Gaussian Processes on Cellular Complexes

Mathieu Alain [*1] **So Takao** [*1 2] **Brooks Paige** [1] **Marc Peter Deisenroth** [1]

Abstract

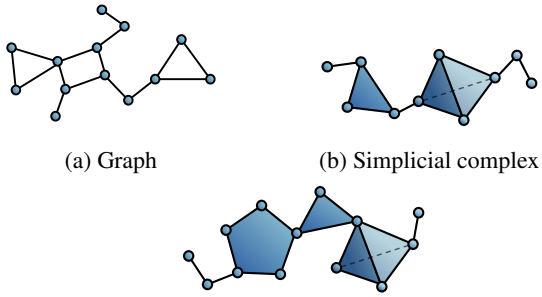
In recent years, there has been considerable interest in developing machine learning models on graphs to account for topological inductive biases. In particular, recent attention has been given to Gaussian processes on such structures since they can additionally account for uncertainty. However, graphs are limited to modelling relations between two vertices. In this paper, we go beyond this dyadic setting and consider polyadic relations that include interactions between vertices, edges and one of their generalisations, known as cells. Specifically, we propose Gaussian processes on cellular complexes, a generalisation of graphs that captures interactions between these higher-order cells. One of our key contributions is the derivation of two novel kernels, one that generalises the graph Matérn kernel and one that additionally mixes information of different cell types.

1. Introduction

The abundance of graph-structured problems in science and engineering stimulates the development of machine learning (ML) models, such as graph neural networks (GNNs) (Scarselli et al., 2008) and graph kernel machines (Smola & Kondor, 2003). The former has achieved great success in a broad range of tasks, from molecular docking (Corso et al., 2023) to text summarisation (Fernandes et al., 2019). However, GNNs do not provide predictive uncertainty, which is an essential feature in decision-making applications. Recent work on Gaussian processes (GPs) defined on graphs (Borovitskiy et al., 2020; Nikitin et al., 2022; Opolka et al., 2022; Zhi et al., 2023) takes the latter approach, which naturally accounts for uncertainty quantification, but may lack the expressibility of GNNs.

[*]Equal contribution [1]Centre for Artificial Intelligence, University College London, London, UK. [2]Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA. Correspondence to: Mathieu Alain <mathieu.alain.21@ucl.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).



(a) Graph (b) Simplicial complex

(c) Cellular complex

Figure 1. Graph, simplicial complex, and cellular complex (specifically: polyhedral complex). A simplicial complex cannot represent arbitrary polygons like the pentagon in (c).

Although graphs are an invaluable data structure, one of their main limitations is that they cannot represent interactions beyond the dyadic setting (i.e., between two vertices). However, these interactions do exist and have important applications, such as group interactions in social networks (Alvarez-Rodriguez et al., 2021), neuronal dynamics in cortexes (Yu et al., 2011), and trigenic interactions in gene networks (Kuzmin et al., 2018). Cellular complexes are a generalisation of graphs that have the ability to model such 'polyadic' interactions (Hatcher, 2001) (see Figure 1). For this reason, they are gradually being used in ML (Hajij et al., 2020; Bodnar et al., 2021) and signal processing (Barbarossa & Sardellitti, 2020; Roddenberry et al., 2022). Although these models have helped to expand the variety of problems one can tackle with ML, as with GNNs, they typically do not quantify uncertainty.

In this paper, we fill this gap by proposing GPs defined on cellular complexes, which enables predictions on different types of cells, such as vertices *and* edges, while equipping them with uncertainty that is consistent with prior assumptions about the model and data. In particular, we propose the cellular Matérn kernel, a generalisation of the graph Matérn kernel (Borovitskiy et al., 2020), which enables modelling of signals on arbitrary cell types. By fixing an orientation on each cell (i.e., a reference 'direction'), this produces predictions on cells that are *directed*, allowing us to consider signals that have an associated direction.

Furthermore, prompted by settings with strong correlations

between data supported on different cell types, we also develop a new kernel, the reaction-diffusion kernel, which leverages the Dirac operator (Bianconi, 2021; Calmon et al., 2023) to mix information between different types of cells. This enables us to model various types of signal jointly so that inference on one type of cell can help the other, and vice versa. Interestingly, the Dirac operator has also been used in higher-order neural networks (Battiloro et al., 2023).

