discretization, a rough lumped method is used to the numerical analysis, which has plenty potentials to be more accurate.

The GP is one of the most important Bayesian machine learning approaches for constructing probability models in the function space (Williams and Rasmussen 2006; Fan and Wang 2020). Previous researches have successfully applied this stochastic process on 1D signals, 2D and 3D images for prediction, regression, registration and classification (Wachinger et al. 2014; Lorenzi et al. 2015). The current Gaussian process studies on manifolds mainly focus on triangle meshes (Liang and Paisley 2015; Lin et al. 2018). In a smooth Riemannian metric space, the Gaussian process is defined by classical kernels or sometimes the weighted kernels by considering well-defined geometric features, such as Gaussian curvature and mean curvature. These algorithms also yield good applications to landmarking (Gao et al. 2018; Fan, Lepore, and Wang 2020; Fan et al. 2021), motion capture (Boukhayma, Franco, and Boyer 2017), and shape recovery (Zhu, Hoi, and Lyu 2009), etc.

Classical GP kernels are the mostly used approaches, such as the Gaussian kernel, heat kernel, periodic kernel and the spectral mixture kernel. These methods work well in applications in Euclidean domain but not perform well on manifold. Many related work uses Riemannian metric and space mapping techniques to do the domain transform (Fahrmeir et al. 2013; Mallasto, Feragen, et al. 2018). The central idea of these methods is to transform the metric space to pursue a reasonable spatial measurement.

But the Euclidean metric is not totally discardable for certain data types, such as the volumetric mesh and the point cloud. Recent studies propose to add geometry-aware properties in prior knowledge learning and kernel design (**gao2019gaussian**; Prisacariu and Reid 2011; Fan, Lepore, and Wang 2020). For example, the weighted GP (W-GP) proposed in (Gao, Kovalsky, and Daubechies 2019) yields state-of-the-art posterior inferences of the topological structure by weighting the classical heat kernel with the mean curvature and the Gaussian curvature; the morphometric GP in (Fan, Lepore, and Wang 2020) uses wave kernel signature feature distances and entropy weight to define a kernel function in tetrahedral meshes. Other applications, such as the intrinsic framework in (Castillo, Kerkyacharian, and Picard 2014) and the extrinsic framework in (Lin et al. 2018), also demonstrate inspiring results after involving proper geometric measurements. However, these existing methods have considerably limitations on universality of the data type. This is because most available geometric measurements are only defined in 2-dimensional manifold and features on

3-dimensional manifold are rarely discussed. Meanwhile, the scattering effect observed in many regression applications decrease the confidence of consistency and robustness of the current mainstream. This motivates the further study on GP kernel design for manifold-based regression and classification.

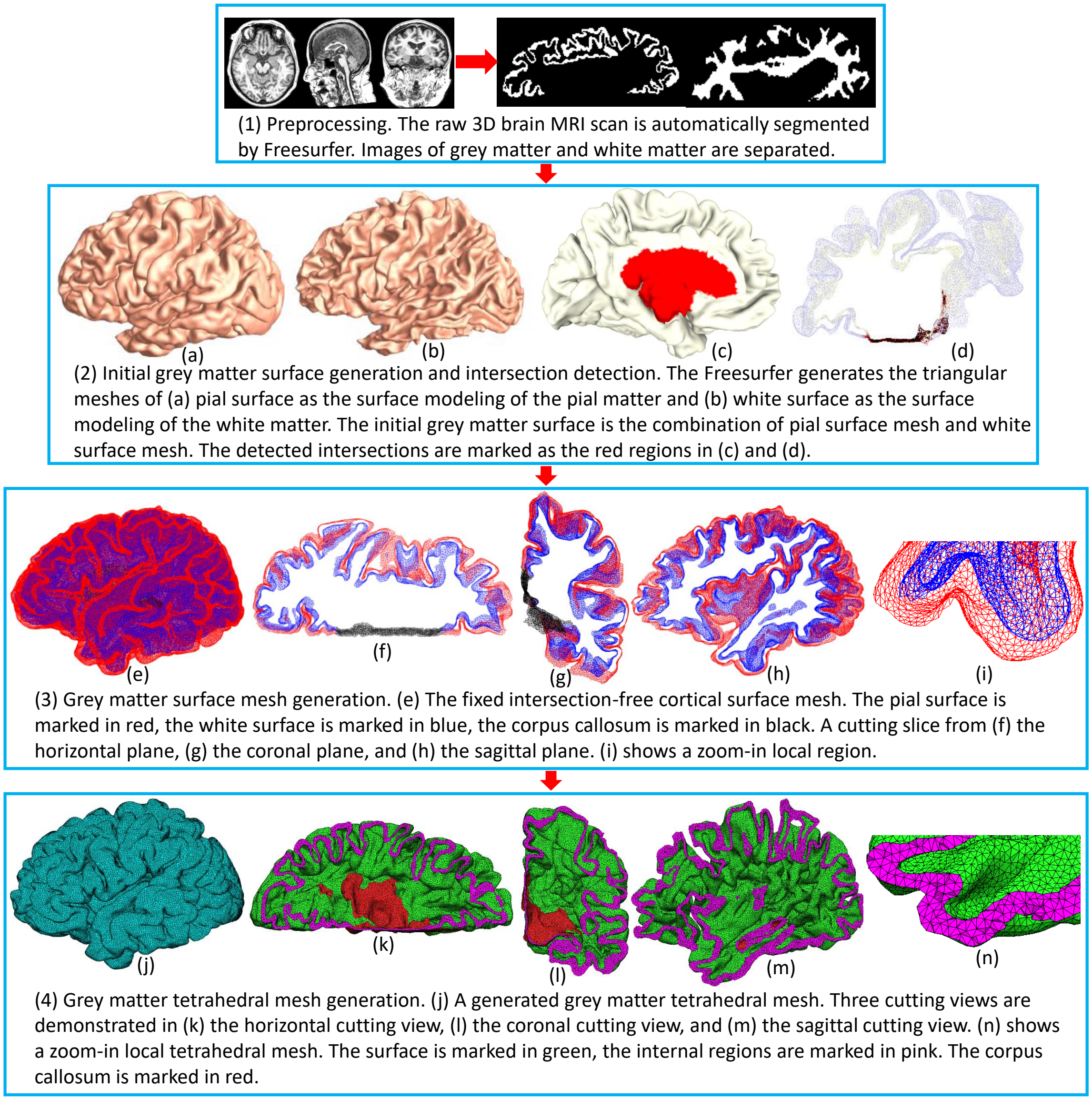
Chapter 3

TETRAHEDRAL MESH GENERATION

3.1 GM Tetrahedral Mesh Generation Pipeline

GM tetrahedral mesh generation is the process of reconstructing the GM with tetrahedrons. The purpose is to generate a computerized volumetric representation based on structural 3D MRI for further quantitative analysis. A fundamental requirement of constructing a tetrahedral mesh in our pipeline is the geometrically accurate and topologically correct boundary surface models. Fortunately, the pial and white matter surfaces of the GM can be segmented from the brain MRI and transferred to triangular meshes successfully with existing approaches. The difficult step is to combine them together as a mutual isolated composition. Even though each surface is logically and geometrically error free, the intersection error may occur during the combination because of the complex sulcal and gyral structures. For instance, one typical error is self-intersection, which is not only clinically meaningless but an impediment to generating the tetrahedral mesh.

Considering a bundle of possible geometrical errors and the mesh quality requirements of finite element analysis, we design our automatic GM tetrahedral mesh generation pipeline as follows (see also Fig 2: (1) Pre-processing. The pipeline uses Freesurfer (Fischl et al. 2002) to segment and generate the pial and white matter surfaces from brain MRI, as shown in the top row in Fig 2. (2) Marking errors. The initial surfaces must be error-free. We combine the pial and white surfaces together and detect errors. Marking all the error vertices v_e, as the red area shown in Fig 2. For



(1) Preprocessing. The raw 3D brain MRI scan is automatically segmented by Freesurfer. Images of grey matter and white matter are separated.

(2) Initial grey matter surface generation and intersection detection. The Freesurfer generates the triangular meshes of (a) pial surface as the surface modeling of the pial matter and (b) white surface as the surface modeling of the white matter. The initial grey matter surface is the combination of pial surface mesh and white surface mesh. The detected intersections are marked as the red regions in (c) and (d).

(3) Grey matter surface mesh generation. (e) The fixed intersection-free cortical surface mesh. The pial surface is marked in red, the white surface is marked in blue, the corpus callosum is marked in black. A cutting slice from (f) the horizontal plane, (g) the coronal plane, and (h) the sagittal plane. (i) shows a zoom-in local region.

(4) Grey matter tetrahedral mesh generation. (j) A generated grey matter tetrahedral mesh. Three cutting views are demonstrated in (k) the horizontal cutting view, (l) the coronal cutting view, and (m) the sagittal cutting view. (n) shows a zoom-in local tetrahedral mesh. The surface is marked in green, the internal regions are marked in pink. The corpus callosum is marked in red.

Figure 2. The pipeline for GM tetrahedral mesh generation.

example, the corpus callosum should be marked, because it does not belong to cerebral cortex. (3) Fixing. For a vertex v_e in the mesh, we define closed neighborhoods $N_p[v_e]$ on pial surfaces and $N_w[v_e]$ on white surfaces, respectively. For v in $N_p[v_e]$, we pull the vertices outwards along the normal direction by a small distance ϵ, and then do a local smoothing of $N_p[v_e]$. For v in $N_w[v_e]$, we push the vertices inwards along the inverse normal direction by ϵ and also do a local smoothing. We then record $\{N_p[v_e], N_w[v_e]\}$ in a list, because these vertices are manually modified and their future results will be adjusted or even abandoned. (4) Combining surfaces and generating tetrahedral meshes. The final step is to combine pial and white surfaces again and fill in tetrahedrons with Tetgen (Si 2015). In the forth block of Fig 2, we demonstrate one left hemisphere example and part of the entorhinal cortex tetrahedral mesh.

3.2 Experiments

In this group of experiments, two ADNI sub-datasets are employed. The first subset, ADNI2 new-package baseline imaging cohort, consists of 50 AD, 50 CU and 50 MCI patients. The second subset, ADNI2 baseline newly-visit imaging cohort, consists of 94 AD patients, 137 CU and 137 MCI patients. All subjects were scanned using 3T MRI scanners in multiple sites across the United States and Canada.

The initial MRI are the inputs to our GM tetrahedral mesh generation pipeline. To balance computation capability and accuracy, the surfaces are down-sampled to 120,000 faces.

The pipeline and classification experiments are run on an Intel(R) Core(TM) i7-4790 3.60GHz CPU and 64 GB globally addressable memory. The average processing time

Table 1. GM tetrahedral mesh Generation Information

Average Vertices Number	Average Tetrahedron Number
154040	600970
Average One Iteration Time(s)	Min/Max One Iteration Time(s)
1042.51	883.93/1217.73
Success Rate After One Iteration(%)	Success Rate After Three Iterations(%)
80.7	98.4

of computing tHFS is about 3.10 hours, including 2.49 hours of one-time tetrahedral mesh generation.

3.2.1 Tetrahedral Mesh Generation

The complex folded structures and inevitable geometric errors (self-intersections, holes etc.) are the bottlenecks of modeling the cortex with tetrahedrons. Our technical pipeline solves these problems by performing minimally invasive surgery on the erroneous structures automatically and generating the error-free mesh by TetGen. Fig.2 (4) shows a randomly-selected GM tetrahedral mesh. Fig.2 (4)(k)-(m) display horizontal, coronal and sagittal planes of (j), respectively. Fig.2 (n) is a zoomed-in figure showing the details of a small region. The information of our tetrahedral mesh generation is listed in Table 1.

Chapter 4

SOLVING SPDES FOR SHAPE DESCRIPTORS

4.1 DLBOD Scheme

Characterization of the volumetric GM morphology is performed based on the spectral analysis. We use the numerical finite element method (FEM) to derive a set of discretization formulations called double Laplace-Beltrami operator discretization (DL-BOD) scheme. As shown in Fig. 3 (c), the scheme includes: (1) tetrahedron-based heat equation and (2) tetrahedron-based LBO spectrum.

4.1.1 Tetrahedron-based Heat Equation

The tetrahedron-based heat equation solves for the interior heat distribution given proper boundary conditions. The heat distribution on the boundary is prescribed and static. We use Galerkin's method to derive a FEM discretization with Dirichlet boundary conditions (Wang, Gu, Yau, et al. 2003; Delkhosh, Delkhosh, and Jamali 2012; Shi and Chan 2010; Rustamov 2007). We firstly define the discrete harmonic energy S as:

$$S = \begin{cases} \frac{1}{12}\sum_{v_j \subseteq N(v_i)} \sum_{T_l \subseteq N(v_i,v_j)} L^{(i,j)} cot\theta_l^{(i,j)}, & if \quad v_j \subseteq N(v_i) \\ \\ 0, & otherwise \end{cases} \tag{4.1}$$

where $N(v_i)$ are the neighboring vertices of v_i; $N(v_i, v_k)$ are the tetrahedrons sharing the edge E(i,k); $L^{(i,j)}$ is the length of the opposite edge of E(i,j); and $\theta_l^{i,j}$ is

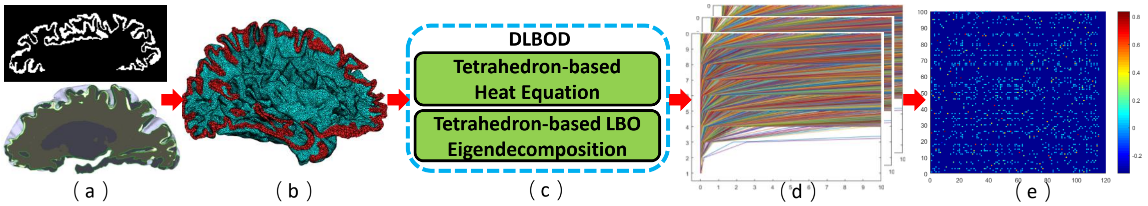

(a)
(b)
DLBOD
Tetrahedron-based Heat Equation
Tetrahedron-based LBO Eigendecomposition
(c)
(d)
(e)

Figure 3. Summarized pipeline of DLBOD scheme and tHFS computation. (a) Pre-processing; (b) Cortical tetradedral mesh generation; (c) DLBOD scheme; (d) A single tHFS; (e) Sparse grouped tHFS.

the dihedral angle of E(i,j). Taking vertex 1 in Fig. 3.3(a) as an example, $N(v_1)$ are the green points, while $N(v_1, v_k)$ are yellow edges. In Fig. 4 (b), two tetrahedrons sharing E(1,2) are colored by green. In Fig. 4 (c), E(3,8) is the opposite edge of E(1,2). The dihedral angle of E(1,2) is θ.

We extract two matrices from S: (1) W_{ii}: a square matrix with inner nodes as rows and columns; (2) W_{ib}: a matrix with inner nodes i and boundary nodes b as rows and columns. Under the Dirichlet boundary condition, the interior heat distribution H_{in} is computed by solving a linear equation:

$$\mathfrak{L}_{tet} H_{in} = H_{\partial\Omega} \tag{4.2}$$

where $H_{\partial\Omega}$ is the prescribed boundary heat. The LBO $\mathfrak{L}_{tet}$ is defined as:

$$\mathfrak{L}_{tet} = diag(\sum_i W_{ii}) + diag(\sum_i W_{ib}) - W_{ii} \tag{4.3}$$

According to Newton's law of cooling (Burmeister 1993), the discretized heat flux in direction s per unit time on vertex m is defined as a weighted heat transfer between m and a neighboring vertex n in direction s: $-k(h_n^s - h_m)$, where k is a constant; The minus sign refers to the inverse direction of the temperature gradient. The heat flow trajectory is defined as a searching path from one surface to the other along the greatest descent direction of the heat flux. The thickness is the accumulated length of all the passing edges in the trajectory. The starting vertex x and destination vertex

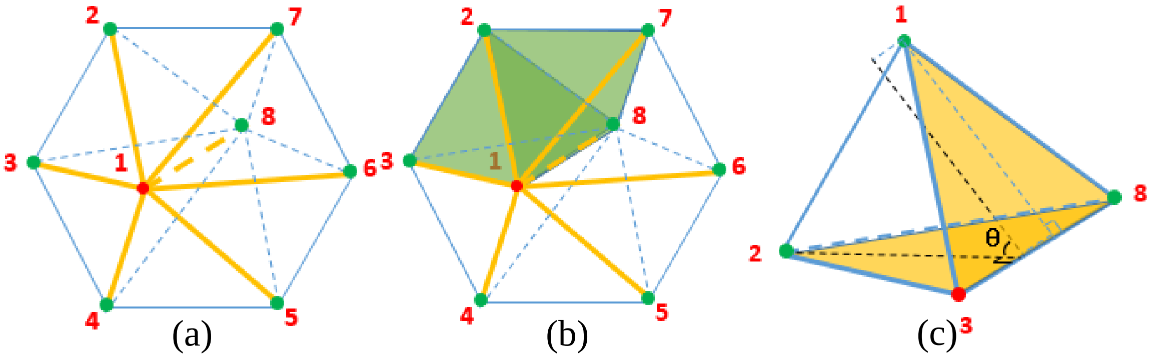


Figure 4. A discretization example on a six-tetrahedron model.

y are recorded as a surfaces point pair (x, y), which is used to define the tHFS. A heat map example is visualized in Fig. 5 We can see the interior heat level sets which represent the heat flows from outer to internal space.

4.1.2 Tetrahedron-based LBO Spectrum

The tetrahedron-based LBO spectrum (λ_n, ϕ_n) is calculated by solving the eigendecomposition: $\mathfrak{L}\phi = \lambda\phi$. A Neumann boundary condition is applied here. We define another matrix form LBO as:

$$\mathfrak{L} = B^{-1}(D - S) \tag{4.4}$$

where S is defined in Eq. 4.11, and D is the diagonal matrix defined as the sum of each row in S. We define the heat capacity matrix B as:

$$B_{i,j} = \begin{cases} \sum_{t_l \subseteq N(v_i)} \frac{|\mathbf{V}_l|}{10} + \sum_{k \subseteq N(v_i)} \sum_{t_l \subseteq N(v_i, v_k)} \frac{|\mathbf{V}_l|}{20}, & if \quad i = j \\ \sum_{t_l \subseteq N(v_i, v_j)} \frac{|\mathbf{V}_l|}{20}, \quad if \quad v_j \subseteq N(v_i) \\ 0, \quad otherwise \end{cases} \tag{4.5}$$

where $N(v_i)$ is the neighboring tetrahedrons of v_i; $N(v_i, v_k)$ is the neighboring tetrahedrons of $E(i, k)$.

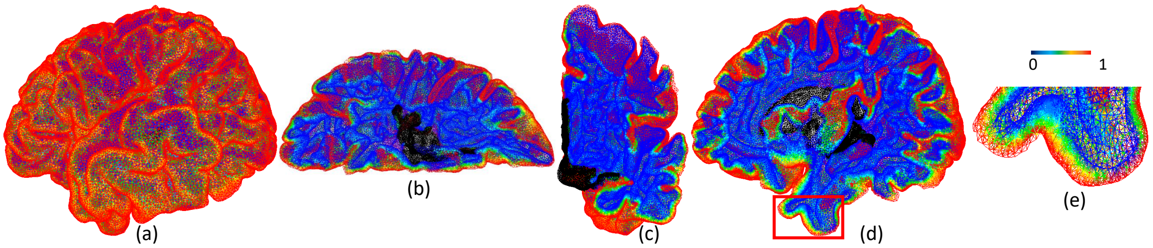


Figure 5. Tetrahedral heat map. The pial surface has the high heat and the white surface has the low heat. (a) Example of the heat map. (b)-(d) Three cutting planes of the volumetric heat map. A zoomed in figure is illustrated in (e).