Our main contribution is to define GPs on cellular complexes, which allows us to extend the modelling capability of graph GPs in the following novel ways:

- Model quantities supported on the edges, and on higher-order 'cells' such as volumes.
- Handles orientation on cells naturally. In particular, we can make directed predictions.
- Allows for joint modelling of data supported on vertices, edges and higher-order cells.

2. Gaussian Processes

A *Gaussian process* (GP) on a set X is a random function $f : X \to \mathbb{R}$, such that for any finite collection of points $x_1, \ldots, x_N \in X$, the random vector $(f(x_1), \ldots, f(x_N)) \in \mathbb{R}^N$ is jointly Gaussian. GPs are defined by a mean function $\mu : X \to \mathbb{R}$ and a kernel $\kappa : X \times X \to \mathbb{R}$, which satisfy $\mu(x) = \mathbb{E}[f(x)]$ and $\kappa(x, x') = \mathrm{Cov}[f(x), f(x')]$, respectively, for any $x, x' \in X$.

Let $(\boldsymbol{x}, \boldsymbol{y}) := \{(x_i, y_i)\}_{i=1}^N$ be training data, where $y_i = f(x_i) + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. We can make inference about $\boldsymbol{f}_* := f(\boldsymbol{x}_*)$ at arbitrary test points $\boldsymbol{x}_*$ by computing the posterior predictive mean and covariance

$$\boldsymbol{\mu}_{\mathbf{f}_*|\mathbf{y}} = \boldsymbol{\mu}_{\mathbf{f}_*} + \mathbf{K}_{\mathbf{f}_*\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_{\mathbf{f}}) \quad (1)$$

$$\boldsymbol{\Sigma}_{\mathbf{f}_*|\mathbf{y}} = \mathbf{K}_{\mathbf{f}_*\mathbf{f}_*} - \mathbf{K}_{\mathbf{f}_*\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{ff}_*}, \quad (2)$$

respectively, where $\boldsymbol{\mu}_{\mathbf{f}} = \mu(\boldsymbol{x}), \boldsymbol{\mu}_{\mathbf{f}_*} = \mu(\boldsymbol{x}_*)$ denotes the mean and $\mathbf{K}_{\mathbf{ff}} = \kappa(\boldsymbol{x}, \boldsymbol{x})$, $\mathbf{K}_{\mathbf{f}_*\mathbf{f}_*} = \kappa(\boldsymbol{x}_*, \boldsymbol{x}_*)$, $\mathbf{K}_{\mathbf{ff}_*} = \kappa(\boldsymbol{x}, \boldsymbol{x}_*)$ denotes the covariance and cross-covariance of f at $\boldsymbol{x}$ and $\boldsymbol{x}_*$.

2.1. Gaussian processes on graphs

While most existing GPs are defined on continuous domains, in this work, we are interested in graphs $\mathcal{G} = (V, E)$, where we take the input set X to be the set of vertices V, and the edges E to model the 'proximity' between two vertices. As a mathematical object, a GP in this sense is identical to a multivariate Gaussian $\boldsymbol{f} \in \mathbb{R}^{|V|}$, whose indices are the vertices of the graph. However, the extra information provided by the edges allows one to impose a more rigid correlation structure, whereby two vertices are more strongly correlated if they are 'closer' in the graph. Thus, this enables $\boldsymbol{f}$ to take on the characteristics of a continuous GP while being discrete.

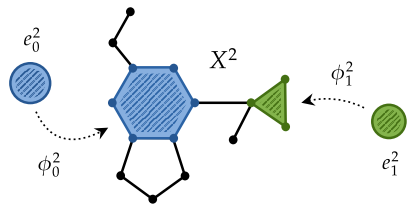


Figure 2. A cellular complex is constructed by attaching boundaries of k-cells e_α^k to the $(k-1)$-skeleton X^{k-1} via a continuous map ϕ_α^k.