4.1.3 Tetrahedral Heat Flow Signature

4.1.3.1 Single tHFS

We build the tHFS based on the results of DLBOD scheme. With point pair (x, y) and the spectrum (λ_n, ϕ_n), the single tHFS is defined as:

$$tHFS(x, y|t_0 + \delta t) = \sum_{n}^{N} \frac{\phi_n(x)'\phi_n(y)}{e^{\lambda_n(t_0+\delta t)}} \tag{4.6}$$

where t_0 is an initial constant; step size δt is defined by:

$$\delta t = \frac{2}{0.8 \times \lambda_{max}} \tag{4.7}$$

The tHFS is considered to be a collection of time series vectors, regarding δt as a time interval and t_0 as the initial time as shown in Fig. 3(d).

4.1.3.2 Grouped tHFS

The single tHFS defines a pointwise cortical thickness pattern in matrix form. But the matrix form is inconvenient when solving a groupwise problem. Therefore, we define a vector form by concatenating the principal components of the tHFS matrix. The individual tHFS is now a vector G_P with length P. In our experiments, the P is 50. Furthermore, a grouped tHFS matrix $G_{N\times P}$ is defined by piling up the N tHFS vectors coming from the same group. The group tHFS is an over-complete feature matrix and it is further expressed in a sparse way by stochastic coordinate coding (Liu, Ji, Ye, et al. 2009). A general process is:

(1) Initialize dictionary D_1^1. Partition the input matrix $G_{N\times P}$ into several segments by setting time slots, $G_{N\times P} = (g_1, g_2, ..., g_q)$, where q is the total number of segments. Set sparse code $z_i^0 = 0$ for all segments.

(2) For each segment, update the sparse code by coordinate descent:

$$z_i^k \leftarrow CoordinateDescent(D_i^k, z_i^{k-1}, g_i) \tag{4.8}$$

(3) Update the Hessian matrix and learning rate η:

$$H \leftarrow H + z_i^k (z_i^k)^T, \eta_{i,j}^k = \frac{1}{H(j,j)} \tag{4.9}$$

(4) Update the dictionary D by using stochastic gradient descent:

$$D_{i+1}^k = P_{B_m}(D_i^k - \eta_i^k \bigtriangledown_{D_i^k} f_i(D_i^k, z_i^k)) \tag{4.10}$$

where P is the projection operator, η is the learning rate, estimated by the inverse of the j^{th} diagonal element of Hessian matrix.

(5) Repeat (2)(3)(4) until all segments are processed. The final group tHFS is a sparse matrix with the number of subjects as the row dimension and feature length as the column dimension.

4.1.4 Tetrahedron-based HO, Time-independent Schrödinger's Equation and Scale-invariant Wave Kernel Signature

We employs the Schrödinger's Equation to define a tetrahedron-based scale-invariant WKS (SIWKS) as a spectral shape descriptor. The Schrödinger's equation describes the behavior of quantum mechanical particles, and it is the theoretical basis of the WKS (Aubry, Schlickewei, and Cremers 2011). In this scenario, we assume the cortex is a quantum system and every point in a cortical shape is taken as a particle according to the theory of classical mechanics. Any shape changes are simulated to be the spatial movement of particles over a period of time. Their movements are not determined beforehand but their static destination at a location x is measured by a probability distribution which is predicted by the time-independent Schrödinger's equation:

$$\mathfrak{H}\Phi(x) = \Lambda\Phi(x) \tag{4.11}$$

where $\Phi(x)$ is the wave equation, Λ is the total energy of the system, and $\mathfrak{H}$ is the HO:

$$\mathfrak{H} = -\frac{\hbar^2}{2}\sum_{d=1}^{D}\frac{1}{m_d}\nabla_d^2 + P \tag{4.12}$$

where $\hbar$ is the Planck constant, m is the particle mass, dimentionality D is 3; P is a potential energy function. Eq. (4.11) indicates that the time-independent Schrödinger's equation is essentially the spectra problem of the Hamiltonian operator (HO). Define the discrete harmonic energy S as in our prior work (Wang, Gu, Yau, et al. 2003):

$$S = \begin{cases} \frac{1}{12}\sum_{t\subseteq N_t(v_i,v_j)} L^{(i,j)} cot\theta_l^{(i,j)}, & if \quad v_j \subseteq N_v(v_i) \\ \\ 0, & otherwise \end{cases} \tag{4.13}$$

$L^{(i,j)}$ is the length of the opposite edge of $e(i,j)$, and $\theta_l^{i,j}$ as the dihedral angle of $e(i,j)$. Taking the vertex 1 in a six-tetrahedron model shown in Fig. 4 (a-b) as an example,

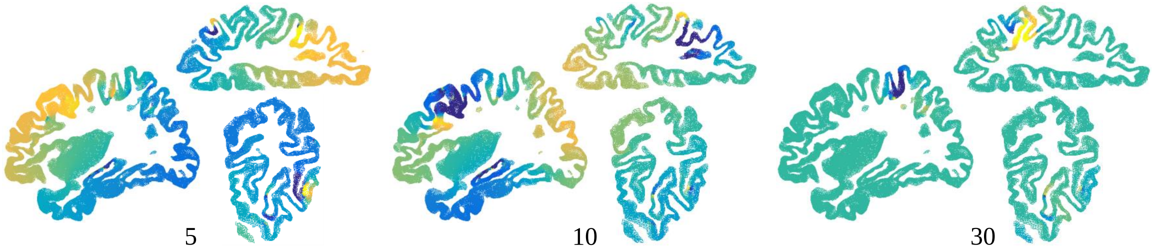


Figure 6. Visualizations of the 5^{th}, 10^{th}, and 30^{th} eigenvectors on three anatomical cutting planes.

$N_v(v_1)$ is the set of green spheres, and $N_t(1,8)$ is the set of all six tetrahedrons. In Fig. 4 (b), two tetrahedrons sharing $e(1,8)$ are shown in green. In Fig. 4 (c), around $e(1,2)$, $e(3,8)$ is its opposite edge and θ is its dihedral angle. Eq. (4.11) defines a $|\mathcal{V}| \times |\mathcal{V}|$ symmetric matrix. Then, we derive a mass matrix B from the tetrahedron-based barycentric coordinate system:

$$B_{i,j} = \begin{cases} \sum_{t\subseteq N_t(v_i)} \frac{|\mathbf{V}|}{35} + \sum_{k\subseteq N_t(t)} \sum_{t\subseteq N_t(v_i,v_k)} \frac{|\mathbf{V}|}{140}, \quad if \quad i=j \\ \sum_{k\subseteq N_t(f)} \sum_{f\subseteq N_f(v_i,v_j)} \frac{|\mathbf{V}|}{420} + \sum_{k\subseteq N_t(t)} \sum_{t\subseteq N_t(v_i,v_j)} \frac{|\mathbf{V}|}{840}, \quad if \quad v_j \subseteq N_v(v_i) \\ 0, \quad otherwise \end{cases} \tag{4.14}$$

where $|\mathbf{V}_l|$ is the volume of the l^{th} tetrahedron. The derivation of coefficients in Eq. 4.14 is given in the Appendix. Finally, the HO with Neumann boundary condition is defined as:

$$\mathfrak{H} = B^{-1}(G - S), \quad G = diag(\sum_i S_{ii}), \tag{4.15}$$

where G is a diagonal matrix of the sum of each row in S. When the potential energy is zero, we can numerically estimate the HO to be same as the LBO given the same boundary condition. Here, stressing the HO instead of LBO is because the wave kernel signature is built on the theorem of wave equation and achieved great success as an effective shape descriptor (Aubry, Schlickewei, and Cremers 2011). By solving Eq. (4.15), we obtain K smallest positive eigenvalues $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_K\}, \lambda > 0$ and the corresponding eigenfunctions $\Phi = \{\phi_1, ..., \phi_K\}$ (the first eigenvalue and its eigenvector are excluded). In Fig. 6, we can see examples of the 5^{th}, 10^{th}, and 30^{th} eigenvectors on the sagittal, horizontal and coronal cutting planes of a GM tetrahedral mesh. Supposing that the mesh $\mathcal{M}$ has a scaled version $\beta\mathcal{M}$, the eigenvalues and eigenvectors have the corresponding scaled values: $\lambda_{\beta\mathcal{M}} = \beta^2\lambda_{\mathcal{M}}$, $\phi_{\beta\mathcal{M}} = \beta\phi_{\mathcal{M}}$. Therefore, multiplying by an inversed eigenvalue will make the WKS scale-invariant (**li2018scale**). The tetrahedron-based scale-invariant WKS (SIWKS) of vertex v is defined as:

$$SIWKS(v,\epsilon) = \frac{1}{\lambda_K} C_\epsilon \sum_{k}^{K} \phi_k^2(v) e^{-\frac{(\epsilon - log\lambda_k)^2}{2\sigma^2}}, C_\epsilon = \left(\sum_{k}^{K} e^{-\frac{(\epsilon - log\lambda_k)^2}{2\sigma^2}} \right)^{-1} \tag{4.16}$$

where ϵ is a evenly spaced vector of the energy scale between $\epsilon_{min} = log(\lambda_1)$ and $\epsilon_{max} = log(\lambda_K)$. The length of ϵ is the length of the SIWKS feature vector. λ_K is the selected largest eigenvalue, hence its inverse is taken as a normalization of the SIWKS. The SIWKS distance map is defined using the accumulated absolute SIWKS differences as a feature-based metric:

$$M(v_i, v_j) = \begin{cases} \sum_{\epsilon_{min}}^{\epsilon_{max}} \left| \frac{SIWKS(v_i,\epsilon) - SIWKS(v_j,\epsilon)}{SIWKS(v_i,\epsilon) + SIWKS(v_j,\epsilon)} \right|, & j \in N(v_i), \\ 0, & i = j \quad or \quad j \notin N(v_i). \end{cases} \tag{4.17}$$

In implementations, we use K-Nearest Neighbor (KNN) algorithm to get the $N(v_i)$, while we simply use the connection information to get $N(v_i)$ in constructing the harmonic energy and the mass matrix. It is clear that M is a $|\mathcal{V}| \times |\mathcal{V}|$ sparse matrix with $M(v_i, N(v_i))$ as the only nonzero entries. The sparsity is determined by the K in KNN. We demonstrate a SIWKS distance map in Fig. 7. Since we reorganize the vertex orders by placing boundary vertices ahead of internal vertices, the distance map shows a more sparse pattern in the top left region than in the other regions. We will use this SIWKS distance map in our Bayesian method to replace the Euclidean

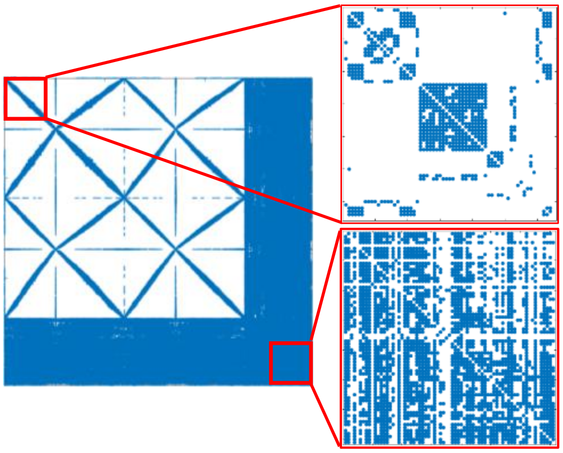


Figure 7. A SIWKS distance map is a sparse matrix. Two regions are zoomed in.

distance. This substitution emphasizes the importance of similarities in geometric features instead of the spatial positions, which helps to make reasonable posterior inferences in a Bayesian model.

4.2 Experiments

4.2.1 GM tetrahedral mesh Spectrum Visualization

The visualization of GM tetrahedral mesh spectrum is solved by (1) Cutting the cortex according to coronal, horizontal and sagittal planes. (2) Scaling the values of the eigenvectors and rendering the color map based on the rescaled values in each eigenvector. In experiments, we selected four eigenvectors, the first, fifth, tenth and thirtieth eigenvector, for the visualization. The results are shown in Fig. 6. We selected cross-section views from all three planes. We tried to choose slices passing through some key regions, such as the posterior cingulate and superior temporal and precuneus regions, which are considered to be closely connected with AD progression.

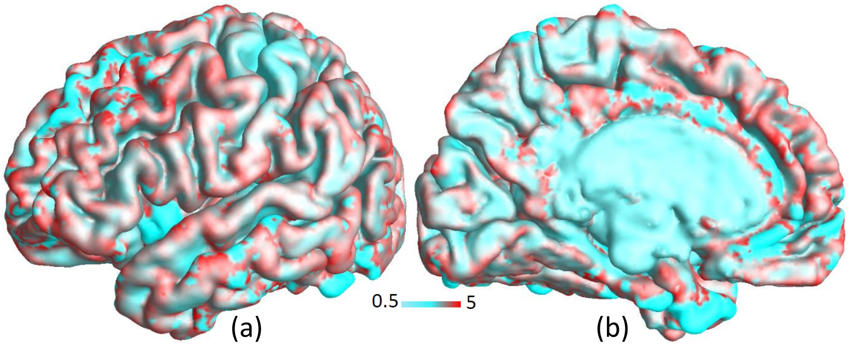


Figure 8. An illustration of the cortical thickness ranging from 0.5 to 5 (mm).

Color changes within the same region reflect the rich spectrum features under different orthogonal basis.

4.2.2 Cortical Thickness Map

The direct definition of the cortical thickness is not numerically tested in this paper, as tHFS is a more powerful measurement of morphological properties, but we show a standard thickness map in Fig. 8. The figure is rendered by scaling, with the thicknesses ranging from 0.5 to 5 (mm).

4.2.3 Cortical Atrophy Severity Classifications

The two comparison methods are: (1) the surface-based thickness morphometry method used in FreeSurfer (Fischl and Dale 2000); (2) the lumped tetrahedron-based volumetric method in (G. Wang and Y. Wang 2017), whose LBO is defined as:

$$L_p = D^{-1}S, \quad D = diag(d_1, ..., d_n). \tag{4.18}$$

Table 2. Classification results of the first dataset

GROUP		15: Lumped	15: tHFS	25: Lumped	25: tHFS	FreeSurfer
AD-CU	ACC	0.847	0.855	0.885	0.897	0.783
	SEN	0.913	0.877	0.932	0.922	0.692
	SPE	0.761	0.826	0.818	0.864	0.9
AD-MCI	ACC	0.836	0.848	0.857	0.884	0.739
	SEN	0.895	0.896	0.904	0.915	0.818
	SPE	0.761	0.784	0.795	0.841	0.667
CU-MCI	ACC	0.809	0.852	0.835	0.844	0.696
	SEN	0.835	0.862	0.849	0.814	0.733
	SPE	0.783	0.842	0.823	0.875	0.625
GROUP		30: Lumped	30: tHFS	40: Lumped	40: tHFS	FreeSurfer
AD-CU	ACC	0.917	0.941	0.931	**0.951**	0.783
	SEN	0.939	**0.965**	0.957	**0.965**	0.692
	SPE	0.886	0.909	0.898	**0.932**	0.9
AD-MCI	ACC	0.896	0.916	0.914	**0.926**	0.739
	SEN	0.913	**0.939**	0.915	0.930	0.818
	SPE	0.875	0.886	0.909	**0.920**	0.667
CU-MCI	ACC	0.887	0.913	0.913	**0.939**	0.696
	SEN	0.878	0.925	0.898	**0.948**	0.733
	SPE	0.896	0.903	0.929	**0.930**	0.625

d_n is the weighted volume sum of all the tetrahedrons sharing vertex i:

$$d_n = \sum_{T_l \in N(i)} \mathbf{V}(t_l)/4 \tag{4.19}$$

We perform classifications using a 10-fold cross validation. Accuracy (AC), sensitivity (SEN) and specificity (SPE) are computed for performance evaluation. For the first dataset, we use AdaBoost (Freund, Schapire, and Abe 1999) as the classifier and 25-step tHFS. Performance increases through selecting 15, 25, 30 and 40 eigenfunctions are shown in Table 2. The results of using the second datasets with the same setting are shown in Table 3. On average, tHFS outperforms the lumped method by 1%-2.4% and the FreeSurfer method by 11%-17% in classifying the first dataset. The accuracy converges to about 95% after using 40 eigenfunctions.

Table 3. Classification results of the second dataset

GROUP		15: Lumped	15: tHFS	25: Lumped	25: tHFS	FreeSurfer Thickness
AD-CU	ACC	0.792	0.818	0.795	0.818	0.762
	SEN	0.861	0.913	0.809	0.817	0.770
	SPE	0.705	0.693	0.783	0.818	0.785
AD-MCI	ACC	0.709	0.767	0.739	0.768	0.674
	SEN	0.80	0.852	0.843	0.852	0.737
	SPE	0.591	0.659	0.602	0.659	0.776
CU-MCI	ACC	0.774	0.822	0.796	0.835	0.534
	SEN	0.739	0.861	0.835	0.861	0.682
	SPE	0.818	0.783	0.757	0.809	0.716
GROUP		30: Lumped	30: tHFS	40: Lumped	40: tHFS	FreeSurfer Thickness
AD-CU	ACC	0.828	0.853	0.927	0.941	0.762
	SEN	0.748	0.8	0.948	0.965	0.770
	SPE	0.932	0.920	0.898	0.909	0.785
AD-MCI	ACC	0.785	0.793	0.886	0.894	0.674
	SEN	0.8	0.8	0.887	0.904	0.737
	SPE	0.761	0.784	0.886	0.875	0.776
CU-MCI	ACC	0.870	0.874	0.861	0.869	0.534
	SEN	0.791	0.765	0.896	0.904	0.682
	SPE	0.948	0.983	0.826	0.826	0.716

Chapter 5

SOLVING SPDES FOR GAUSSIAN PROCESSES

5.1 Morphological Gaussian Processes

Gaussian process (GP), especially Gaussian process regression (GPR), is widely used in spatial inference problems known as the “Kriging” (**gao2019gaussian**; Rasmussen 2004). Suppose $V_n = \{v_1, ..., v_n\}$ is the set of n training samples and $Y_n = \{y(v_1), ..., y(v_n)\}$ is the set of corresponding observations. The classical GPR learns a multivariate distribution to fit the training data and predict the observation $y(v_{n+1})$ when a testing sample v_{n+1} comes. The Bayes theorem is used to infer the posterior prediction from the prior knowledge. One important property of GP is that every finite marginal distribution of a GP still follows a multivariate Gaussian distribution $\mathcal{GP}(m_{n+1}, K_{n+1})$ and the new distribution is explicitly determined by:

$$m_{n+1} = K_{n,n+1}^T K_{n+1,n+1}^{-1} Y_n \tag{5.1}$$

$$K_{n+1} = K_n - K_{n,n+1}^T K_{n+1,n+1}^{-1} K_{n,n+1} \tag{5.2}$$

$$K_{n,n+1} = K_{n+1,n}^T = \begin{pmatrix} K(v_1, v_1) & \cdots & K(v_1, v_n), & K(v_1, v_{n+1}) \\ \vdots & & & \vdots \\ K(v_n, v_1) & \cdots & K(v_n, v_n), & K(v_n, v_{n+1}) \end{pmatrix} \tag{5.3}$$

$$K_{n+1,n+1} = \begin{pmatrix} K(v_1, v_1) & \cdots & K(v_1, v_n), & K(v_1, v_{n+1}) \\ \vdots & & & \vdots \\ K(v_n, v_1) & \cdots & K(v_n, v_n), & K(v_n, v_{n+1}) \\ K(v_{n+1}, v_1) & \cdots & K(v_{n+1}, v_n), & K(v_{n+1}, v_{n+1}) \end{pmatrix} \tag{5.4}$$

Motivations of choosing GPs: Despite the commonly-mentioned advantages of GP regression and Bayesian inference in spatial data analysis, we summarize two reasons of choosing GPR as the framework in our applications: (1) for implementations on large-scale data, GP regression is capable of working with active learning strategy (Liang and Paisley 2015; Gao, Kovalsky, and Daubechies 2019). At each time step, only one vertex is selected and one dimension is added to the kernel matrix after the current extraction. This consideration has a significant acceleration effect in computational efficiency; (2) GP is the best tool we can find to construct a process that allows for the geometric dependence varying as a function of the distance lags. A stationary kernel function is a perfect platform for defining local spatial features.