A common way of encoding graph structures into multivariate Gaussians is by imposing specific sparsity patterns in the precision matrix, as seen in Gaussian Markov random fields (Rue & Held, 2005). For example, the Matérn GP on graphs (Borovitskiy et al., 2021) is formally defined as $\boldsymbol{f} \sim \mathcal{N}(\mathbf{0}, (\frac{2\nu}{\ell^2} + \boldsymbol{\Delta})^{-\nu})$, where $\boldsymbol{\Delta}$ is the graph Laplacian matrix, a discrete analogue of the standard Laplacian operator on $\mathbb{R}^n$. If we observe the sparsity pattern of the corresponding precision matrix $(\frac{2\nu}{\ell^2} + \boldsymbol{\Delta})^\nu$ for $\nu \in \mathbb{N}$, we see that each vertex x is connected to vertices that are within a radius ν of x in the graph, controlling the 'smoothness' of the process. More generally, we can impose a graph structure by penalising frequencies in the spectral domain of the kernel (Smola & Kondor, 2003). However, graphs are limited in the data they can support. If the data contains polyadic interactions, then a more general structure than graphs is needed.

3. Modelling with Cellular Complexes

Cellular complexes generalise graphs by incorporating higher-order interactions via 'cells', extending the dyadic relation modelled by edges on a graph. Concretely, a *k-cell* is a topological space that is homeomorphic to the unit disk $D^k := \{x \in \mathbb{R}^k : \|x\| < 1\}$. We will employ the standard notation ∂ to refers to the boundary of a topological space. A finite cellular complex X of dimension $n < \infty$ is constructed iteratively, as follows (Hatcher, 2001):

(Step 0**)** Start with a collection $X^0 = \{e_\alpha^0\}_{\alpha=1}^{N_0}$ of 0-cells (i.e., points), called the 0-skeleton.

(Steps $k = 1, \ldots, n$**)** Take a collection $\{e_\alpha^k\}_{\alpha=1}^{N_k}$ of k-cells and glue their boundaries to points in X^{k-1} via a continuous *attaching map* $\phi_\alpha^k : \partial e_\alpha^k \to X^{k-1}$. The resulting space X^k is the k-skeleton (see Figure 2).

(Step $n+1$**)** Define the cellular complex to be the topological space $X := \cup_{k=0}^n X^k$.

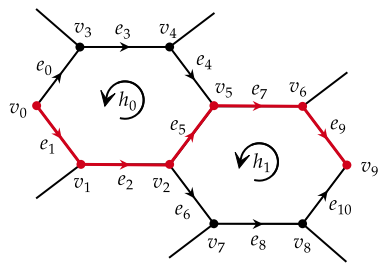


Figure 3. The path going from vertex v_0 to v_9 (in red) can be expressed as a 1-chain $c = e_1 + e_2 + e_5 + e_7 + e_9$, while $-c$ represents the reverse path going from v_9 to v_0. Boundaries of cells can be expressed as chains, whose direction is consistent with the cell orientation. For example, $\partial h_0 = e_1 + e_2 + e_5 - e_4 - e_3 - e_0$.

In the special case where the attaching maps $\phi_\alpha^k : \partial e_\alpha^k \to X^{k-1}$ are embeddings, we say that the cellular complex is *regular*. This omits pathological cases, such as when boundaries of a k-cell collapse to a point or fold into itself. Hereafter, when we refer to cellular complexes, we will always assume that they are regular and finite. Further, when we fix an orientation on each cell (viewed as a topological manifold), we say that X is *oriented*, which we will also assume hereafter.

Example 1. *A directed/undirected graph (see Figure 1a) is a one-dimensional oriented/non-oriented cellular complex, where the vertices are the* 0*-cells, the edges are the* 1*-cells, and the attaching maps associate the two endpoints of an edge to a pair of vertices.*

Example 2. *Another important class are the simplicial complexes (see Figure 1b), where the k-cells are taken to be the k-simplices and the attaching maps glue the boundaries of two k-simplices homeomorphically. Simplices are cells that contain all their sub-cells.*

3.1. Chains and Cochains

Chains and cochains are key concepts on cellular complexes that formalise the notion of paths and functions over k-cells, respectively. Given an n-dimensional cellular complex X, we define a *k-chain* c_k for $0 \leq k \leq n$ as a formal sum of k-cells

$$c_k = \sum_{\alpha=1}^{N_k} n_\alpha e_\alpha^k, \quad n_\alpha \in \mathbb{Z}. \tag{3}$$

Intuitively, this generalises the notion of directed paths on a graph, as we show in Figure 3. We denote the set of all k-chains on X by $C_k(X)$, which has the algebraic structure of a free Abelian group[1] with basis $\{e_\alpha^k\}_{\alpha=1}^{N_k}$.