5.1.1 Scale-Invariant Wave Kernel Signature Distance Map

Comparing with several matured geometric feature descriptors, WKS is more powerful in depicting local features but it is initially scale-variant (Aubry, Schlickewei, and Cremers 2011). Here, we propose a scale-invariant WKS. It is known that WKS is rooted in the Schrödinger Equation. Under an estimation circumstance, the Schrödinger Equation is approximated to be the Neumann eigenvalue problem of LBO. We apply the discretized solution on tetrahedral meshes mentioned in (Fan et al. 2018) to obtain K pairs of eigenfunctions $\Psi = \{\psi_1, ..., \psi_K\}$ and non-negative eigenvalues $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_K\}$. Then a scale-invariant WKS is defined as:

$$W = \frac{1}{\lambda_K} C_e \sum_i^K \psi_i^2 e^{-\frac{(e_i - log\lambda_i)^2}{2\sigma^2}}, C_e = \left(\sum_i^K e^{-\frac{(e_i - log\lambda_i)^2}{2\sigma^2}} \right)^{-1} \tag{5.5}$$

where ϵ is a evenly spaced vector of the energy scale between $log(\lambda_1)$ and $log(\lambda_K)$. We use the same setting of ϵ and σ with those in (Aubry, Schlickewei, and Cremers 2011).

Lemma 1. *WKS defined by Eq. 5.5 is scale-invariant.*

Proof. Suppose that there exists a scaled tetrahedral model M' of M with the relationship: $M' = \alpha M$, where α is a constant. If M has the eigenvalues λ and eigenfunctions ψ, then the eigenvalues and eigenfunctions of M' are $\lambda' = \alpha^2\lambda$ and $\psi' = \alpha\psi$. Substituting the above eigenvalues and eigenfunctions in Eq. (5.5) to calculate the WKS of M', W'. We can see that $W' = W$ and the constant α has no effect to the results. Therefore, Eq. (5.5) defines a scale-invariant WKS. □

The scale-invariant WKS distance map is then defined as:

$$S = \begin{cases} \sum_e \left| \frac{W(v_i) - W(v_j)}{W(v_i) + W(v_j)} \right|, & j \in N(v_i) \\ 0, & i = j \quad or \quad j \notin N(v_i) \end{cases} \tag{5.6}$$

where $N(v_i)$ is the n neighboring vertices of v_i. $N(v_i)$ is generated by K-Nearest Neighbor (KNN) algorithm. It is clear to see that S is a $|V| \times |V|$ sparse matrix with $S(v_i, N(v_i))$ as the only nonzero entries. In designing the covariance function of M-GP, we replace the Euclidean distance of the Cartesian coordinates with this feature space distance. This choice emphasizes the similarities of geometric features but the spatial positions, which fits better with the expectations. A visualization of the four dimensions of the spectrum is shown in Fig. 9.

5.1.2 Heat Flow Entropy

Assuming the cortex is a thermodynamic equilibrium system, the internal temperature distribution, as shown in Fig. 3.4, is resolved by the heat equation with Dirichlet boundary condition: $\mathfrak{L}_{\mathfrak{tet}} H_{in} = H_{\partial M}$, where H_{in} is the desired interior temperature distribution; $H_{\partial M}$ is the prescribed temperature on two boundary surfaces. The pial

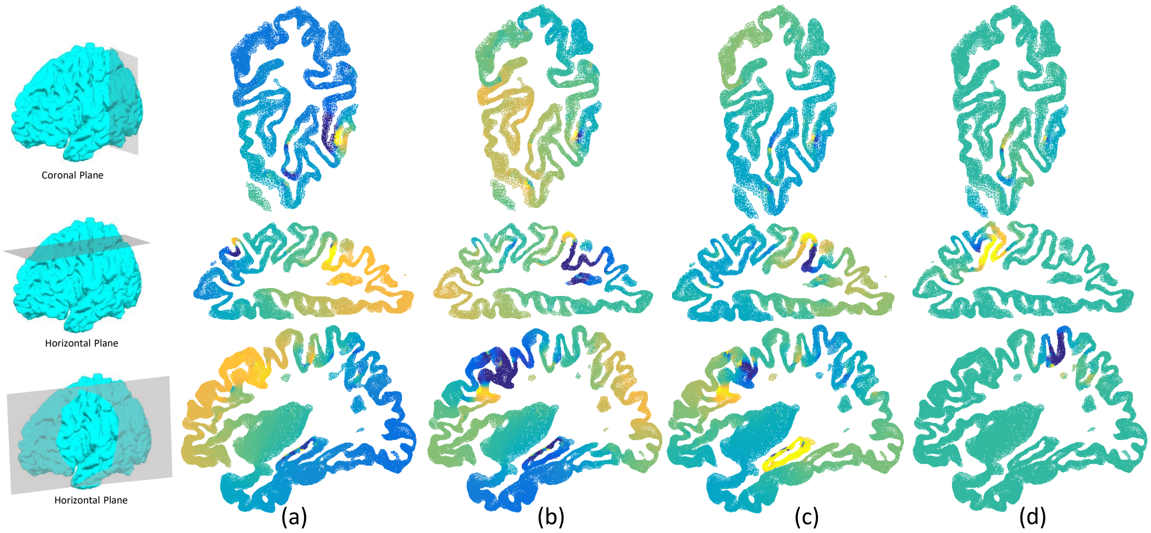


Figure 9. Examples of the spectrum of a randomly selected GM tetrahedral model.

surface has a high temperature 1 and the white surface has a low temperature 0; The LBO $\mathfrak{L}_{\mathfrak{tet}}$ is specified by using the method in (Fan et al. 2018). There are several heat flow tendencies denoted as degrees of freedom on each vertex. Each degree comes with a heat transition probability. The heat flow on v_i in direction s per unit time is defined by the weighted temperature variation between v_i and the neighbors along s. In tetrahedral meshes, this is defined by the weighted Newton's law of cooling: $-k(h^s - h_i)$. Here h is the temperature, the conductivity k is set to be 1. The minus sign refers to the inversed direction of the temperature gradient. An HFE is defined to measure the amount of thermodynamics information:

$$H_{HFE}(v_i) = -\sum_{j}^{J} \frac{|h_j - h_i|}{\sum_j |h_j - h_i|} log\left(\frac{|h_j - h_i|}{\sum_i |h_j - h_i|}\right), \quad v_j \in N(v_i) \tag{5.7}$$

where J indicates the total degrees of the freedom of v_i. On a vertex, J equals to the number of edges connected to this point. The HFE is a positive scalar function inspired by the Shannon entropy. The meanings of HFE is also interpretable from the view of information theory. Furthermore, HFE is not only a statistical measurement of

Algorithm 1 Landmark Selection based on M-GP regression

```
procedure M-GP($M$,k)                      ▷ Tetrahedral Model $M$, number of landmarks k
    $W \leftarrow$ Scale-invariant WKS     ▷ calculate scale-invariant WKS with Eq. 5.5
    $N(V) \leftarrow$ K-Nearest Neighbors  ▷ calculate $n$ nearest neighbors of each vertex
    $S \leftarrow$ WKS Distance Map        ▷ calculate the WKS distance map with Eq. 5.6
    $H \leftarrow$ Heat Flow Entropy       ▷ calculate the diagonal matrix of HFE with Eq. 5.7
    $K \leftarrow$ Covariance function     ▷ calculate the covariance function with Eq. 5.8
    $\tilde{V} = \left\{v^1, ..., v^k\right\} \leftarrow \varnothing$   ▷ initialize the set of landmarks
    $D \leftarrow diag(K)$                 ▷ get the initial uncertainty score
    while $i \leq k$ do
        if i=1 then
            $\Sigma_M(v_i) \leftarrow D$
        else
            $\Sigma_M(v_i)$                ▷ update uncertainty score with Eq. A.3,A.4
        end if
        $v^i \leftarrow argmax \Sigma_M$   ▷ select a landmark with Eq. A.5
        $\tilde{V} \leftarrow v^i$         ▷ save new landmark to $\tilde{V}$
        $i \leftarrow i + 1$
    end while
    return $\tilde{V}$
end procedure
```

a thermodynamic system but also a valid dimension-free geometric feature descriptor. The dimension-free means HFE is feasible on any dimensional manifolds.

Lemma 2. *The heat flow equals to curvature flow. Heat flow entropy is a point-wise dimension-free measurement of the amount of curvature flow.*

The proof is given in Appendix C. Lemma 3 states that the curve flow is analogous to the heat flow. On a 3-dimensional manifold, such as a tetrahedral mesh, the curve function is a 2D function and the curvature flow can be regarded as the level-set.

5.1.3 Morphometric Gaussian Process Landmarking

Suppose a prior function of v satisfies M-GP, $f(v) \sim \mathcal{GP}_M(0, K)$, if the covariance function K is defined by:

$$K = S_w H S_w \tag{5.8}$$

$$S_w = \begin{bmatrix} S(v_1, \cdot) - \overline{S(v_1, N(v_i))} \\ \vdots \\ S(v_{|V|}, \cdot) - \overline{S(v_{|V|}, N(v_{|V|}))} \end{bmatrix}, H = \begin{bmatrix} HFE(v_1) & & \\ & \ddots & \\ & & HFE(v_{|V|})) \end{bmatrix} \tag{5.9}$$

For each v_i, there exists a unique $f(v_i)$. The conditional marginal probability of predicting the $(k+1)^{th}$ landmark given previous k landmarks $\left\{v^1, ..., v^k\right\}$ follows the Gaussian distribution:

$$f_{(v^{k+1}|v^1,v^2,...,v^k)} \sim \mathcal{GP}_M(0, K') \tag{5.10}$$

$$K' = diag(K) - K^T_{k,k+1} K^{-1}_k K_{k,k+1} \tag{5.11}$$

$$K_{k,k+1} = \begin{pmatrix} K(v_i, v^1) \\ \vdots \\ K(v_i, v^{k-1}) \end{pmatrix}, \quad K_k = \begin{pmatrix} K(v^1, v^1) & \cdots & K(v^1, v^{k-1}) \\ \vdots & & \vdots \\ K(v^{k-1}, v^1) & \cdots & K(v^{k-1}, v^{k-1}) \end{pmatrix} \tag{5.12}$$

where $diag(K)$ is the diagonal of K. We define the covariance function of each iteration as the uncertainty score Σ_M. The $(k+1)^{th}$ landmark is greedily selected by choosing the vertex with the maximal uncertainty score:

$$v^{k+1} = \underset{v_i \in V \setminus \left\{v^1,...,v^k\right\}}{argmax} \Sigma_M(v_i) \tag{5.13}$$

The first landmark is determined by the maximum value of $diag(K)$. The whole process is shown in Algorithm. 5.1.2.

5.2 Periodic Potential Diffusion Gaussian Processes

5.2.1 Periodic Potential Diffusion Process

A diffusion process is a solution to a stochastic differential equation. It usually has specific physical meanings in a prescribed scenario (Oksendal 2013). Taking the acknowledged heat conduction as an example, the diffusion process solves for a heat distribution of a bounded region at certain time given proper initial and boundary conditions. Let $T(v,t)$ denotes the heat at position v at time t in region D in $\mathbb{R}^d$. Given an initial value and a boundary condition, a diffusion process solves for this parabolic partial differential equation:

$$\frac{\partial T}{\partial t} = \alpha \Delta T + F \quad t \geq 0 \tag{5.14}$$

where Δ is the Laplace operator, F is the potential function. Constant α is usually 1. A fundamental solution to Eq. (A.7) is:

$$T = e^{-t\Delta} f + \int_0^t e^{-(t-s)\Delta} F(s) ds \tag{5.15}$$

The potential function F is a spatial-temporal function for defining the location and pattern of the heat source. We can write it as the multiplication of a Dirac delta function at location v with neighborhood $N_{\mathcal{G}}(v)$ and a temporal function $h(t)$: $F = h(t)\delta(v - N_{\mathcal{G}}(v))$. Denote the Green's function of Laplace operator under the Dirichlet boundary condition as $G(v, N_{\mathcal{G}}(v), t)$. Then Eq. (5.15) is:

$$T(v, N_{\mathcal{G}}(v), t) = \int_0^t G(v, N_{\mathcal{G}}(v), t-s) h(s) ds \tag{5.16}$$

Green's function in a $\mathcal{R}^d$ diffusion problem has the standard form: $G = \frac{e^{-v^2/4t}}{(4\pi t)^{d/2}}$. If the temporal function is periodic $h(t) = cos(\omega t)$, then Eq. (A.8) with the standard form and the periodic temporal function is a periodic Dirac delta potential heat equation:

$$T = \int_0^t cos\omega(t-s)\frac{e^{-v^2/4s}}{(4\pi t)^{d/2}}ds \tag{5.17}$$

Eq. (A.9) is called *periodic potential diffusion process*, which is one of the theoretical foundations of our work.

5.2.2 Equivalence between Diffusion Process and Curvature Flow on Manifold

In differential geometry, a curvature flow numerically links intrinsic geometric features and extrinsic flows together (Kichenassamy et al. 1995). We recall the proof on a planar curve by assuming $\mathcal{G}$ is on 2-dimensional manifold in $\mathbb{R}^3$ for convenience. Figure 3.7 shows sketch plots of the symbols used in this proof. Suppose v is a point on the manifold. $C(v)$ is the intersection between the manifold and the normal plane on v. As known, $C(v)$ is a 1-dimensional smooth curve. Assume one point moves along C from v to v'. Let Δs be the arc length of this movement and θ be the rotation angle of the tangent vector, then we can define the following concepts: (i) the velocity vector at v is $\frac{dC}{dv}$; (ii) the velocity is the magnitude of the velocity vector, which is $\left|\frac{dC}{dv}\right| = \frac{ds}{dv}$; (iii) the unit tangent vector $T = \frac{dC}{ds}/\left|\frac{dC}{ds}\right|$ and the unit normal vector $N = \mathcal{R}T$, $\mathcal{R}$ is the $\pi/2$ rotation matrix; (iv) the curvature κ measures how fast the unit tangent vector rotates relative to the arc length, therefore, κ is defined as: $\kappa = \lim_{\Delta s \to 0}\left|\frac{\Delta\theta}{\Delta s}\right|$. And we can further get $\frac{dT}{ds} = (-sin\theta, cos\theta)\frac{d\theta}{ds} = \kappa N$ and similarly $\frac{dN}{ds} = -\kappa T$. Assume all points on the curve start to move along their normal directions at a velocity of $\kappa(v)$ during time t, we have curvature flow: $\frac{dC}{dt} = \kappa N$. With the equation (iii) and (iv), we write the curvature flow as: $\frac{dC}{dt} = \frac{d}{ds}\frac{dC}{ds}$, which suggests a diffusion process. This equivalence indicates that the diffusion process can theoretically reflect the geometric features. The intrinsic example is more commonly

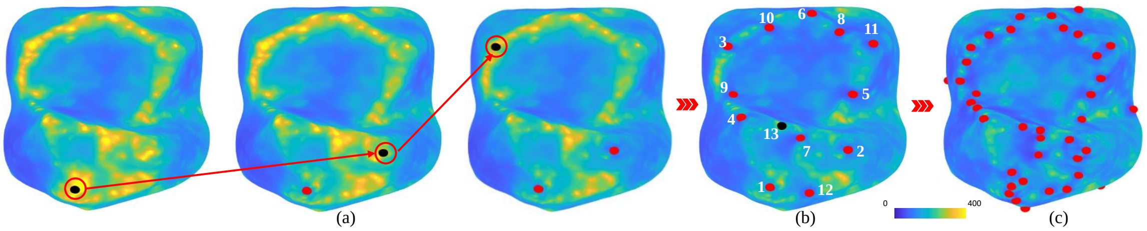


Figure 10. Saliency maps and saliency extraction of a molar model (Boyer et al. 2011). The saliency map is rendered by the normalized saliency values. (a) The progress of sequentially selecting the first three salient vertices. The candidate and selected salient vertices are shown as the black and red spheres, respectively. Note the saliency change after a candidate is selected. (b) The first 12 salient vertices and the 13^{th} salient vertex candidate. (c) The saliency map during the 40^{th} saliency extraction. The high saliency regions are still centralized in ROIs.

seen in heat kernel based work (Bronstein et al. 2017). This is another theoretical foundation of PPDK and it explains why PPDK is able to capture geometric features without adding any explicit feature descriptors.

5.2.3 Periodic Potential Diffusion Kernel

Our inspiration starts from the following two observations:

(1) **The first observation** is that the integral Laplace transform of function $f(t) = t^{d-1}e^{-\frac{1}{4}at}$ in $\mathbb{R}^d$ has several similar terms with the well-known Matérn kernel as shown in Eq. (A.10) and (A.11) respectively:

$$\int_0^\infty t^{d-1}e^{-\frac{1}{4}at}e^{-st}dt = 2\left[(\frac{1}{4}a)^{\frac{1}{2}}s^{-\frac{1}{2}}\right]^d \mathcal{K}_d(a^{\frac{1}{2}}s^{\frac{1}{2}}) \tag{5.18}$$

$$C(\tau) = \frac{\sigma^2}{\Gamma(d)2^{d-1}}(2\sqrt{d}\tau\kappa)^d\mathcal{K}_d(2\sqrt{d}\tau\kappa) \tag{5.19}$$

Both equations have a term of modified Bessel function of the second kind $\mathcal{K}_d$ and the rest parts are functions with the same dimension order d. This indicates a way to derive a solid kernel function from solving the integral.

(2) **The second observation** is that the Green's function of Δ in 3D diffusion problem is a specific realization from the family of $t^{d-1}e^{-\frac{1}{4}at}, d = 3$. Furthermore, Bochner (Bochner 2005) proved that a stationary kernel is positive definite in $\mathbb{R}^d$ if it has the Fourier transform of such measure function F: $K(x, x') = \int_{\mathbb{R}^d} cos(\omega(x - x'))Fd\omega$. Therefore, Eq. (4.14) is a temporal stationary kernel function regarding the temporal variable. If we can prove that it is positive semi-definite (PSD) regarding the spatial part, then Eq. (A.10) is a solid spatial-temporal stationary kernel function.