[1] One may interpret this as a vector space with coefficients restricted to the integers.

For any cellular complex X, there is a canonical operation defined on chains called the *boundary operator* $\partial_k : C_k(X) \to C_{k-1}(X)$, associating the boundary of a k-chain to a $(k-1)$-chain. This is defined as a linear map[2]

$$\partial_k \left(\sum_{\alpha=1}^{N_k} n_a e_\alpha^k \right) = \sum_{\alpha=1}^{N_k} n_a \partial e_\alpha^k, \tag{4}$$

where ∂e_α^k is the boundary of the k-cell e_α^k, expressed as a $(k-1)$-chain. The cycle direction of ∂e_α^k must agree with the orientation of e_α^k, as illustrated in Figure 3 (see Appendix A for more details).

Parallel to chains, there exist dual objects known as *cochains* on a cellular complex X, defined as follows.

Definition 3. *A k-cochain on X is a linear map*[2] $f : C_k(X) \to \mathbb{R}$ *assigning real numbers to k-chains, i.e.,*

$$f\Big(\sum_{\alpha=1}^{N_k} n_\alpha e_\alpha^k \Big) = \sum_{\alpha=1}^{N_k} n_\alpha f(e_\alpha^k), \tag{5}$$

where $f(e_\alpha^k) \in \mathbb{R}$ is the value of f at cell e_α^k.

The space of k-cochains, denoted $C^k(X)$ (with a superscript), forms a real vector space with dual basis $\{(e_\alpha^k)^*\}_{\alpha=1}^{N_k}$, satisfying $(e_\alpha^k)^* e_\beta^k = \delta_{\alpha\beta}$. The boundary operators on chains naturally induce analogous operators on the space of cochains, referred to as the *coboundary operators*, defined as follows.

Definition 4. *For $0 \leq k < n$, the coboundary operator $d_k : C^k(X) \to C^{k+1}(X)$ is defined via the relation*

$$d_k f(c) = f(\partial_{k+1} c) \tag{6}$$

for all $f \in C^k(X)$ and $c \in C_{k+1}(X)$. To simplify our presentation, we also define $d_k f \equiv 0$ for $k \in \{-1, n\}$.

3.2. Hodge Laplacian and Dirac Operator

We introduce a generalisation of the graph Laplacian to cellular complexes, which will be used in our construction of kernels. For each k, let $\{w_\alpha^k\}_{\alpha=1}^{N_k}$ be a set of real, positive weights. Then, for any $f, g \in C^k(X)$, we define the weighted L^2-inner product as

$$\langle f, g \rangle_{L^2(\boldsymbol{w}^k)} := \sum_{\alpha=1}^{N_k} w_\alpha^k \, f(e_\alpha^k) \, g(e_\alpha^k). \tag{7}$$

The inner product induces an adjoint of the coboundary operator, denoted $d_k^* : C^{k+1}(X) \to C^k(X)$, i.e.,

$$\langle d_k^* f, g \rangle_{L^2(\boldsymbol{w}^k)} = \langle f, d_k g \rangle_{L^2(\boldsymbol{w}^{k+1})} \tag{8}$$

for any $f \in C^{k+1}(X)$ and $g \in C^k(X)$. This leads to the following definition.

[2] A group homomorphism to be more precise.