The theoretical inspiration is straightforward, but the challenge is that Eq. (A.10) has no explicit solution (Hartikainen and Särkkä 2010; Särkkä 2011). Hence, reasonable approximations are used to solve the integral. Eq. (A.10) is estimated to be the combination of a cosine Fourier transform$\hat{f}_c(\omega)$ and a sine Fourier transform$\hat{f}_s(\omega)$ when t is large:

$$\begin{aligned} T &= cos(\omega t)\int_0^t cos(\omega s)G(s)ds + sin(\omega t)\int_0^t sin(\omega s)G(s)ds \\ &\approx cos(\omega t)\hat{f}_c(\omega) + sin(\omega t)\hat{f}_s(\omega) \end{aligned} \tag{5.20}$$

The cosine and sine transforms from the Laplace transform in Eq. (A.12) are:

$$\hat{f}_c(\omega) = 2^{-1-d/2}\pi^{-d/2}\left[(\sqrt{-i\omega})^{d-2}\Sigma_1 + (\sqrt{i\omega})^{d-2}\Sigma_2\right] \tag{5.21}$$

$$\hat{f}_s(\omega) = -i2^{-1-d/2}\pi^{-d/2}\left[(\sqrt{-i\omega})^{d-2}\Sigma_1 - (\sqrt{i\omega})^{d-2}\Sigma_2\right] \tag{5.22}$$

where Σ_1 and Σ_2 are:

$$\Sigma_1 = (\|v\|\sqrt{-i\omega})^{1-d/2}\mathcal{K}_{1-d/2}(\|v\|\sqrt{-i\omega}) \tag{5.23}$$

$$\Sigma_2 = (\|v\|\sqrt{i\omega})^{1-d/2}\mathcal{K}_{1-d/2}(\|v\|\sqrt{i\omega}) \tag{5.24}$$

The solution to the periodic potential diffusion problem is then produced by substituting Eq. (A.13) and Eq. (A.14) into Eq. (A.12):

$$\begin{aligned}T &= 2^{-1-d/2}\pi^{-d/2}\left[(\sqrt{-i\omega})^{d-2}\Sigma_1 + (\sqrt{i\omega})^{d-2}\Sigma_2\right] cos(\omega t) \\ &- i2^{-1-d/2}\pi^{-d/2}\left[(\sqrt{-i\omega})^{d-2}\Sigma_1 - (\sqrt{i\omega})^{d-2}\Sigma_2\right] sin(\omega t)\end{aligned} \tag{5.25}$$

If we define two complex functions f and f^* as:

$$\begin{aligned}f &= (\sqrt{-i\omega})^{d-2}\Sigma_1 - (\sqrt{i\omega})^{d-2}\Sigma_2 \\ f^* &= (\sqrt{-i\omega})^{d-2}\Sigma_1 + (\sqrt{i\omega})^{d-2}\Sigma_2\end{aligned} \tag{5.26}$$

Then, Eq. (5.25) is simplified to be:

$$T = 2^{-1-d/2}\pi^{-d/2}\left[cos(\omega t)f - isin(\omega t)f^*\right] \tag{5.27}$$

Each sine and cosine term in Eq. (5.27) is weighted with a complex function. Noting that this composite form is different from the sine-cosine form of the Fourier series of a periodic function. When the manifold is in $\mathbb{R}^3$, Eq. (5.23) and (5.24) become

$$\Sigma_1 = \sqrt{\frac{\pi}{2}}e^{-\|v\|\sqrt{-i\omega}}, \quad \Sigma_2 = \sqrt{\frac{\pi}{2}}e^{-\|v\|\sqrt{i\omega}} \tag{5.28}$$

and Eq. (5.27) is written as:

$$T = \frac{1}{4\pi}e^{-\|v\|\sqrt{\frac{1}{2}\omega}}cos(\|v\|\sqrt{\frac{1}{2}\omega} + \omega t) \tag{5.29}$$

The frequency ω can be a single value or a multi-frequency setting. In our applications, we define N_{fre} frequencies with $\omega = \sqrt{0.2\pi n}, n = [1, ..., N_{fre}]$. We use the same number of values for t by dividing $[0, \frac{\pi}{2}]$ into N_{fre} equal line-spaces. The final kernel matrix is the average of accumulated all single value covariance functions. The ω and t can also be hyper-parameters and determined by parameter tuning methods. Such setting is widely used in variational inference or posterior estimation, and this is beyond our topic in this paper. Inspired by the kernel principal component analysis (KPCA)

Algorithm 2 Saliency Extraction on manifold with PPD-GP

```
procedure $\mathcal{G}$:PPD-GP
    $\mathcal{S}$                          ▷ Number of salient vertices needed, $\mathcal{S} << |\mathcal{V}|$
    $N_{\mathcal{G}}$                         ▷ Neighborhood via KNN or Fast Marching
    $K = TWT$                        ▷ Kernel building with Eq. (5.30), (5.31)
    $\tilde{V} \leftarrow \varnothing$                        ▷ initialize salient vertex set $\tilde{V}$ as empty
    while $s \leq \mathcal{S}$ do
        if s=1 then
            $\Sigma_{\mathcal{G}} \leftarrow diag(K)$                ▷ initialize saliency map
        else
            $\Sigma_{\mathcal{G}}$                            ▷ calculate saliency map.
        end if
        $\tilde{v}^s \leftarrow argmax\Sigma_{\mathcal{G}}$               ▷ vertex with the highest saliency
        $s \leftarrow s + 1$
    end while
    return $\tilde{V} = \left\{\tilde{v}^1, ..., \tilde{v}^{\mathcal{S}}\right\}$          ▷ low-dimensional embedding
end procedure
```

which uses a weighted multiplication of the covariance matrix (Schölkopf, Smola, and Müller 1998), we define PPDK in a weighted squared form: $K = TWT$. The weight matrix W is a diagonal matrix with the absolute sum of each row as the diagonal entry. The final version of the kernel function K and weight matrix W is:

$$K(v, N_{\mathcal{G}}(v)) = \frac{1}{4\pi N_{fre}} \sum_{n=1}^{N_{fre}} e^{-\|v\|\sqrt{\frac{1}{2}\omega_n}} cos(\|v\| \sqrt{\frac{1}{2}\omega_n} + \omega_n t_n) \tag{5.30}$$

$$W(v) = \sum |K(v, \cdot)| \tag{5.31}$$

Lemma 3. *PPDK is a spatial-temporal stationary kernel function.*

Proof. The proof of the temporal part is solved by using Bochner's theorem regarding the variable t. The rest is to prove Eq. (5.29) is also PSD regarding the spatial variable v before summarizing to Eq. (5.30). Eq. (5.29) is the composition of an exponential function $e^{-||v||\sqrt{\frac{1}{2}\omega}}$ and a cosine function $cos(||v||\sqrt{\frac{1}{2}\omega} + \omega t)$. It is acknowledged that these two functions are PSD according to variable v. By calling the composition property of PSD matrix, we get that Eq. (5.29) is also PSD regarding the variable v. Entries in weight matrix W are positive. By calling the property of a PSD matrix that the multiplication between a positive real function and a PSD function is still PSD (Bhatia 2007), we get that Eq. (5.30) is PSD. Lemma 3 is proved. Another proof method is using the Cholesky decomposition which is often applied in posterior approximation and variational inference. Cholesky decomposition restricts that only a PSD matrix can be factorized into the product of a lower triangular matrix and its conjugate transpose. PPDK is feasible to be decomposed in tests. □

Although we focus on the spatial analysis here, we mention the proof of temporal part for giving the possibility of extending to some spatial-temporal applications. The temporal variable is used as a parameter like frequency ω in our applications. Lemma 1 provides a thought of deriving eligible families of kernels through proper ordinary derivative equation (ODE) or partial derivative equation (PDE) that describes a real physics process. It is said that a kernel based on a rational physics process guarantees the inference well matches with the inherent structures (Stein 1991).

Theorem 1. *A real valued function $T(v,t)$ on $\mathbb{R}^d$ is a spatial-temporal kernel function if it is a diffusion process given proper initial values v_0 and boundary conditions: $\frac{\partial T}{\partial t} = \alpha \Delta T + P(t)\delta(v)$. α is a positive constant. $P(t)$ is a periodic function. $\delta(v - v_0)$ is the Dirac delta function. Δ is the Laplace operator.*

Proof. Theorem 1 is the generalization of Lemma 1. As known, any parabolic partial differential equation has a corresponding Green's function so that the solution has the form of Eq. (4.13). If a periodic function P has a general cosine form, then the proof will be similar with that of Lemma 1 as the spatial part is the same Dirac delta function. The generalization mainly happens in temporal part. The

Fourier series of a periodic function is: $P(t) \sim \frac{1}{2}a_0 + \sum_{n=1}^{\infty}\left[a_n cos(nt) + b_n sin(nt)\right]$, a, b are real numbers. By using the trigonometric sum formulae, we get: $x_n = \pm\sqrt{a_n^2 + b_n^2}$ and $y = arctan(-\sqrt{\frac{b_n}{a_n}})$ when assuming the existence of a cosine function $x_n cos(nt + y_n)$ that is equal to $a_n cos(nt) + b_n sin(nt)$. Then the Fourier series is $P(t) \sim \frac{1}{2}a_0 \pm \sum_{n=1}^{\infty}\sqrt{a_n^2 + b_n^2} cos\left[nt + arctan(-\sqrt{\frac{b_n}{a_n}})\right]$. So any periodic functions can be approximated to be the linear combination of cosine functions. Because of the PSD summation property, the solution to the parabolic PDE is PSD. □

In (Särkkä 2011), Särkkä indicates that any stochastic partial differential equation is potential to be a kernel. Theorem 1 provides an applicable class of solutions. Comparing with methods derived from analogous theory background, e.g. the heat kernel (HK) (Berline, Getzler, and Vergne 2003), PPDK is in a dynamic potential scenario which is more flexible and accurate than those without potential definitions. An insight is its connection with discrete convolution operation. On a smooth manifold, the convolution operation is in a discrete accumulated form: $D(v)f = \sum_V f(v')u(v, v')$ (Bronstein et al. 2017). Weight function $u(v, v')$ measures the similarities between x and x', which matches with the meaning of a kernel. So the convolution on point x is essentially weighted accumulation of function values on x. Therefore, PPDK accomplishes a result of pointwise convolutional filtering.

5.3 Experiments

5.3.1 Morphological Gaussian Processes

This group of experiments focus on the cortical morphometry analysis for the study of AD. Previous research (Fan et al. 2018; Thompson et al. 2003) has demonstrated a

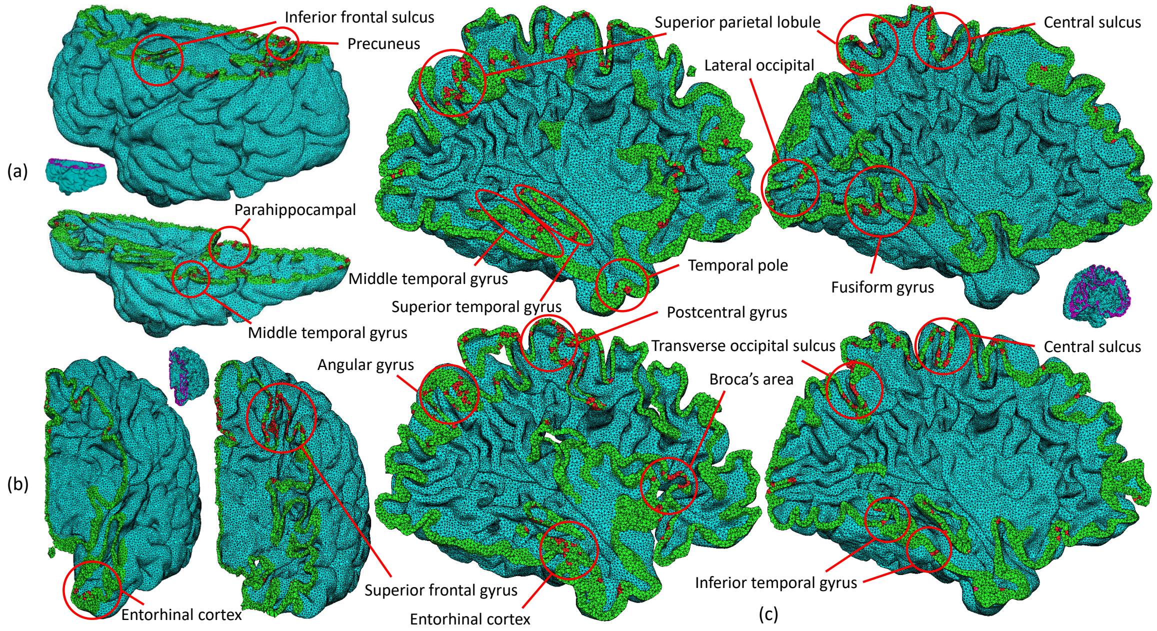


Figure 11. Visualization of landmarks on a grey matter tetrahedral mesh. (a) Slices of horizontal plane; (b) Slices of sagittal plane; (c) Slices of coronal plane. The landmarks are identified by the red spots. Some regions of interest are marked by red circles with annotations on the side.

strong correlation between grey matter morphometry and the AD severity and here we aim to discriminate AD clinical groups by cortical atrophy analyses. We use 518 left cerebral hemisphere structural MR images including 144 AD patients, 187 cognitively unimpaired (CU) visitors and 187 mild cognitive impairment (MCI) patients from the baseline subset of the Alzheimer's Disease Neuroimaging Initiative 2 (ADNI2) (S. G. Mueller et al. 2005). Their grey matter tetrahedral meshes are generated by using the pipeline in (Fan et al. 2018) and the average number of vertices in each mesh is about 154,000. When computing the scale-invariant WKS, we select the smallest 50 pairs of eigenvalues and eigenfunctions. The first eigenvalue λ_0 is abandoned because $\lambda_0 \approx 0$; we select 100 ϵ values by evenly dividing the scope $(\epsilon_{min}, \epsilon_{max})$, where $\epsilon_{min} = log(\lambda_{min})$, $\epsilon_{max} = log(\lambda_{max})$; the increment of ϵ is $(\epsilon_{max} - \epsilon_{min})/100$.

10-fold cross validation is applied in classifications. The performance measurements include accuracy (ACC), sensitivity (SEN) and specificity (SPE). We adopt Support Vector Machine (SVM) as the classifier. The feature for classifications is not a key consideration in this paper. Any pointwise features are acceptable. For convenience, we still use the scale-invariant WKS. We choose the following methods for comparisons: (1) Full feature space method denoted as Full. The feature vector on each vertex is truncated to be a feature matrix. We use principal component analysis (PCA) to organize the feature matrices of different subjects into the same dimension (Fan et al. 2018). (2) The heat kernel GP denoted as HK-GP. HK-GP is also used to select landmarks and form a feature subspace. The definition is the same as that in (Gao et al. 2018). (3) The Spectral Mixture GP denoted as SM-GP. We use the SM-GP defined in (Wilson and Adams 2013) to select landmarks.

5.3.1.1 Visualization of the Landmarks

We randomly select one grey matter tetrahedral model and choose 5,000 landmarks by Algorithm. 5.1.2. The landmarks are shown in Fig. 11. We render one face that the landmark vertex belongs to so that the result is more visible. The landmarks are illustrated from three anatomical cutting planes: coronal plane, sagittal plane, and horizontal plane. We choose slices that contain as many landmarks as possible on each cutting plane. The landmarks are marked by red spots. Some regions of interests are marked by red circles with annotations on the side. The vertices in the corpus callosum have been excluded from the candidates before the landmarking because this region is not part of the cortex. We can observe that the landmarks have a good aggregation in regions of interests. Some regions have been clinically proved to be

GROUP		Full	HK-GP (200)	SM-GP(200)	M-GP(100)	M-GP(150)	M-GP(200)
AD	ACC	0.9158	0.8391	0.9049	0.90	0.9455	**0.9455**
-	SEN	0.913	0.8174	0.9036	0.9231	0.9739	**0.9826**
CU	SPE	**0.9195**	0.8609	0.9085	0.8571	0.9080	0.8966
AD	ACC	0.8911	0.8221	0.9196	0.8267	0.9158	**0.9307**
-	SEN	0.8783	0.8261	0.9275	0.8261	**0.9391**	0.9304
MCI	SPE	0.9080	0.8161	0.9080	0.8276	0.8391	**0.931**
CU	ACC	0.813	0.797	0.8076	0.7478	0.7522	**0.8304**
-	SEN	0.7565	0.7739	**0.9652**	0.8261	**0.9652**	0.8
MCI	SPE	**0.8696**	0.8276	0.7816	0.5304	0.5391	0.8609

Table 4. Classifications of different AD clinical groups. The highest value of each row is in bold. The number of used landmarks is annotated in brackets. Full denotes using the features of all vertices. HK-GP denotes the landmarking by heat kernel GP. SM-GP denotes the landmarking by spectral mixture GP. M-GP is our method.

closely related to the progression of AD, such as the precuneus, entorhinal cortex, temporal pole, and superior parietal lobule (L. Pini et al. 2016).

5.3.1.2 Classifications of AD Clinical Groups

This is a further numerical experiment of the previous one. A representative subspace is expected to yield a similar or even better classification result compared with using the full feature space. We apply algorithm 5.1.2 on all the grey matter tetrahedral meshes and generate $N = 100, 150, 200$ landmarks. Then we use these landmarks to form differently sized subspaces and do the classification. The results are given in Table 4. We can see that the performance of M-GP is similar to or even better than that of using the full feature space. The numerical results prove that the subset inherits or even improves the discriminability of the original data, which means we can trust this simplified subset in cortical morphometry analysis as an effective substitution of the full point set and reduce the computation burden.

5.3.2 Periodic Potential Diffusion Gaussian Processes

There are two main categories of applications in this group of experiments: saliency extraction and shape retrieval as classifications. We use 10-fold cross-validation and support vector machine (SVM) as the classifier in all retrieval applications.

Comparison methods: (1) Heat kernel GP (HK-GP) (Berline, Getzler, and Vergne 2003). We choose this method because it shares the same theory foundations but consider no source function. The diffusion kernel in (Kondor and Lafferty 2002) also shares the same theoretic basis. It yields similar performance with HK-GP in tests, so we skip it in this paper; (2) Spectral mixture kernel GP (SMK-GP) (Wilson 2014). SMK has a convincing performance in image domain and it has a quite similar expression with PPDK. We also test some other successful kernels such as the Matérn kernel family (Stein 1991). But the performances are not comparable, so we neither demonstrate them here; (3) Periodic kernel GP (PK-GP) (Williams and Rasmussen 2006). PK is the exponential of a periodic function. We use PK to check if a simple periodicity consideration can contribute to the task; (4) Mesh saliency (MS) (Lee, Varshney, and Jacobs 2005). This is a highly cited classical method in saliency detection on meshes; (5) W-GP (Gao, Kovalsky, and Daubechies 2019). W-GP is the current state-of-the-art GP method on manifolds. We directly use the parameter settings in their paper.

We tested more than five methods, but some of them yield similar results. So we choose five representatives to demonstrate. Since the curvature is only defined in 2-dimensional manifold space, W-GP and MS are only used in the first two experiments.