Object	Representation
Chain $c = \sum_{\alpha=1}^{N_k} n_\alpha e_\alpha^k$	$\boldsymbol{c} = (n_1, \dots, n_{N_k})^\top$
Cochain $f = \sum_{\alpha=1}^{N_k} f_\alpha (e_\alpha^k)^*$	$\boldsymbol{f} = (f_1, \dots, f_{N_k})^\top$
Boundary operator ∂_k	$\mathbf{B}_k \in \mathbb{Z}^{N_{k-1} \times N_k}$
Coboundary operator d_k	$\mathbf{B}_{k+1}^\top \in \mathbb{Z}^{N_{k+1} \times N_k}$
Inner-product $\langle f, g \rangle_{L^2(\boldsymbol{w}^k)}$	$\boldsymbol{f}^\top \mathbf{W}_k \boldsymbol{g}$
Hodge Laplacian Δ_k	$\boldsymbol{\Delta}_k := \mathbf{B}_k^\top \mathbf{W}_{k-1}^{-1} \mathbf{B}_k \mathbf{W}_k + \mathbf{W}_k^{-1} \mathbf{B}_{k+1} \mathbf{W}_{k+1} \mathbf{B}_{k+1}^\top$

Table 1. Numerical representation of key objects and operations defined on cellular complexes. Here, $\mathbf{W}_k = \texttt{diag}(w_1^k, \dots, w_{N_k}^k)$ is the matrix of cell-weights and $\mathbf{B}_k$ is the order-k incidence matrix, whose j-th column corresponds to the vector representation of the cell boundary ∂e_j^k, viewed as a $(k-1)$-chain.

Definition 5. *The Hodge Laplacian $\Delta_k : C^k(X) \to C^k(X)$ on the space of k-cochains is defined as*

$$\Delta_k := d_{k-1} \circ d_{k-1}^* + d_k^* \circ d_k. \tag{9}$$

We defer to Section 4 to see that this generalises the graph Laplacian by considering its numerical representation.

The *Dirac operator*

$$\delta_k := d_{k-1}^* \oplus d_k : C^k(X) \to C^{k-1} \oplus C^{k+1}(X) \tag{10}$$

maps a k-cochain onto a direct sum of $(k-1)$ and $(k+1)$-cochains. One can verify that the relation $\Delta_k = \delta_k^* \delta_k$ holds. Hence, the Dirac operator is understood as a formal 'square root' of the Hodge Laplacian.

3.3. Numerical Representation

In Table 1, we display how the objects considered above can be represented as matrices and vectors in order to make explicit computations with them. We refer to Appendix B for the full derivation. In the case $k = 0$ and taking $\mathbf{W}_0 = \mathbf{I}$ (i.e., no vertex weights), we see that the expression for the Hodge Laplacian reduces to the expression for the weighted graph Laplacian $\boldsymbol{\Delta}_0 = \mathbf{B}_1 \mathbf{W}_1 \mathbf{B}_1^\top$. Furthermore, taking $\mathbf{W}_1 = \mathbf{I}$ (no edge weights), we obtain the expression for the standard graph Laplacian $\boldsymbol{\Delta}_0 = \mathbf{B}_1 \mathbf{B}_1^\top$.

4. Gaussian Processes on Cellular Complexes

In this section, we establish our notion of Gaussian processes on cellular complexes, which we take to be a direct sum of Gaussian random cochains. Given a sample space Ω, an event space $\mathcal{F}$ and a probability measure $\mathbb{P}$, we denote by $(\Omega, \mathcal{F}, \mathbb{P})$ the underlying probability triple for which our random variables will be defined over.

4.1. Gaussian Random Cochains

Definition 6. *A random variable $f : \Omega \to C^k(X)$ is called a Gaussian random cochain if for any k-chain $c \in C_k(X)$, the random variable $f(c) : \Omega \to \mathbb{R}$ is Gaussian.*

To facilitate computations, we characterise them using a mean function and a kernel. Defining a mean is straightforward—this is just a fixed cochain. For the kernel, we consider the following definition.

Definition 7. *A kernel on $C_k(X)$ is a symmetric bilinear form*[3] *$\kappa : C_k(X) \times C_k(X) \to \mathbb{R}$ such that for any $c_1, \dots, c_m \in C_k(X)$, we have*

$$\sum_{i,j=1}^{m} \kappa(c_i, c_j) \geq 0. \tag{11}$$

This is an appropriate notion of the kernel:

Theorem 8. *A Gaussian random cochain $f : \Omega \to C^k(X)$ is fully characterised by a mean $\mu \in C^k(X)$ and a kernel $\kappa : C_k(X) \times C_k(X) \to \mathbb{R}$.*

Proof: Appendix C.1.