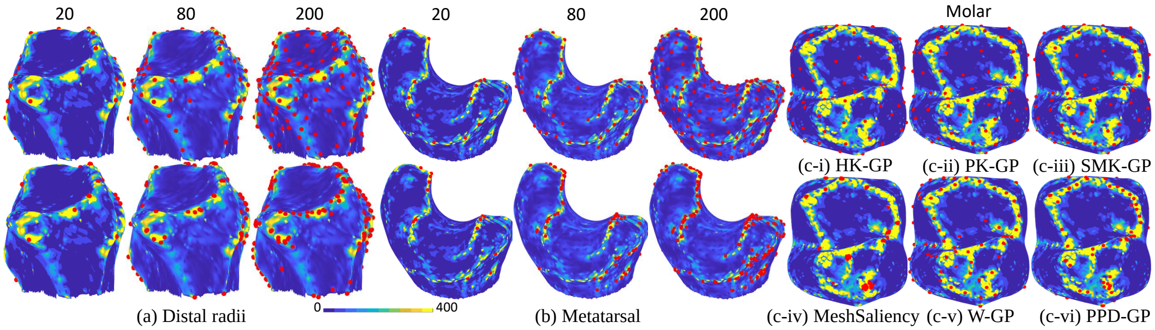
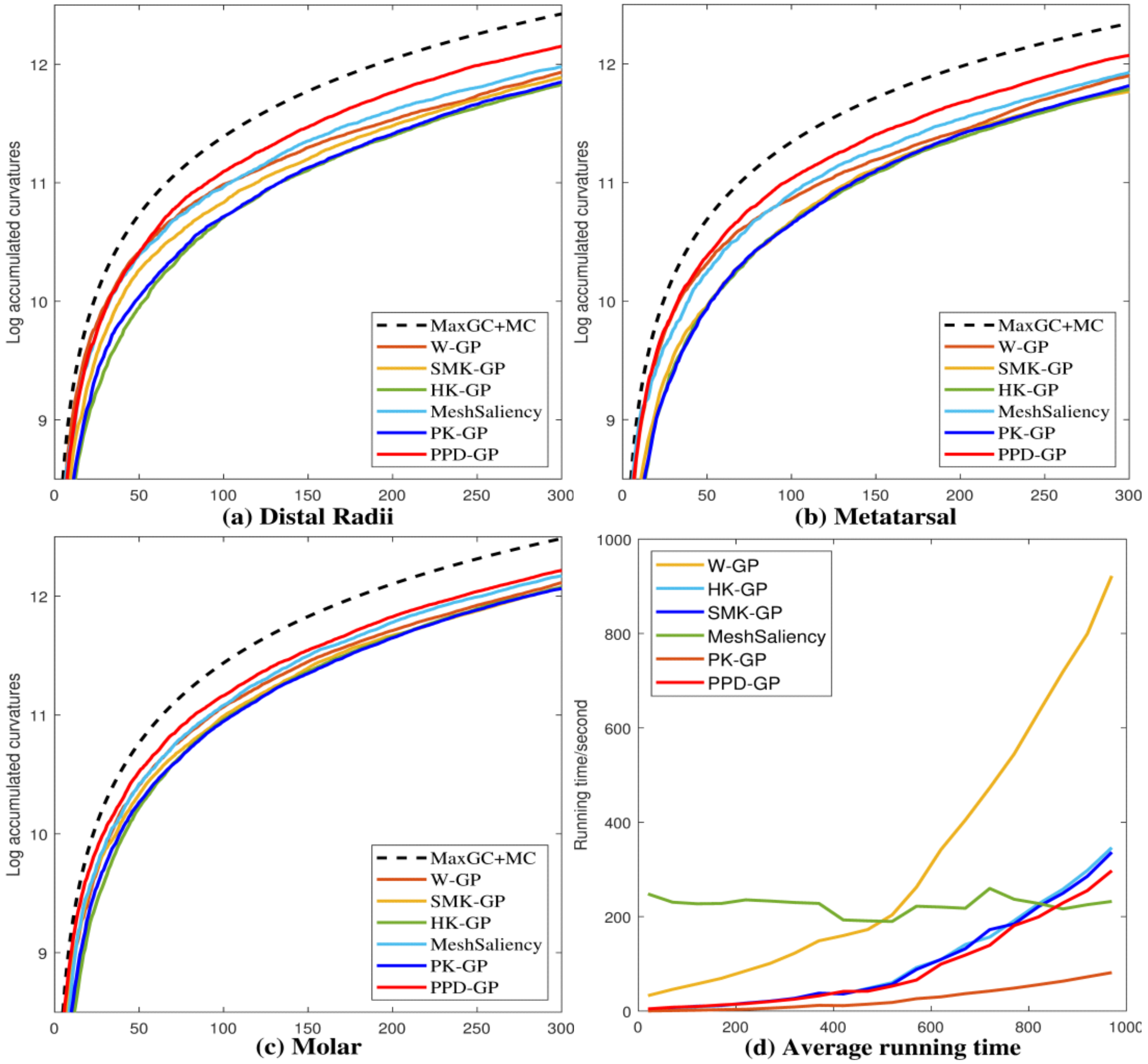


Figure 12. Visualization of salient vertices. (a) and (b) illustrate 20, 80, 200 salient vertices of distal radii and metatarsal models. The upper row is the results of W-GP (Gao, Kovalsky, and Daubechies 2019), the bottom row is ours. (c-i)-(c-vi) illustrate 50 salient vertices recognized by comparison methods and PPD-GP on a molar model. Each mesh is rendered by the normalized Gaussian curvatures. The ROI is the high-curvature yellow region. The salient vertices are marked by the red spheres.

5.3.2.1 Saliency Extraction on Triangle Meshes

In the first experiment, we evaluate the geometry-aware property of PPD-GP and its stability in continuous saliency extraction. We further compare the computational efficiency by counting the average running time of updating the saliency map once. Three datasets released in (Boyer et al. 2011) are used: (i) “Mandibular molars”, or “molar”, contains 116 teeth surface models shown in Fig. 4.3(c); (ii) “First metatarsals”, or “metatarsal”, contains 57 models shown in Fig. 4.3(b); (iii) Distal radii contains 45 models shown in Fig. 4.3(a). Please refer to (Boyer et al. 2011) for more information about the data. We follow the same parameter setting with (Gao, Kovalsky, and Daubechies 2019). We use the fast marching method to compute geodesic distances and select the neighborhood. When a small number of neighborhood is considered, Euclidean distance and geodesic distance have similar performance in tests.



(a) Distal Radii

(b) Metatarsal

(c) Molar

(d) Average running time

Figure 13. (a)-(c) AC curves of distal radii, metatarsal and molar datasets. (d) Average running time of updating the saliency map once. The x-axis is the number of extracted salient vertices. The unit of running time is second.

In paleobiology and taxonomy, the study regions of such fossil data are usually the marginal ridges, teeth crowns and outline contours where distinguishable geometric features are found (Boyer et al. 2011; Couette and White 2010). We render the meshes with the normalized Gaussian curvature as the ground-truth (Guennebaud, Germann, and Gross 2008). We select 20, 80 and 200 salient vertices and illustrate them in

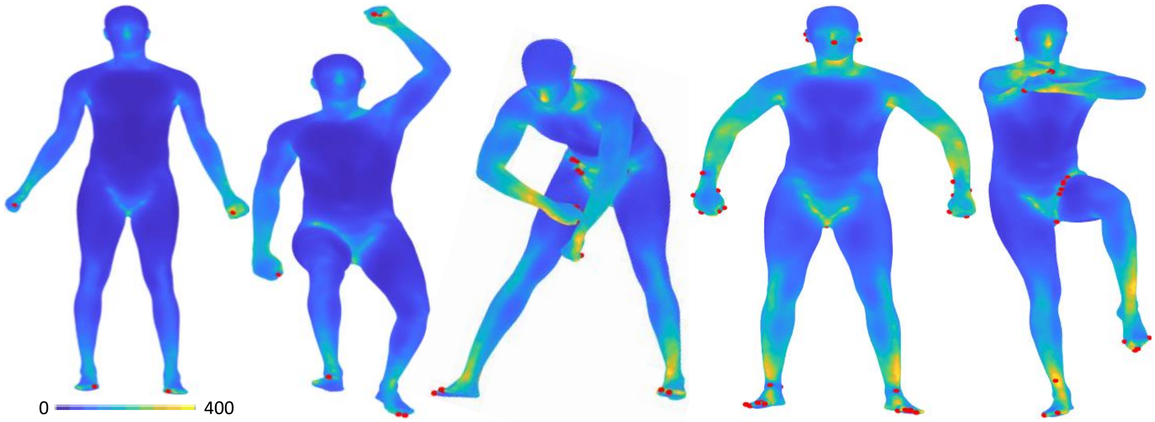


Figure 14. Saliency maps and salient vertices on 3D human models. Salient vertices are marked as red spheres.

Fig. 12. The desired selection is expected to aggregate in yellow regions. Fig. 12(a) and (b) visualize the results of distal radii and metatarsal meshes. The top row is the results of W-GP and the bottom row is ours. We give one example of other comparison methods in Fig. 12(c-i) to (c-vi) since their results are generally not comparable with W-GP and PPD-GP from all aspects. It is clear to see that salient vertices recognized by PPD-GP fit better with expectations as most of them are clustered in significant regions consistently. Numerically, we define an Accumulated Curvature value (AC value) to measure the scattering effect: $AC_\kappa = log \sum_{k=1}^{\kappa} |GC_k| + |MC_k|$, where GC and MC are normalized Gaussian curvature and mean curvature; κ is the number of salient vertices. A larger AC value is preferred here.

The AC values of each mesh aggregate to an AC curve. We average the AC values of meshes within the same dataset at each number of salient vertices to get an average AC curve of the dataset which is shown in Fig. 13(a)-(c). Obviously, a maximum AC value exists when only selecting points with the largest accumulated curvatures at each time. It returns the highest AC curve as the dash curves in the figures. Although this maximum curve is usually meaningless in applications (selected vertices are too locally clustered to be the ground-truth), it is a good reference to others. To a certain extent, AC curve measures the clustering performance but the visual facts, so we recommend to use both the visual and numerical way to verify. Results show that the scattering effect is well suppressed and PPD-GP has excellent stability in preserving the geometry-aware property during continuous regressions.

The computational efficiency is measured by averaging the running time of updating the saliency map once, as shown in Fig. 13(d). The MS method is not affected by the number of selected points, but active learning methods gradually slow down because of the dimension increase in the prior knowledge. PPD-GP has an acceptable increment of the computational time considering its performance.

5.3.3 Saliency Extraction on Tetrahedral Meshes and Volumetric Shape Retrieval

In the third experiment, we demonstrate that PPD-GP is competent for morphometric analysis on 3-dimensional manifolds, i.e., the tetrahedral meshes as our choice. Volumetric modeling has unique advantages in solving complex structural problems but it also suffers from the huge dimensions of the data due to the dense spatial sampling over the volume. Our goal is to select a much smaller discriminative subset from internal body or the surface of the object by applying Algorithm 5.1.2. Furthermore, we take the grey matter atrophy classification as a volumetric shape retrieval problem to numerically evaluate the representativeness of the embedding. We classify the patients and healthy visitors based on their grey matter atrophy levels measured by geometric features. Structurally, Alzheimer's disease causes abnormal atrophy of the cerebral cortex, which results in a gradually thinner grey matter, i.e.

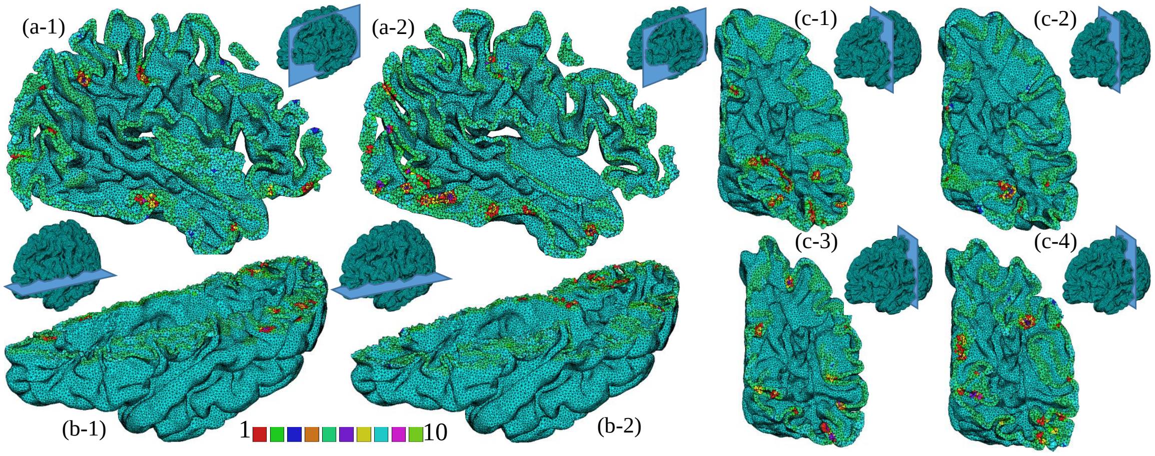


Figure 15. Results of Saliency extraction on Tetrahedral Meshes. Visualization on three anatomical cutting planes: (a) sagittal cutting plane; (b) coronal cutting plane; (c) horizontal cutting plane.

thinner cortical thickness, than the normally aged people (Lorenzo Pini et al. 2016; Gang Wang and Yalin Wang 2017).

We use 3D magnetic resonance imaging (MRI) data from the baseline collection of ADNI2 (Susanne G Mueller et al. 2005; Jack Jr et al. 2008). We follow the pipeline in (Fan et al. 2018) to generate the grey matter tetrahedral meshes. Each mesh contains about 150,000 vertices. We compute 5,000 salient vertices for visualization so that the targets are visible. Numerically, we use the WKS features of 1000, 1500, 2000, and 3000 salient vertices to separate patients from healthy visitors. Noting that this WKS is different from the one used in the second experiment. We use the discretization method in (Fan et al. 2018) to solve the eigen decomposition problem and define the WKS. Totally we use 275 left cerebral hemisphere structural MRIs including 88 AD patients and 187 Cognitively Unimpaired (CU) visitors.The Euclidean KNN is used to compute the neighborhood.

Figure 15(a) shows the saliency map of one randomly-selected grey matter tetrahedral mesh on the sagittal cutting plane. We sum up the saliency values of four vertices as the saliency of the tetrahedron. Then, we normalize the tetrahedron saliency values and divide them into 10 classes to visualize. Fig. 15(b) show the salient vertices from different anatomical cutting planes. For a better visual effect, we render the tetrahedron to red if it consists of any numbers of salient vertex. The visualization results show that points mainly centralize in ROIs such as the temporal pole and temporal gyrus as expected. Previous studies show that these regions may be closely related to AD judged by their functionalities, clinical diagnosis and morphometry changes (Lorenzo Pini et al. 2016). Table 5 shows the classification results by using WKS of the embeddings. It proves that a well-selected subset may be more discriminative than using global features. We can reason that an optimized number of salient vertices exists, but we stop testing more points due to a consideration of balancing the accuracy and the computational time. In general, we successfully decrease the dimension of the effective regions from hundred thousand level to thousand level and this simplified embedding is more distinguishable than that of comparisons or using the whole features in classifying the healthy visitors from patients.

5.3.3.1 Saliency Extraction on Point Clouds

In the fourth experiment, we demonstrate that our method is applicable to point clouds. In many vision applications, an object or a scene is scanned and stored as a point cloud. The lack of connection information disables many existing saliency detection and saliency extraction methods when analyzing point clouds.

We use McGill 3D shape benchmark (Siddiqi et al. 2008) in this experiment. It

AD-CU		HK-GP	P-GP	SM-GP	PPD-GP	Global
1000	ACC	0.9500	0.90	0.9643	**0.9714**	0.9733
	SEN	0.9535	0.9032	0.9474	**0.9605**	0.9570
	SPE	0.9459	0.8966	0.9844	**0.9844**	0.9927
1500	ACC	0.9333	0.9325	0.9583	**0.9750**	0.9733
	SEN	0.9355	0.919	0.9403	**0.9659**	0.9570
	SPE	0.9310	0.9586	0.9811	**0.9861**	0.9927
2000	ACC	0.9417	0.9000	0.9500	**0.9778**	0.9733
	SEN	0.9552	0.8636	0.9286	**0.9703**	0.9570
	SPE	0.9245	0.9444	0.9773	**0.9873**	0.9927
3000	ACC	0.9409	0.9125	0.9500	**0.9800**	0.9733
	SEN	0.9527	0.9302	0.9302	**0.9737**	0.9570
	SPE	0.9275	0.8919	0.973	**0.9884**	0.9927

Table 5. GM atrophy classification. The performance of PPD-GP escalates with more salient vertices added. The results of using global features are references.

contains 400 triangle meshes of different objects. We remove the connection information and only take the point coordinates information as inputs. Euclidean KNN is used to compute the neighborhood and distances. Fig. 16 shows results of 20-30 salient vertices on point cloud inputs. The results indicate that our method can detect the potential geometric features from the mass and select points in the distinguishable regions, such as the fingers, wings, and the heads and legs of animals. Noticing that our method still yields reasonable inferences when inputs contain non-isometric or ambiguity deformations, such as the first two hand models. Additionally, our method works better on evenly distributed point clouds where the density of samplings is generally the same everywhere. The results on unevenly distributed point clouds show that high-density regions have a higher priority to be selected. Further investigation is warranted to characterize its performance with more applications.

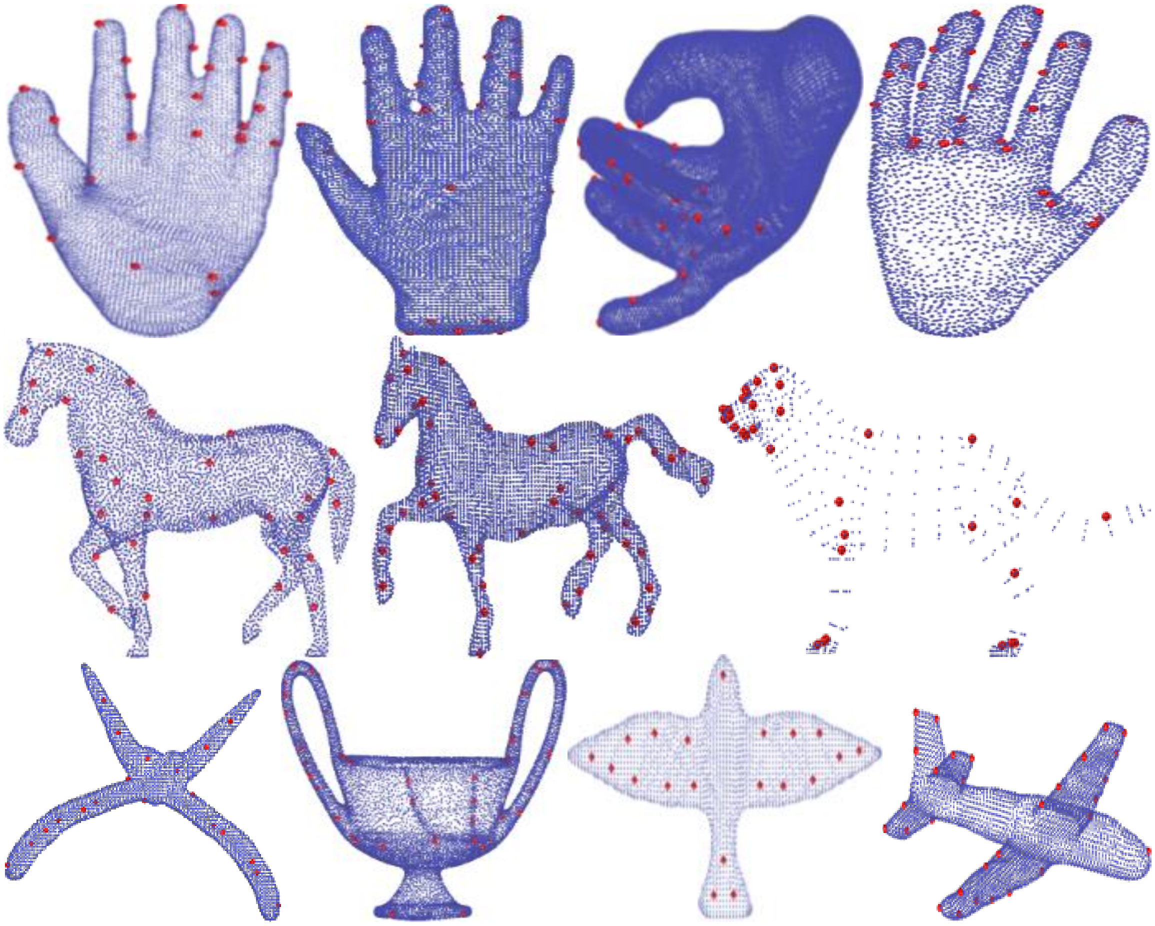


Figure 16. saliency extraction on point clouds of different shapes. salient vertices are marked by the red spheres.

5.3.3.2 Human Pose Retrieval

In the second experiment, we numerically evaluate the geometric significance of the salient embeddings by solving a shape retrieval problem. We use SHREC14 non-rigid 3D human model dataset (Pickup et al. 2014) in this experiment. This dataset contains 400 closed-surface triangle meshes of 40 human subjects with 10 different poses. Each mesh contains about 15000 vertices. Our goal is to retrieve human poses, or called the pose recognition, in manner of classifications. The Scale Invariant Heat

SIHKS		HK-GP	W-GP	P-GP	SM-GP	MS	PPD-GP	Global	WKS	HK-GP	W-GP	P-GP	SM-GP	MS	PPD-GP	Global
50	ACC	0.5258	0.5444	0.5022	0.5667	0.5272	**0.6247**	0.5478	50	0.9378	**0.9869**	0.9775	0.9764	0.9264	0.9847	0.9736
	SEN	0.5383	0.5461	0.5133	0.5744	0.5267	**0.6278**	0.5500		0.9383	**0.9889**	0.9789	0.9794	0.9244	0.9828	0.9772
	SPE	0.5133	0.5428	0.4911	0.5589	0.5278	**0.6217**	0.5456		0.9372	0.9850	0.9761	0.9733	0.9283	**0.9867**	0.9700
80	ACC	0.5092	0.5106	0.5378	0.5131	0.5428	**0.6411**	0.5478	80	0.9544	0.9861	0.9633	0.9731	0.9186	**0.9864**	0.9736
	SEN	0.500	0.5172	0.5406	0.5139	0.5422	**0.6433**	0.5500		0.9567	**0.9878**	0.9650	0.9744	0.9167	0.9856	0.9772
	SPE	0.5183	0.5039	0.5350	0.5122	0.5433	**0.6389**	0.5456		0.9522	0.9844	0.9617	0.9717	0.9206	**0.9872**	0.9700
250	ACC	0.4908	0.4972	0.5219	0.4903	0.5886	**0.5992**	0.5478	250	0.9722	0.9617	0.9453	0.9458	0.9436	**0.9903**	0.9736
	SEN	0.4833	0.5022	0.5344	0.4878	**0.6050**	0.5928	0.5500		0.9739	0.9617	0.945	0.9528	0.9433	**0.9911**	0.9772
	SPE	0.4983	0.4922	0.5094	0.4928	0.5722	**0.6056**	0.5456		0.9706	0.9617	0.9456	0.9389	0.9439	**0.9894**	0.9700

Table 6. Human pose retrieval with SIHKS and WKS. The best measurements are marked in bold. Regardless of the features, the performance of PPD-GP is generally better than comparisons and global features. This proves our method is more representative and significant. A well-selected number of salient vertices and a proper feature can maximize the performance.

Kernel Signature (SIHKS) (Bronstein and Kokkinos 2010) and WKS are used as the vertex-wise feature descriptors. We transform the 10 poses retrieval problem to 45 binary classifications and use 10-folds cross-validation on each of them. The final performance is measured by averaging all the results. For avoiding the bias brought by different saliency extraction sequences, we apply principal component analysis (PCA) on the initial concatenated feature matrices to regularize the inputs. We select six sets of salient vertices with the dimension from 50 to 450. A demonstration of saliency map is shown in Fig. 14 and the retrieval results of 50, 80 and 250 salient vertices are shown in Table 6. By comparing the performance within each feature space, it shows that the results of PPD-GP are generally better than comparisons. This verifies that PPD-GP is able to stably capture the geometric significance and well infer the representative spatial positions from the original data.

5.3.3.3 Unsupervised Salient Point Selection on ModelNet40

We provide more salient point selection results on point clouds here. Figure. 17 shows twelve examples of salient point selection on McGill 3D shape benchmark (Siddiqi

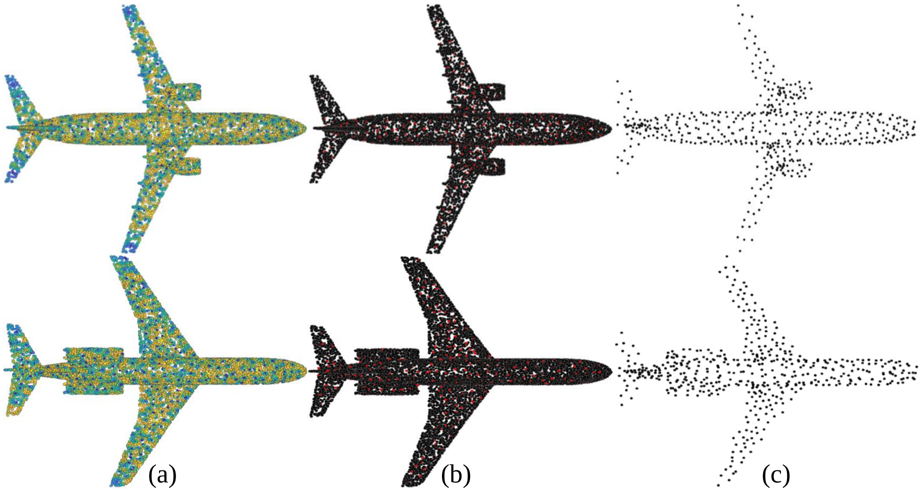


Figure 17. Illustration of saliency selection on Modelnet40 (**wu20153d**).
(a)Demonstrations of saliency maps after selecting 200 salient points.
(b)Demonstrations of 200 salient points. The salient points are marked by red spheres.
(c) The selected 200 salient points out of 10000 vertices can generally represent the original shape.

et al. 2008). The shapes have different sampling densities. The original data type is the triangle mesh. We remove their edge connections and only use the vertex coordinates as inputs. The results show that the GAC-GP can learn the geometric property of the inputs in the prior and the selected salient points are representative in distinguishing the shapes.

Figure. 17 illustrates two examples of salient point selection on Modelnet40 dataset. Figure. 17 (a) demonstrates the saliency maps after selecting 200 salient points. Figure. 17 (b) shows salient points on the original point clouds. Figure. 17 (c) is 200 salient points. We can see that 200 salient points can generally depict the original shapes. Noting that we did not use salient point selection algorithm in the point cloud classifications in the main submission. This is mainly because the comparison methods used the whole data as inputs, for fairness and convenience, we also use 10000 vertices as inputs in the experiments.

Chapter 6

SOLVING SPDES FOR BAYESIAN LEARNING

6.1 Deep Gaussian Processes with Doubly Stochastic Variational Inference

A DGP is a deep belief network that hierarchically concatenates multiple Gaussian process latent variable models together (GP-LVMs) (Damianou and Lawrence 2013). It mimics the composition of restricted Boltzmann machines (RBMs) in NNs. The sparse variational inference is usually used in GPR to estimate the posterior and avoid the cubic complexity (**sheth2015sparse**). Suppose M inducing points $Z = \{z_1, ..., z_M\}(M \ll N)$ are selected, the complexity is decreased to $\mathcal{O}(M^2N)$ in a single GPR. For a DGP, the doubly stochastic variational inference is often applied to estimate the posterior (**salimbeni2017doubly**; **blomqvist2019deep**). Specifically, the sparse variational inference is used to simplify the correlations within layers and keep the correlations between layers unchanged. In a DGP with L layers, the prior is recursively defined on a series of vector-valued stochastic functions $F = \{F^1, ..., F^L\}$. The i^{th} row of F^l is denoted as f_i^l. Function values at inducing points Z are U. Each single function has an independent Gaussian prior and inducing points. A joint density of a DGP can be expressed as:

$$p(Y, \{F^l, U^l\}_{l=1}^L) = \\ \underbrace{\prod_{i=1}^{N} p(y_i|f_i^L)}_{likelihood} \underbrace{\prod_{l=1}^{L} p(F^l|U^l; F^{l-1}, Z^{l-1}) p(U^l; Z^{l-1})}_{prior} \tag{6.1}$$

According to the theories of variational inference, a factorized form of the posterior joint density is defined as (**salimbeni2017doubly**):

$$q(\{f^l, U^l\}_{l=1}^L) = \prod_{l=1}^{L} p(f^l|f^{l-1}, U^l, Z^l)q(U^l) \tag{6.2}$$

where $q(U^l)$ is a Gaussian with mean function m^l and covariance function S^l for layer l. Eq. (6.2) indicates that the prediction of the l^{th} layer, f^l, depends on the previous prediction f^{l-1} and the inducing points of the current layer. By marginalising the approximation $q(U^l)$ from each layer, the i^{th} factorized variational posterior of the final layer is the integral of all paths $(f_i^1, ..., f_i^L)$ through the Gaussian distributions defined by parameters m^l, and S^l:

$$q(f_i^L) = \int \prod_{l=1}^{L-1} q(f_i^l|m^l, S^l; f_i^{l-1}, Z^{l-1})df_i^l \tag{6.3}$$

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

$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_{q(f_i^L)}[logp(y_i|f_i^L)] - \sum_{l=1}^{L} \mathcal{KL}(q(U^l)||p(U^l)) \tag{6.4}$$

where $\mathcal{KL}$ is the Kullback–Leibler divergence. The ELBO has the complexity $\mathcal{O}(M^2N(D^1 + ... + D^L))$ to compute, D^l is the size of the l^{th} layer. The variational expection likelihood $\mathbb{E}$ in Eq. 6.4 is computed using the Monte Carlo approximation. Please refer to (**salimbeni2017doubly**) for more details.

6.2 Hierarchical Bayesian Model on Manifolds

We define a GAC-GP layer with the GAC kernel and follow the framework of DGPs to construct a hierarchical Bayesian learning model by stacking up multiple GP layers. Thanks to the intra-kernel convolution property, the GAC-GP layer has a good feature aggregation ability. In a pure hierarchical Bayesian learning model on manifolds, a computational pipeline is shown in Figure. 18. The first step is to process the input. Assume κ salient points are selected by Algorithm 5.2.3. The feature on each salient point $f_{\tilde{v}}$ is a vector of length l. For shape i, we link all features of salient points in the order of their selections: $f_i = \{f_{\tilde{v}^1}...f_{\tilde{v}^\kappa}\}$. Noting that all shapes here belong to the same dataset and all salient points are selected with the same parameter setting in Algorithm 5.1.2. Otherwise, point-to-point registration is needed to concatenate features. Suppose H shapes are used, the input X is an $H \times (\kappa \times l)$ matrix. We compose a sequence of layers that map the input x_i to its label y_i in a hierarchical Bayesian model for classifications:

$$\underbrace{x_i = f^0}_{1\times(\kappa\times l)} \overset{\mathcal{GP}_0}{\to} \underbrace{f^1}_{1\times S^1} \to \cdots \overset{\mathcal{GP}_{L-1}}{\to} \underbrace{f^L}_{1\times C} \overset{softmax}{\to} \underbrace{y_i}_{C_i} \tag{6.5}$$

The output of hidden layer l is a vector of the size $1 \times S^l$, where S^l is the layer size. This is similar to the relationship of the input channel and output channel in a NN. When the batch processing is applied, the output of each hidden layer has dimension $B \times S^l$, B is the batch size. A final layer is appended with a softmax multi-class likelihood. The output vector has the dimension $1 \times C$, where C is the number of classes. Each entry stands for the probability belonging to a certain class. Arbitrary numbers of GP layers can be added as hidden layers. We use the doubly stochastic variational inference approach to estimate the posterior (**salimbeni2017doubly**). The optimization process is to maximize the ELBO in Eq. (6.4). The K-means method is used to choose inducing points.

Because the input of the Bayesian model is the point-wise features on manifolds, and NNs are strong in feature learning, we are inspired to further explore the potential of NN+Bayesian methods. Such a mixed model can take advantage of both the feature learning ability of NNs and the feature aggregation ability of Bayesian models. We

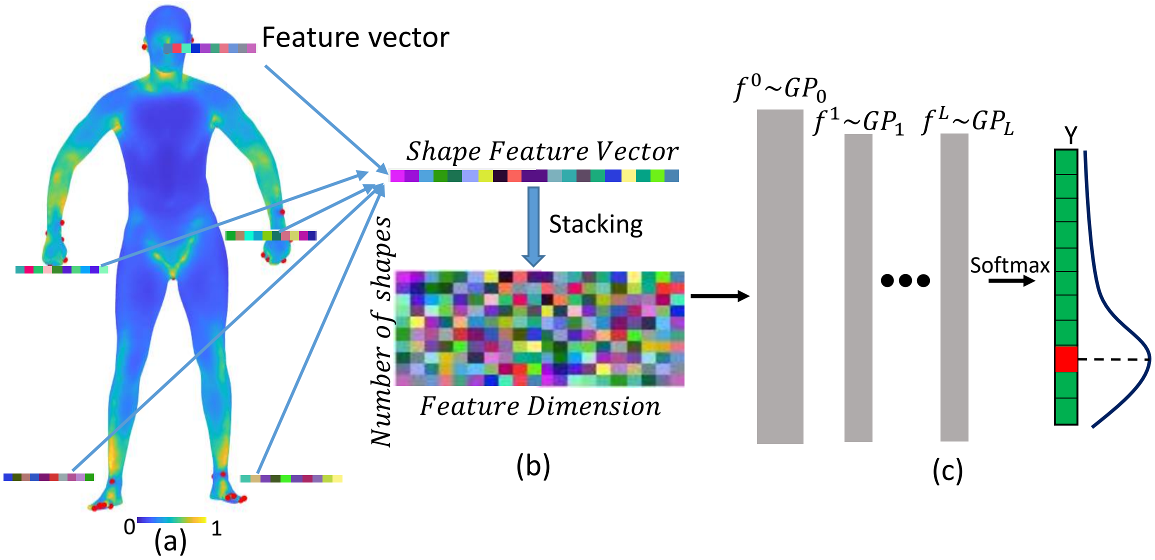


Figure 18. Pipeline of human pose retrieval. (a) Point-wise feature computation and salient point selection. The mesh is rendered by normalized mean curvatures. (b) Shape feature preparation. (c) Hierarchical Bayesian learning model for feature aggregation and inference. A softmax likelihood function is used at the last layer.

generally follow the pipeline in deep kernel learning (**wilson2016deep**). The input format is determined by the NN part. The output of NNs is the shape feature. The feature is then fed into a Bayesian model, and the following processing is the same as the pure Bayesian method. The negative marginal log-likelihood (MLL) is used as the loss function.

6.3 Experiments

6.3.1 Human Pose Retrieval

In the second experiment, the task is to classify different human poses modeled by triangle meshes. The first purpose is to further evaluate the salient point selections by fixing the Bayesian learning architecture. The second purpose is to fix the inputs

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

Table 7. Results of human pose retrieval with Bayesian models defined by different kernels and numbers of salient points.

and evaluate different hierarchical Bayesian learning architectures. The pipeline in Figure. 18 is used. We choose the SIWKS as the feature. When computing the SIWKS, 30 smallest eigenvalues are used. The other parameter settings are the same as those in (Aubry, Schlickewei, and Cremers 2011). Features of 50, 100, and 250 salient points are used. In customizing the Bayesian model, we use the multitask variational strategy and the softmax likelihood in GPytorch. The number of inducing points is 50. We use Adam as the optimizer with an initial learning rate of 0.001. After the first 200 epochs, the learning rate changes to 10^{-4}. We trained for 2000 epochs. The cost function is the variational ELBO mentioned in Sec 6.1. The comparison methods are the same as the prior experiment.

The SHREC14 non-rigid 3D human model is used as the dataset (Pickup et al. 2014). It contains 400 triangle meshes of 40 human subjects with 10 poses. Each mesh contains about 15000 vertices. We randomly split the dataset into three portions: 90% for training, 5% for validation, and 5% for testing.

For the first purpose, we fix the Bayesian model to be a one-layer GAC-GP and feed in features from different methods. Table 7 shows the results. Classification with the features of all points has an accuracy of 0.910. Taking this value as a reference, we can draw conclusions that (1) our strategy of selecting salient points works for distinguishing different shapes. When enough salient points are selected, it is possible

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

*(Using self-reproduced code.)

Table 8. Human pose retrieval with different Bayesian learning architectures.

to use a small subset to represent the original data; (2) the geometry-aware selection of GAC-GP is more distinguishable than other comparison methods.

For the second purpose, we fix the inputs to be GAC-GP salient features and evaluate different Bayesian learning architectures. Here we use GCGP (**walker2019graph**) as a comparison method. The results are shown in Table. 8. Noting that the code of GCGP is not available and we use our implementations, so we put a star mark on GCGP's result. We can see that the accuracy is generally increased after adding a GAC layer, supporting a strong feature aggregation property. Meanwhile, a hierarchical concatenation of GAC layers shows a better accuracy than the single layer structure.

6.3.2 Point Cloud Classification

In the third experiment, the task is to classify different point cloud models. The purpose of this experiment is to demonstrate the work of integrating NNs with the Bayesian model. Here, we use the hierarchical feature learning architecture in PointNet++ (**qi2017pointnet++**) to learn the point cloud features. We perform multi-class classification on ModelNet40 which contains 12311 3D CAD models of 40 categories. Each point cloud has 10000 points. We use 9843 models for training and 2468 models for testing. The setting of feature extraction part is the same

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

Table 9. Multi-class classifications on ModelNet40.

as (**qi2017pointnet++**). In the feature aggregation part, we use one single GAC-GP layer (ten mixtures). 64 inducing points are used. The optimizer is Adam and the initial learning rate is 0.04. The comparison methods include PointNet++, PointNet++ with normal information, and the Pointwise Convolutional NNs (PCNN) in (**hua2018pointwise**). The results in Table 9 show that (1) the mechanism of NN+Bayesian can be jointly trained for tasks on manifolds; (2) models with Bayesian aggregation layers generally outperforms the classical multiple fully connected layers in our tests. We notice that the performance gain of using single GAC layer shrinks after adding normal information. Our hypothesis is that the features become more complicated, and the inference capability of single GAC layer is not powerful enough to well aggregate the new features. By adding 2&3 GAC layers, the improvements increase to 0.9% and 1.2%, respectively.

The overall results demonstrate that the performances of architectures with GAC layers are universally better than their original ones, which also proves that such a co-design directly incorporates the strengths of both approaches and benefits the final performance. A reasonable outlook is to investigate more effective architectures that integrate both methods for end-to-end tasks on manifolds.

Chapter 7

DISCUSSION

7.1 Summary of Contributions

The main contribution of this thesis was to exploit the potential of different SPDEs for effective and efficient shape analysis. The high-dimensional shape analysis was firstly studied. In Chapter 3, a new pipeline of generating cortical tetrahedral mesh was proposed. In Chapter 4, several shape descriptors was introduced. The FEM discretization was derived for tetrahedral meshes as the DLBOD scheme. Based on the solution of the heat equation and the eigen decomposition of the Schrödinger's equation, the single tHFS and the grouped tHFS were able to be defined for high-dimensional shapes. The applications around DLBOD scheme include cortical thickness computation and cortical atrophy severity classifications. Furthermore, a morphological GP was introduced for realizing reasonable landmarking on tetrahedral meshes in Chapter 5. The MGP can be taken as a high-level application of the above shape descriptors because the metric in MGP is the feature space defined by SIWKS. Considering the limitations of MGP, a more universal GP method, the GAC kernel was derived from a specific SPDE. The GAC-GP realized hand-craft feature-free geometry-aware for the first time. Around the GAC kernel, a series of Bayesian methods was proposed. A hierarchical Bayesian network was introduced to realize a deep GP on manifolds. The NN+Bayesian architecture was further introduced to embed the advantages of both NN and Bayesian methods together.