The vector representation $\boldsymbol{f}$ of f is simply a multivariate Gaussian $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$, whose covariance

$$\mathbf{K} := \begin{pmatrix} \kappa(e_1^k, e_1^k) & \cdots & \kappa(e_1^k, e_{N_k}^k) \\ \vdots & \ddots & \vdots \\ \kappa(e_{N_k}^k, e_1^k) & \cdots & \kappa(e_{N_k}^k, e_{N_k}^k) \end{pmatrix} \tag{12}$$

is the matrix representation of the kernel. Due to the bilinearity of κ, for any $c, d \in C_k(X)$, and their vector representations $\boldsymbol{c}, \boldsymbol{d} \in \mathbb{Z}^{N_k}$, we can write

$$\kappa(c, d) = \boldsymbol{c}^\top \mathbf{K} \boldsymbol{d}. \tag{13}$$

By fixing orientations on each k-cell, we also have a notion of *direction* for the signal f – for a k-cell e_α^k, depending on whether the sign of $f(e_\alpha^k)$ is positive or negative, the direction of f at e_α^k is aligned to, or runs counter to the orientation of e_α^k, respectively.

4.2. GPs on Cellular Complexes

Next, we extend our notion of Gaussian random cochains to direct sums of cochains of different orders. We take this as our definition of Gaussian processes on cellular complexes.

Definition 9. *Let X be an n-dimensional cellular complex. We define a Gaussian process on X as a random variable $f : \Omega \to \bigoplus_{k=0}^n C^k(X)$ such that for any $c = (c_0, \dots, c_n) \in \bigoplus_{k=0}^n C_k(X)$, the random variable $f(c) : \Omega \to \mathbb{R}$ is univariate Gaussian.*

[3] A group bi-homomorphism to be more precise.

As before, we have an appropriate notion of a kernel on this space as a symmetric bilinear form[3]

$$\kappa : \bigoplus_{k=0}^{n} C_k(X) \times \bigoplus_{k=0}^{n} C_k(X) \to \mathbb{R} \tag{14}$$

satisfying $\sum_{i,j} \kappa(c_i, c_j) \geq 0$ for $c_i, c_j \in \bigoplus_{k=0}^{n} C_k(X)$. We also have the following result characterising GPs on cellular complexes via a mean and a kernel.

Theorem 10. *A GP on a cellular complex X is fully characterised by a mean $\mu \in \bigoplus_{k=0}^{n} C^k(X)$ and a kernel $\kappa : \bigoplus_{k=0}^{n} C_k(X) \times \bigoplus_{k=0}^{n} C_k(X) \to \mathbb{R}$.*

Proof: Appendix C.2.

Again, we can represent (14) by a matrix

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{11} & \cdots & \mathbf{K}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{n1} & \cdots & \mathbf{K}_{nn} \end{pmatrix} \tag{15}$$

with $[\mathbf{K}_{nm}]_{ij} = \kappa(e_i^n, e_j^m)$, which defines the covariance of $\boldsymbol{f} \in \mathbb{R}^{N_1 + \cdots + N_n}$, the vector representation of f.

Remark 11. *We emphasise that our notion of a GP on a cellular complex X is not defined as a random function $X \to \mathbb{R}$ (i.e. a GP on the topological space X), but rather as a direct sum of Gaussian random cochains, i.e., a random function $\bigoplus_{k=0}^{n} C_k(X) \to \mathbb{R}$. This will allow us to define covariance structures between cells themselves, instead of between points on cells.*

4.3. Kernels on Cellular Complexes

We provide some concrete examples of kernels defining GPs on cellular complexes, which encompass existing kernels in the literature. For simplicity, we take unit cell-weights $w_\alpha^k = 1$ here and defer the treatment of the general case to Appendix D.3. For any chain $c \in C_k(X)$, we also denote by $c^\flat \in C^k(X)$ the cochain defined by $f(c) = \langle f, c^\flat \rangle_{L^2(\boldsymbol{w}^k)}$ for any $f \in C^k(X)$.