7.2 Bayesian Networks vs. Neural Networks

The superior performances of convolutional neural networks (CNNs) in solving computer vision problems are closely related to the adaptive convolutional layers (**voulodimos2018deep**). This motivates the studies of introducing convolutional structures to Gaussian processes (GPs) for pursuing an expressive global generalizations of vision data in a Bayesian framework (Van der Wilk, Rasmussen, and Hensman 2017; Blomqvist, Kaski, and Heinonen 2018). Meanwhile, the theoretical feasibility of taking GPs as neural networks under certain conditions is proved (Neal 1996; Williams and Rasmussen 2006). Considering a fully-connected neural network with i.i.d. random parameters, each single fully-connected layer is a collection of adaptive functions (Lee et al. 2017). The training of a neural network is to find optimized parameters of adaptive functions so that a prescribed cost function is minimized. Assuming the hidden layers have an infinite width, the Central Limit Theorem indicates that the learned adaptive functions and the output y form a GP (Neal 1996; Williams 1997). This implies priors over parameters of a neural network are analogous to priors of a GP and the output is a Bayesian inference result (Lee et al. 2017).

Above information indicates the connections between Bayesian networks and neural networks. But they also have their uniquenesses. NNs are strong in learning features of the input. This is one of the reasons why the NNs are widely used in various tasks currently. Bayesian network has limited ability of learning features directly. Therefore, the input of Bayesian networks is the feature of raw data. This limitation causes the application of Bayesian network is restrained. Besides, NNs are much easier to be created, manipulated, trained and applied while the application of Bayesian network is still narrower and difficult. Training a Bayesian network is usually slower than that of a NN. Building a deep NN with several layers is not a problem in nowadays, but building a Bayesian network with more than 5 layers still has many problems. How to increase the training efficiency of a deep Bayesian network is still an open problem.

Overall, the current Bayesian network is not competitive with NNs in many tasks over performances. But it is reasonable and feasible to combine both methods together and yield a better result than only using either single one of them.

7.3 Summary of Conclusions

It shows clearly that many achievements in applied mathematics and physics are providing inspirations to many studies in other fields. In this thesis, different SPDEs play an important role in discovering expressive features, metrics, descriptors and kernels. Interdisciplinary information could provide broader and different perspectives and sources for the new findings of the problem. Useful hints may hide in different scenarios and wait for us to dig. This is also what I wish to reflect in this thesis. Hope this thesis could make contributions and give inspirations to the studies in this or other fields.

NOTES

Notations: Suppose $M = (V, E, T)$ is a tetrahedral mesh; V is the vertex list; E is the edge list; T is the tetrahedron list; v_i is the i^{th} vertex; t_l is the l^{th} tetrahedron; V_l is the volume of the l^{th} tetrahedron; E(i,j) is the edge defined by the i^{th} and j^{th} vertices. H is the heat distribution; h is the static heat value. $\mathfrak{L}$ is the LBO; λ_n and ϕ_n are the n^{th} eigenvalue and eigenvector. $N(\cdot)$ is the open neighborhood function; $N[\cdot]$ the closed neighborhood function. Some other definitions will be introduced where they are defined.

Data is downloaded from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database ((Susanne G Mueller et al. 2005), adni.loni.usc.edu). ADNI is the result of efforts of many co-investigators from a broad range of academic institutions and private corporations. Subjects have been recruited from over 50 sites across the U.S. and Canada. The primary goal of ADNI is to test whether biological markers, such as serial MRI and positron emission tomography (PET), combined with clinical and neuropsychological assessments, can measure the progression of MCI and early AD. Subjects originally recruited for ADNI-1 and ADNI-GO had the option to be followed in ADNI-2. For up-to-date information, see www.adniinfo.org.

In this work, we adopt the ADNI-2 cohort, for which subjects' structural MRI data was collected from 16 sites across the United States and Canada. Each subject underwent whole-brain MRI scanning on 3-Tesla GE Medical Systems scanners. T1-weighted SPGR (spoiled gradient echo) sequences (256 × 256 matrix; voxel size = $1.2 \times 1.0 \times 1.0\ mm^3$; TI = 400 ms; TR = 6.98 ms; TE = 2.85 ms; flip angle = 11°), were collected; more imaging details may be found at http://adni.loni.usc.edu/wp-content/uploads/2010/05/ADNI2_GE_3T_22.0_T2.pdf. In total, we downloaded MR images of 542 subjects from the baseline subset of ADNI-2 (June 2016). During processing, seven subjects were excluded by the quality control. Finally, we used the left cerebral hemisphere structural MR images of 535 subjects, including 105 AD patients, 289 MCI patients and 141 cogntively unimpaired (CU) subjects. Their demographic information is summarized in Table 10.

Table 10. Demographic characteristics of the baseline ADNI-2 used in this work.

	AD (n=105)	MCI (n=289)	CU (n=141)
Male/Female	58/47	154/135	69/72
Age	74.6±7.8	71.6±7.6	73.4±6.5
Education	15.8±2.7	16.4±2.6	16.6±2.5
MMSE	23±2.1	28.0±1.7	29.1±1.2

REFERENCES

Ashburner, John, and Karl J Friston. 2000. “Voxel-based morphometry—the methods.” *Neuroimage* 11 (6): 805–821.

Aubry, Mathieu, Ulrich Schlickewei, and Daniel Cremers. 2011. “The wave kernel signature: A quantum mechanical approach to shape analysis.” In *IEEE International Conference on Computer Vision Workshops,* 1626–1633.

Baron, JC, G Chetelat, B Desgranges, G Perchey, B Landeau, V De La Sayette, and F Eustache. 2001. “In vivo mapping of gray matter loss with voxel-based morphometry in mild Alzheimer’s disease.” *Neuroimage* 14 (2): 298–309.

Bas-Hoogendam, Janna Marie, Henk van Steenbergen, J Nienke Pannekoek, Jean-Paul Fouche, Christine Lochner, Coenraad J Hattingh, Henk R Cremers, Tomas Furmark, Kristoffer NT Månsson, Andreas Frick, et al. 2017. “Voxel-based morphometry multi-center mega-analysis of brain structure in social anxiety disorder.” *NeuroImage: Clinical* 16:678–688.

Berline, Nicole, Ezra Getzler, and Michele Vergne. 2003. *Heat kernels and Dirac operators.* Springer Science & Business Media.

Bhatia, Rajendra. 2007. “Positive Definite Matrices, Princeton Ser.” *Appl. Math., Princeton University Press, Princeton, NJ.*

Blomqvist, Kenneth, Samuel Kaski, and Markus Heinonen. 2018. “Deep convolutional Gaussian processes.” *arXiv preprint arXiv:1810.03052.*

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

Bookstein, Fred L. 2001. “Voxel-based morphometry should not be used with imperfectly registered images.” *Neuroimage* 14 (6): 1454–1462.

Boukhayma, Adnane, Jean-Sébastien Franco, and Edmond Boyer. 2017. “Surface motion capture transfer with gaussian process regression.” In *IEEE Conference on Computer Vision and Pattern Recognition,* 9.

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

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

Bronstein, Michael M, and Iasonas Kokkinos. 2010. “Scale-invariant heat kernel signatures for non-rigid shape recognition.” In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 1704–1711. IEEE.

Burmeister, Louis C. 1993. *Convective heat transfer.* John Wiley & Sons.

Castillo, Ismaël, Gérard Kerkyacharian, and Dominique Picard. 2014. “Thomas Bayes’ walk on manifolds.” *Probability Theory and Related Fields* 158 (3-4): 665–710.

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

Cuingnet, R., E. Gerardin, J. Tessieras, G. Auzias, S. Lehericy, M. O. Habert, M. Chupin, H. Benali, and O. Colliot. 2011. “Automatic classification of patients with Alzheimer’s disease from structural MRI: a comparison of ten methods using the ADNI database.” *Neuroimage* 56, no. 2 (May): 766–781.

Damianou, Andreas, and Neil Lawrence. 2013. “Deep gaussian processes.” In *Artificial Intelligence and Statistics,* 207–215.

Davatzikos, Christos. 2004. “Why voxel-based morphometric analysis should be used with great caution when characterizing group differences.” *Neuroimage* 23 (1): 17–20.

Delkhosh, Mehdi, Mohammad Delkhosh, and Mohsen Jamali. 2012. “Green’s Function and its Applications.” *J. Basic. Appl. Sci. Res.* 2 (9): 8865–76.

Di Martino, A., C. G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, et al. 2014. “The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism.” *Mol. Psychiatry* 19, no. 6 (June): 659–667.

Draganski, B, C Gaser, V Busch, U Schuierer Gand Bogdahn, and A May. 2004. “Neuroplasticity: changes in grey matter induced by training.” *Nature* 427 (6972): 311.

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

Fan, Yonghui, Natasha Lepore, and Yalin Wang. 2020. “Morphometric Gaussian process for landmarking on grey matter tetrahedral models.” In *15th International Symposium on Medical Information Processing and Analysis,* vol. 11330, 113300H. International Society for Optics and Photonics.

Fan, Yonghui, Gang Wang, Qunxi Dong, Yuxiang Liu, Natasha Lepore, Yalin Wang, Alzheimer’s Disease Neuroimaging Initiative, et al. 2021. “Tetrahedral Spectral Feature-Based Bayesian Manifold Learning for Grey Matter Morphometry: Findings from the Alzheimer’s Disease Neuroimaging Initiative.” *Medical Image Analysis,* 102123.

Fan, Yonghui, Gang Wang, Natasha Leporé, and Yalin Wang. 2018. “A Tetrahedron-based Heat Flux Signature for Cortical Thickness Morphometry Analysis.” *Med Image Comput Comput Assist Interv* 11072 (September): 420–428.

Fan, Yonghui, and Yalin Wang. 2020. “Convolutional Bayesian Models for Anatomical Landmarking on Multi-dimensional Shapes.” In *International Conference on Medical Image Computing and Computer-Assisted Intervention,* 786–796. Springer.

Fischl, B., and A. M. Dale. 2000. “Measuring the thickness of the human cerebral cortex from magnetic resonance images.” *Proc. Natl. Acad. Sci. U.S.A.* 97, no. 20 (September): 11050–11055.

Fischl, Bruce, David H Salat, Evelina Busa, Marilyn Albert, Megan Dieterich, Christian Haselgrove, Andre Van Der Kouwe, Ron Killiany, David Kennedy, Shuna Klaveness, et al. 2002. “Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain.” *Neuron* 33 (3): 341–355.

Freund, Yoav, Robert Schapire, and Naoki Abe. 1999. “A short introduction to boosting.” *Journal-Japanese Society For Artificial Intelligence* 14 (771-780): 1612.

Gao, Tingran, Shahar Z Kovalsky, Doug M Boyer, and Ingrid Daubechies. 2018. “Gaussian Process Landmarking on Manifolds.” *arXiv preprint arXiv:1802.03479.*

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

Gaonkar, Bilwaj, Kilian Pohl, and Christos Davatzikos. 2011. “Pattern based morphometry.” In *International Conference on Medical Image Computing and Computer-Assisted Intervention,* 459–466. Springer.

Ge, Tian, Martin Reuter, Anderson M Winkler, Avram J Holmes, Phil H Lee, Lee S Tirrell, Joshua L Roffman, Randy L Buckner, Jordan W Smoller, and Mert R Sabuncu. 2016. "Multidimensional heritability analysis of neuroanatomical shape." *Nature communications* 7:13291.

Guennebaud, Gaël, Marcel Germann, and Markus Gross. 2008. "Dynamic sampling and rendering of algebraic point set surfaces." In *Computer Graphics Forum,* 27:653–662. 2. Wiley Online Library.

Guhaniyogi, Rajarshi, and David B Dunson. 2016. "Compressed Gaussian process for manifold regression." *The Journal of Machine Learning Research* 17 (1): 2472–2497.

Hartikainen, Jouni, and Simo Särkkä. 2010. "Kalman filtering and smoothing solutions to temporal Gaussian process regression models." In *2010 IEEE International Workshop on Machine Learning for Signal Processing,* 379–384. IEEE.

Jack Jr, Clifford R, Matt A Bernstein, Nick C Fox, Paul Thompson, Gene Alexander, Danielle Harvey, Bret Borowski, Paula J Britson, Jennifer L. Whitwell, Chadwick Ward, et al. 2008. "The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods." *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine* 27 (4): 685–691.

Kichenassamy, Satyanad, Arun Kumar, Peter Olver, Allen Tannenbaum, and Anthony Yezzi. 1995. "Gradient flows and geometric active contour models." In *Proceedings of IEEE International Conference on Computer Vision,* 810–815. IEEE.

Kluger, B. M., Q. Zhao, J. J. Tanner, N. A. Schwab, S. A. Levy, S. E. Burke, H. Huang, M. Ding, and C. Price. 2019. "Structural brain correlates of fatigue in older adults with and without Parkinson's disease." *Neuroimage Clin* 22:101730.

Kondor, Risi Imre, and John Lafferty. 2002. "Diffusion kernels on graphs and other discrete structures." In *Proceedings of the 19th international conference on machine learning,* 2002:315–322.

Labayru, G., I. Diez, J. Sepulcre, E. Fernandez, M. Zulaica, J. M. Cortes, A. Lopez de Munain, and A. Sistiaga. 2019. "Regional brain atrophy in gray and white matter is associated with cognitive impairment in Myotonic Dystrophy type 1." *Neuroimage Clin* 24 (November): 102078.

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

Lee, Jaehoon, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. 2017. "Deep neural networks as gaussian processes." *arXiv preprint arXiv:1711.00165.*

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

Lin, A., C. R. K. Ching, A. Vajdi, D. Sun, R. K. Jonas, M. Jalbrzikowski, L. Kushan-Wells, et al. 2017. "Mapping 22q11.2 Gene Dosage Effects on Brain Morphometry." *J. Neurosci.* 37, no. 26 (June): 6183–6199.

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

Liu, Jun, Shuiwang Ji, Jieping Ye, et al. 2009. "SLEP: Sparse learning with efficient projections." *Arizona State University* 6 (491): 7.

Liu, W., J. Gan, J. Fan, H. Zheng, S. Li, R. C. K. Chan, C. Tan, and X. Zhu. 2019. "Associations of cortical thickness, surface area and subcortical volumes with insight in drug-naïve adults with obsessive-compulsive disorder." *Neuroimage Clin* 24:102037.

Lorenzi, Marco, Gabriel Ziegler, Daniel C Alexander, and Sebastien Ourselin. 2015. "Efficient Gaussian process-based modelling and prediction of image time series." In *International Conference on Information Processing in Medical Imaging,* 626–637. Springer.

Mallasto, Anton, Aasa Feragen, et al. 2018. "Wrapped Gaussian process regression on Riemannian manifolds." *CSGB Research Reports, Department of Mathematics, Aarhus University.*

Mateos, M. J., A. Gastelum-Strozzi, F. A. Barrios, E. Bribiesca, S. Alcauter, and J. A. Marquez-Flores. 2019. "A novel voxel-based method to estimate cortical sulci width and its application to compare patients with Alzheimer's disease to controls." *Neuroimage* (November): 116343.

Mueller, S. G., M. W. Weiner, L. J. Thal, R. C. Petersen, C. Jack, W. Jagust, J. Q. Trojanowski, A. W. Toga, and L. Beckett. 2005. "The Alzheimer's disease neuroimaging initiative." *Neuroimaging Clin. N. Am.* 15, no. 4 (November): 869–877.

Mueller, Susanne G, Michael W Weiner, Leon J Thal, Ronald C Petersen, Clifford Jack, William Jagust, John Q Trojanowski, Arthur W Toga, and Laurel Beckett.

2005. “The Alzheimer’s disease neuroimaging initiative.” *Neuroimaging Clinics* 15 (4): 869–877.

Neal, Radford M. 1996. “Priors for infinite networks.” In *Bayesian Learning for Neural Networks,* 29–53. Springer.

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

Penny, William D, Karl J Friston, John T Ashburner, Stefan J Kiebel, and Thomas E Nichols. 2011. *Statistical parametric mapping: the analysis of functional brain images.* Elsevier.

Pickup, David, X Sun, Paul L Rosin, RR Martin, Z Cheng, Z Lian, M Aono, A Ben Hamza, A Bronstein, M Bronstein, et al. 2014. “SHREC’14 track: Shape retrieval of non-rigid 3D human models.” In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval,* 1:6. 2. Eurographics Association.

Pini, L., M. Pievani, M. Bocchetta, D. Altomare, P. Bosco, E. Cavedo, S. Galluzzi, M. Marizzoni, and G. B. Frisoni. 2016. “Brain atrophy in Alzheimer’s Disease and aging.” *Ageing Res. Rev.* 30 (September): 25–48.

Pini, Lorenzo, Michela Pievani, Martina Bocchetta, Daniele Altomare, Paolo Bosco, Enrica Cavedo, Samantha Galluzzi, Moira Marizzoni, and Giovanni B Frisoni. 2016. “Brain atrophy in Alzheimer’s disease and aging.” *Ageing research reviews* 30:25–48.

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

Rasmussen, Carl Edward. 2004. “Gaussian processes in machine learning.” In *Advanced lectures on machine learning,* 63–71. Springer.

Reuter, Martin, Silvia Biasotti, Daniela Giorgi, Giuseppe Patanè, and Michela Spagnuolo. 2009. “Discrete Laplace–Beltrami operators for shape analysis and segmentation.” *Computers & Graphics* 33 (3): 381–390.

Rustamov, Raif M. 2007. “Laplace-Beltrami eigenfunctions for deformation invariant shape representation.” In *Proceedings of the fifth Eurographics symposium on Geometry processing,* 225–233. Eurographics Association.

Särkkä, Simo. 2011. "Linear operators and stochastic partial differential equations in Gaussian process regression." In *International Conference on Artificial Neural Networks,* 151–158. Springer.

Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller. 1998. "Nonlinear component analysis as a kernel eigenvalue problem." *Neural computation* 10 (5): 1299–1319.

Shi, J., O. Collignon, L. Xu, G. Wang, Y. Kang, F. Lepore, Y. Lao, A. A. Joshi, N. Lepore, and Y. Wang. 2015. "Impact of Early and Late Visual Deprivation on the Structure of the Corpus Callosum: A Study Combining Thickness Profile with Surface Tensor-Based Morphometry." *Neuroinformatics* 13, no. 3 (July): 321–336.

Shi, J., P. M. Thompson, B. Gutman, and Y. Wang. 2013. "Surface fluid registration of conformal representation: application to detect disease burden and genetic influence on hippocampus." *Neuroimage* 78 (September): 111–134.

Shi, R., W. Zeng, Z. Su, J. Jiang, H. Damasio, Z. Lu, Y. Wang, S. T. Yau, and X. Gu. 2017. "Hyperbolic Harmonic Mapping for Surface Registration." *IEEE Trans Pattern Anal Mach Intell* 39, no. 5 (May): 965–980.

Shi, Yan, and Chi Hou Chan. 2010. "Multilevel Green's function interpolation method for analysis of 3-D frequency selective structures using volume/surface integral equation." *JOSA A* 27 (2): 308–318.

Si, Hang. 2015. "TetGen, a Delaunay-based quality tetrahedral mesh generator." *ACM Transactions on Mathematical Software (TOMS)* 41 (2): 11.

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

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

Thompson, P. M., K. M. Hayashi, G. de Zubicaray, A. L. Janke, S. E. Rose, J. Semple, D. Herman, et al. 2003. "Dynamics of gray matter loss in Alzheimer's disease." *J. Neurosci.* 23, no. 3 (February): 994–1005.