4.3.1. Matérn Kernel

We first consider a generalisation of the Matérn kernel on cellular complexes. Following (Borovitskiy et al., 2020; 2021), consider the stochastic system

$$\left(\frac{2\nu}{\ell^2} + \Delta_k\right)^{\nu/2} f = \mathcal{W}, \tag{16}$$

where $f \in C^k(X)$, Δ_k is the Hodge Laplacian (Definition 5), and $\mathcal{W} : \Omega \to C^k(X)$ is a Gaussian random cochain satisfying $\mathbb{E}[\mathcal{W}(c_0)] = 0$ and $\mathbb{E}[\mathcal{W}(c_1)\mathcal{W}(c_2)] = \langle c_1^\flat, c_2^\flat \rangle_{L^2}$ for any $c_0, c_1, c_2 \in C_k(X)$. The operator $\left(\frac{2\nu}{\ell^2} + \Delta_k\right)^{\nu/2}$ is defined rigorously in Appendix D.1.

The Matérn kernel $\kappa : C_k(X) \times C_k(X) \to \mathbb{R}$ is then defined as a solution to the system

$$\left(\frac{2\nu}{\ell^2} + \Delta_k\right)^{\nu} \kappa(c, \cdot) = c^\flat, \quad \forall c \in C_k(X), \tag{17}$$

which is a kernel in the sense that it satisfies the following property.

Proposition 12. *The solution to (17) is related to the solution f of the system (16) by*

$$\kappa(c, c') = \mathbb{E}[f(c)f(c')], \quad \forall c, c' \in C_k(X). \tag{18}$$

Thus, κ solving (17) is the kernel of the Gaussian random cochain f.

Proof: Appendix D.1.1.

We can express the kernel in (17) by a matrix

$$\mathbf{K} = \mathbf{U}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \boldsymbol{\Lambda}^2\right)^{-\nu}\mathbf{U}^\top, \tag{19}$$

where $\boldsymbol{\Delta}_k = \mathbf{U}\boldsymbol{\Lambda}^2\mathbf{U}^\top$ is the eigendecomposition of the Hodge Laplacian. In the case $k = 0$, (19) recovers exactly the graph Matérn kernel by (Borovitskiy et al., 2021). We may also extend this construction to direct sums of cochains by replacing the Hodge Laplacian in (16) and (17) by the *super Laplacian* $\mathcal{L} := \bigoplus_{k=0}^{n} \Delta_k$, resulting in kernel (19), where $\mathbf{U}\boldsymbol{\Lambda}^2\mathbf{U}^\top$ is now the eigendecomposition of the super Laplacian matrix $\boldsymbol{\mathcal{L}} = \texttt{blockdiag}(\boldsymbol{\Delta}_0, \ldots, \boldsymbol{\Delta}_n)$. While this defines a GP on direct sums of cochains, due to the block-diagonal structure of the super-Laplacian, no mixing occurs between different cochains.

In the special case of GPs defined over the edges of a graph (i.e., Gaussian 1-cochains), a similar construction of the Matérn kernel has been explored in the concurrent work Yang et al. (2024), where they also employ the Hodge decomposition to add further flexibility of the model. Another related work (Pinder et al., 2021) explores the construction of Matérn GPs on hypergraphs to model higher-order interactions. However, the work only considers inference on the vertices, whereas our method focuses on making inferences on signals supported on the interactions themselves (i.e., the cells).

4.3.2. Reaction-diffusion Kernel

Next, we introduce a new type of kernel, which we term the *reaction-diffusion kernel*, that enables mixing of information between cochains of different orders. We will operate here entirely with the vector representation of cochains for ease of presentation.