Tuokkola, T., M. Karrasch, J. Koikkalainen, R. Parkkola, J. Lotjonen, E. Loyttyniemi, S. Hurme, and J. O. Rinne. 2019. "Association between Deep Gray Matter Changes and Neurocognitive Function in Mild Cognitive Impairment and Alzheimer's

Disease: A Tensor-Based Morphometric MRI Study." *Dement Geriatr Cogn Disord* 48 (1-2): 68–78.

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

Verma, Ragini, Susumu Mori, Dinggang Shen, Paul Yarowsky, Jiangyang Zhang, and Christos Davatzikos. 2005. "Spatiotemporal maturation patterns of murine brain quantified by diffusion tensor MRI and deformation-based morphometry." *Proceedings of the national academy of sciences of the United States of America* 102 (19): 6978–6983.

Wachinger, C., P. Golland, M. Reuter, and W. Wells. 2014. "Gaussian process interpolation for uncertainty estimation in image registration." *Med Image Comput Comput Assist Interv* 17 (Pt 1): 267–274.

Wang, G., and Y. Wang. 2017. "Towards a Holistic Cortical Thickness Descriptor: Heat Kernel-Based Grey Matter Morphology Signatures." *Neuroimage* 147 (February): 360–380.

Wang, Gang, and Yalin Wang. 2017. "Towards a Holistic Cortical Thickness Descriptor: Heat Kernel-Based Grey Matter Morphology Signatures." *Neuroimage* 147 (February): 360–380.

Wang, Yalin, Xianfeng Gu, Shing-Tung Yau, et al. 2003. "Volumetric harmonic map." *Communications in Information & Systems* 3 (3): 191–202.

Wang, Yalin, Jie Shi, Xiaotian Yin, Xianfeng Gu, Tony F Chan, Shing-Tung Yau, Arthur W Toga, and Paul M Thompson. 2012. "Brain surface conformal parameterization with the Ricci flow." *IEEE transactions on medical imaging* 31 (2): 251–264.

Watkins, K. E., D. G. Gadian, and F. Vargha-Khadem. 1999. "Functional and structural brain abnormalities associated with a genetic disorder of speech and language." *Am. J. Hum. Genet.* 65, no. 5 (November): 1215–1221.

Whitwell, Jennifer L. 2009. "Voxel-based morphometry: an automated technique for assessing structural changes in the brain." *Journal of Neuroscience* 29 (31): 9661–9664.

Williams, Christopher KI. 1997. "Computing with infinite networks." In *Advances in neural information processing systems,* 295–301.

Williams, Christopher KI, and Carl Edward Rasmussen. 2006. *Gaussian processes for machine learning.* Vol. 2. 3. MIT press Cambridge, MA.

Wilson, Andrew, and Ryan Adams. 2013. "Gaussian process kernels for pattern discovery and extrapolation." In *International Conference on Machine Learning,* 1067–1075.

Wilson, Andrew Gordon. 2014. "Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes." PhD diss., University of Cambridge.

Wolters, A. F., A. J. H. Moonen, R. Lopes, A. F. G. Leentjens, A. A. Duits, L. Defebvre, C. Delmaire, P. A. Hofman, F. C. van Bussel, and K. Dujardin. 2019. "Grey matter abnormalities are associated only with severe cognitive decline in early stages of Parkinson's disease." *Cortex* 123 (October): 1–11.

Xia, J., F. Wang, Z. Wu, L. Wang, C. Zhang, D. Shen, and G. Li. 2020. "Mapping hemispheric asymmetries of the macaque cerebral cortex during early brain development." *Hum Brain Mapp* 41, no. 1 (January): 95–106.

Xiao, X., H. Fang, J. Wu, C. Xiao, T. Xiao, L. Qian, F. Liang, Z. Xiao, K. K. Chu, and X. Ke. 2017. "Diagnostic model generated by MRI-derived brain features in toddlers with autism spectrum disorder." *Autism Res* 10, no. 4 (April): 620–630.

Zhu, Jianke, Steven CH Hoi, and Michael R Lyu. 2009. "Nonrigid shape recovery by Gaussian process regression." In *IEEE Conference on Computer Vision and Pattern Recognition,* 1319–1326. IEEE.

APPENDIX A

DISCRETIZATION OF MASS MATRIX B

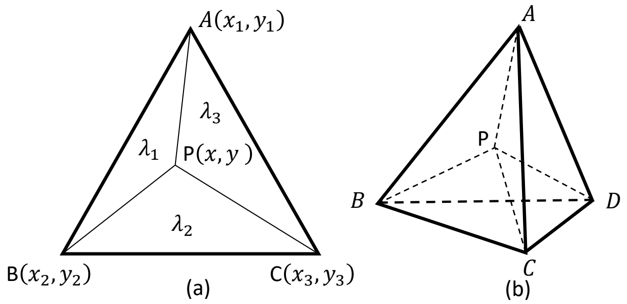


Figure 19. Barycentric coordinates example.

A.1 Discretization of Mass Matrix B

Defining mass matrix B based on finite element analysis is implemented through the barycentric coordinate integral formulation (**cook2007concepts**). More exactly, defining mass matrix B is to solve a coefficient function at each entry in the matrix and this coefficient function is the integrands in a simplex region. For convenience, we start from a fundamental case, a typical triangle on 2-dimensional manifold as illustrated in Fig. 19. Suppose $P(x, y)$ is an arbitrary point in this triangle and it is expressed as $(\lambda_1, \lambda_2, \lambda_3)$ in a barycentric coordinate system. λ_1, λ_2 and λ_3 stand for the area ratio between the splited triangle and the whole region, for example,λ_1 is the area of $\Delta(PAB)$ divided by the area of $\Delta(ABC)$. Clearly, the sum of three coordinates is 1, so any barycentric coordinate can be expressed as the residual of 1 minus the sum of other two coordinates, such as: $\lambda_3 = 1 - \lambda_1 - \lambda_2$. The relationship between the barycentric coordinates and the spatial coordinates is:

$$\begin{cases} x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 \\ y = \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3 \end{cases} \Rightarrow \begin{cases} x = \lambda_1 x_1 + \lambda_2 x_2 + (1 - \lambda_1 - \lambda_2) x_3 \\ y = \lambda_1 y_1 + \lambda_2 y_2 + (1 - \lambda_1 - \lambda_2) y_3 \end{cases} \tag{A.1}$$

The coefficient function is expressed as the integral over all triangular elements:

$$I = \int_\mu \lambda_1^m \lambda_1^n \lambda_1^q d\mu \tag{A.2}$$

where μ remarks the area of the triangle; m, n, q is the order of each coordinate, the maximum order is the number of vertices. According to Eq. A.1 and A.2,

$$\begin{cases} \int_\mu \lambda_1^m \lambda_1^n \lambda_1^q d\mu = \int_\mu \lambda_1^m \lambda_1^n (1 - \lambda_1 - \lambda_2)^q d\mu \\ \qquad d\mu = det(\frac{\partial(x,y)}{\partial(\lambda_1, \lambda_2)}) d\lambda_1 d\lambda_2. \end{cases} \tag{A.3}$$

The area of the triangle can be expressed by using the determinant of the Jacobian matrix.

$$det\begin{pmatrix}\frac{\partial x}{\partial \lambda_1} & \frac{\partial x}{\partial \lambda_2}\\ \frac{\partial y}{\partial \lambda_1} & \frac{\partial y}{\partial \lambda_2}\end{pmatrix} = \begin{bmatrix}x_1 - x_3 & x_2 - x_3\\ y_1 - y_3 & y_2 - y_3\end{bmatrix} = 2A(\mu) \tag{A.4}$$

We can rewrite the integral as:

$$\int_\mu \lambda_1^m \lambda_1^n \lambda_1^q d\mu = 2A\int_0^1 d\lambda_2 \int_0^{1-\lambda_2} \lambda_1^m \lambda_2^n (1-\lambda_1-\lambda_2)^q d\lambda_1 \tag{A.5}$$

Define t as:

$$t = \frac{\lambda_1}{1-\lambda_2} \rightarrow \lambda_1 = t(1-\lambda_2) \tag{A.6}$$

Further derive the integral:

$$\begin{aligned}\int_\mu \lambda_1^m \lambda_1^n \lambda_1^q d\mu &= 2A\int_0^1 d\lambda_2 \int_0^1 \lambda_1^m \lambda_2^n (1-\lambda_1-\lambda_2)^q (1-\lambda_2) dt\\ &= 2A\int_0^1 d\lambda_2 \int_0^1 t^m (1-\lambda_2)^m \lambda_2^n (1-t)^q (1-\lambda_2)^q (1-\lambda_2) dt\\ &= 2A\int_0^1 \lambda_2^n (1-\lambda_2)^{m+q+1} d\lambda_2 \int_0^1 t^m (1-t)^q dt\end{aligned} \tag{A.7}$$

According to Euler formula:

$$\int_0^1 s^\alpha (1-s)^\beta ds = \frac{\alpha!\beta!}{(\alpha+\beta+1)!} \tag{A.8}$$

The integral can be further derived:

$$\begin{aligned}\int_\mu \lambda_1^m \lambda_1^n \lambda_1^q d\mu &= 2A\frac{n!(m+q+1)!}{(m+n+q+2)!}\frac{m!q!}{(m+q+1)!}\\ &= 2A\frac{m!n!q!}{(m+n+q+2)!}\end{aligned} \tag{A.9}$$

The coefficients are computed by considering the orders under certain scenarios. An example of implementing this method on triangle meshes is provided in (**shi2014metric**). The tetrahedron is on 3-dimensional manifold, therefore, a simplex region contains 4 vertices as illustrated in Fig. 19(b). The integral over tetrahedral elements is defined as $\int_\Omega \lambda_1^i \lambda_2^j \lambda_3^k \lambda_4^l d\Omega$, where Ω is the volume of the tetrahedron. Similarly, i, j, k, l is the order of the barycentric coordinates $\lambda_1, \lambda_2, \lambda_3, \lambda_4$, respectively and their maximum is 4. By adding one more dimension and repeating above derivations in the same way, the integral over all tetrahedral elements is computed as:

$$\int_\Omega \lambda_1^i \lambda_2^j \lambda_3^k \lambda_4^l d\Omega = \frac{i!j!k!l!}{(i+j+k+l+3)!}6V \tag{A.10}$$

Now considering the explicit coefficient on each entry of the matrix. The matrix reflects the adjacent relationship among vertices, thus, the type of connections to a target vertex includes: non-adjacency, edge and vertex itself. For non-adjacent vertices, the coefficient is set to be zero. There are two cases when the entry stands for the vertex itself (this is actually the diagonal of the matrix). The first case is that the arbitrary point only falls on the target vertex. Then one vertex reaches the highest order 4, and the other three vertices are not involved. In this case, only tetrahedrons around the vertex are considered. The coefficient is computed as:

$$\int_\Omega \lambda_1^4 d\Omega = \frac{4!}{7!} \cdot 6V = \frac{V}{35} \tag{A.11}$$

The second case is that the arbitrary point falls on the vertex that connects to the target vertex. In this case, we traverse all the tetrahedrons that connect with edges involving the target vertex. The coefficient under such situation is computed as:

$$\int_\Omega \lambda_1^3 \lambda_1 d\Omega = \frac{3!}{7!} \cdot 6V = \frac{V}{140} \tag{A.12}$$

When one vertex forms an edge with the target vertex, there are also two cases. The first case is that one more vertex connecting to this edge is considered. In this case, the target vertex has an order 2 and these three vertices form a face. We consider all faces with the target vertex as one endpoint and accumulate all tetrahedrons that share with each face. The coefficient under this situation is computed as:

$$\int_\Omega \lambda_1^2 \lambda_2^1 \lambda_3^1 d\Omega = \frac{2!}{7!} \cdot 6V = \frac{V}{420} \tag{A.13}$$

Finally, all coordinates have order one. In this case, we consider all tetrahedrons involving this edge and accumulate all tetrahedrons neighboring with each tetrahedron. The coefficient is computed as:

$$\int_\Omega \lambda_1^1 \lambda_2^1 \lambda_3^1 \lambda_4^1 d\Omega = \frac{1!}{7!} \cdot 6V = \frac{V}{840} \tag{A.14}$$

We also note that another method for finite element analysis is the unit coordinate integration. It has one less dimension comparing with the barycentric coordinate integration:

$$\int_\Omega \lambda_1^i \lambda_2^j \lambda_3^k d\Omega = \frac{i!j!k!}{(i+j+k+3)!} 6V \tag{A.15}$$

An example of this method on triangle meshes is shown in (Reuter et al. 2009). We tested both methods and found the barycentric coordinate integration is slightly better than the unit coordinate integration regarding to the classification performance though the computation of the mass matrix B is easier. Therefore, we do not provide further explanations about the unit coordinate integration and its implementations here.

APPENDIX B

THEOREM1 AND PROOF

B.1 Theorem1 and Proof

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

Proof. As known, there exists a corresponding Green's function $G(v,v',t,t')$ for a parabolic partial differential equation so that the diffusion process expressed by this parabolic partial differential equation has the form (**ehrlich1980surface**; **strauss2013partielle**):

$$T(v,v',t) = \int_0^t G(v,v',t-s)P(s)ds \tag{B.1}$$

In the main submission we have proved that the analytical solution to the following equation is PSD:

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

Since the spatial variable mainly exists in the Green's function and the Green's function is spatial stationary in a $\mathcal{R}^d$ diffusion process, the primary task is to prove any choices of periodic function $P(t)$ can lead to the same conclusion. Because Eq. (B.2) has been prove to be PSD, we can draw the same conclusion if Eq. (B.1) has a similar form with Eq. (B.2). Therefore, the main idea is using cosine function to generalize a periodic function $P : \mathbb{R} \to \mathbb{R}$. As known, a periodic function can be estimated with the Fourier series expansion:

$$P(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} [a_n cos(nt) + b_n sin(nt)] \tag{B.3}$$

where a, b are arbitrary real numbers. Assuming there exists a cosine function $x_n cos(nt + y_n)$ that is equal to $a_n cos(nt) + b_n sin(nt)$, n is a constant. By using the trigonometric sum formulae, we get: $x_n = \pm\sqrt{a_n^2 + b_n^2}$ and $y = arctan(-\sqrt{\frac{b_n}{a_n}})$. Eq. (B.3) is then transformed to:

$$P(t) = \frac{1}{2}a_0 \pm \sum_{n=1}^{\infty} \sqrt{a_n^2 + b_n^2} cos \left[nt + arctan(-\sqrt{\frac{b_n}{a_n}}) \right] \tag{B.4}$$

Eq. (B.4) shows that any periodic functions can be approximated by the linear combination of cosine functions. By substituting Eq. (B.4) into Eq. (B.1), we see that the diffusion process is estimated to be the integral of Green's function times the combination of cosine functions. Applying the PSD summation identity, we can draw the conclusion that the solution to Eq. (B.1) is also PSD. Therefore, it is a valid kernel function. □

APPENDIX C

LEMMA 1 AND PROOF

C.1 Lemma 1 and Proof

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

Proof. In differential geometry, a curvature flow numerically links intrinsic geometric features and extrinsic flows together (Kichenassamy et al. 1995). We take the proof on a planar curve by assuming manifold $\mathcal{M}$ is a two dimensional manifold in $\mathbb{R}^3$ as an example for convenience. In this case, the GAC kernel is actually equivalent to a curve-shortening flow which can be considered as a one dimension mean curvature flow (**altschuler1993shortening**). Figure 20 shows sketch plots of the symbols used in this proof. Suppose v is a point on the manifold. $C(v)$ is the intersection between the manifold and the normal plane on v. As known, $C(v)$ is a 1-dimensional smooth curve. Assume one point moves along C from v to v'. Let Δs be the arc length of this movement and θ be the rotation angle of the tangent vector, then we can define the following concepts:
(i) the velocity vector at v is $\frac{dC}{dv}$;
(ii) the velocity is the magnitude of the velocity vector, which is $\left|\frac{dC}{dv}\right| = \frac{ds}{dv}$;
(iii) the unit tangent vector $T = \frac{dC}{ds} / \left|\frac{dC}{ds}\right|$ and the unit normal vector $N = \mathcal{R}T$. $\mathcal{R}$ is a $\pi/2$ rotation matrix;
(iv) the curvature κ, which measures how fast the unit tangent vector rotates relative to the arc length: $\kappa = \lim\limits_{\Delta s \to 0} \left|\frac{\Delta\theta}{\Delta s}\right|$. And we can further get $\frac{dT}{ds} = (-sin\theta, cos\theta)\frac{d\theta}{ds} = \kappa N$ and similarly $\frac{dN}{ds} = -\kappa T$.

Assume all points on the curve start to move along their normal directions at a velocity of $\kappa(v)$ during time t, we have the curvature flow: $\frac{dC}{dt} = \kappa N$. With the equation (iii) and (iv), we write the curvature flow as: $\frac{dC}{dt} = \frac{d}{ds}\frac{dC}{ds}$, which is clearly

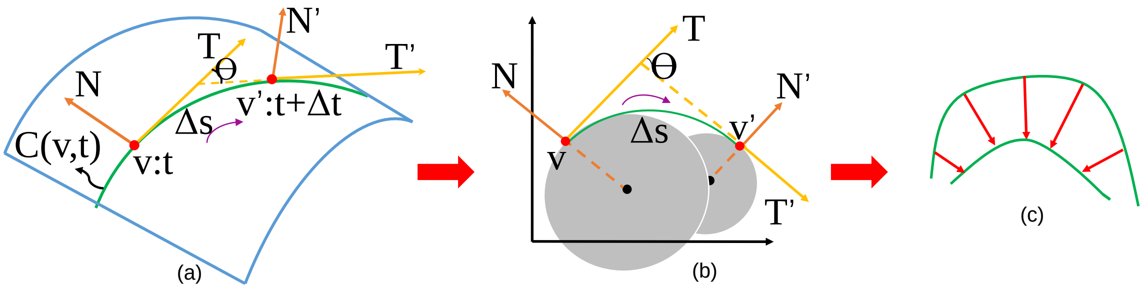


Figure 20. Sketch plots of (a) a planar curve example; (b) the curve in a 2D plane. The shaded circles are inscribed circles on v and v'; (c) the curve moves from top to down by curvature flow.

a diffusion process with a zero reaction function. This equivalence proves that the GAC kernel can theoretically reflect the geometric features. From the perspective of physical meanings, $G(v, v')$ means how much curvature changes from point v to its neighborhood v' during a period of time. Therefore, the physical meanings also support that the GAC kernel embeds the geometric information of the manifolds. Lemma 1 is proved. □

APPENDIX D

LEMMA 2 AND PROOF

D.1 Lemma 2 and Proof

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

Proof. According to the reaction diffusion theory (**kuttler2011reaction**; **strauss2013partielle**), Eq. (B.1) can be expressed as:

$$T = \int_0^t \int_{-\infty}^{\infty} G(v - v', t - s) P(v', s) dv' ds \tag{D.1}$$

If integrating along the temporal variable, then the result has the form $\int P(v')G(v, v')dv'$, which matches with the definition of a convolutional filtering $\int f(v')h_t(v, v')dv' = (f_0 * h_t)(v)$. Our kernel derivation also indicates the existence of a convolution on manifolds. Reminding that we estimate the integral in Eq. (B.1) as the summation of a sine Fourier transform and a cosine Fourier transform (Eq.14 in the main submission). Each term implements the transform from time domain to frequency domain. According to the Convolution Theorem, we can draw the same conclusion. The similar theory has also been applied in geometric deep learning to realize the convolution on manifolds (Bronstein et al. 2017). Lemma 2 is proved. □