Consider the *Dirac matrix*

$$\boldsymbol{\mathcal{D}} = \begin{pmatrix} \mathbf{0} & \mathbf{B}_1 & \cdots & \mathbf{0} \\ \mathbf{B}_1^\top & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{B}_n \\ \mathbf{0} & \dots & \mathbf{B}_n^\top & \mathbf{0} \end{pmatrix}, \tag{20}$$

whose k-th column is a numerical representation of the k-th Dirac operator (10). We can check that $\boldsymbol{\mathcal{D}}^2 = \boldsymbol{\mathcal{L}}$ holds. Thus, the Dirac matrix $\boldsymbol{\mathcal{D}}$ and the super-Laplacian matrix $\boldsymbol{\mathcal{L}}$ share a common eigenbasis $\mathbf{U}$ with eigenvalues $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}^2$, respectively. Now consider the stochastic system

$$(r\mathbf{I} - c\boldsymbol{\mathcal{D}} + d\boldsymbol{\mathcal{L}})^{\frac{\nu}{2}} \boldsymbol{f} = \boldsymbol{w}, \quad \boldsymbol{w} \sim \mathcal{N}(0, \mathbf{I}) \tag{21}$$

for some constants $r, c, d, \nu \in \mathbb{R}_{\geq 0}$. Then, the corresponding kernel matrix is given by

$$\mathbf{K} = \mathbf{U}(r\mathbf{I} - c\boldsymbol{\Lambda} + d\boldsymbol{\Lambda}^2)^{-\nu}\mathbf{U}^\top, \tag{22}$$

which we term the *reaction-diffusion kernel*.[4] This is a kernel in the sense that it satisfies the following result.

Proposition 13. *The kernel defined by* (22) *is related to the solution* $\boldsymbol{f}$ *of the system* (21) *by*

$$[\mathbf{K}]_{ij} = \mathbb{E}[f_i f_j], \quad \forall i, j. \tag{23}$$

Proof: Appendix D.2.1.

Remark 14. *Our naming of the reaction-diffusion kernel derives from the similarity of system* (21) *with the multi-component reaction-diffusion equation*

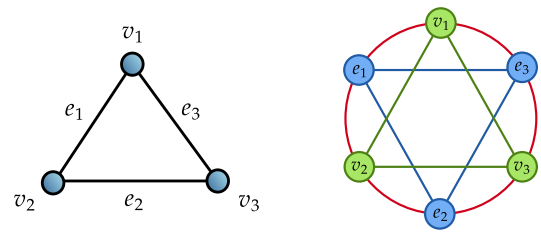
$$\frac{\partial \boldsymbol{f}}{\partial t} = (r - c\boldsymbol{\mathcal{D}} + d\boldsymbol{\mathcal{L}})\boldsymbol{f}, \tag{24}$$

where the first and third term model the reaction and diffusion of a quantity respectively, and the second term models the cross-diffusion of multiple quantities.

To interpret this kernel, we look at the corresponding precision matrix $\mathbf{P} = \mathbf{K}^{-1}$, which encodes the probabilistic graphical model (PGM) representation of the model $\boldsymbol{f}$, wherein variables f_i and f_j are linked if and only if $[\mathbf{P}]_{ij} \neq 0$ (Rue & Held, 2005). For simplicity, taking $\nu = 1$ and restricting to a 1-dimensional cellular complex, the precision takes the expression

$$\mathbf{P} = \begin{pmatrix} r\mathbf{I} + d\boldsymbol{\Delta}_0 & -c\mathbf{B}_1 \\ -c\mathbf{B}_1^\top & r\mathbf{I} + d\boldsymbol{\Delta}_1 \end{pmatrix}. \tag{25}$$

[4] In general, $\mathbf{K}$ is indefinite. To fix this, we set ν to be an even integer in (22), which will make it positive definite for all $r, c, d \in \mathbb{R}_{\geq 0}$, excluding the set $\mathbb{X} := \bigcup_i \{(r, c, d) : r \pm c\lambda_i + d\lambda_i^2 = 0\}$. This set has Lebesgue measure zero. If $(r, c, d) \in \mathbb{X}$, $\mathbf{K}$ becomes positive *semi-definite*. Therefore it defines a degenerate Gaussian measure. Inference using degenerate Gaussian measures is, however, still valid, provided $\sigma > 0$ is strictly positive in (1)–(2).



(a) 1D cellular complex (b) PGM representation

Figure 4. Probabilistic graphical structure of the reaction-diffusion GP. Interactions between vertices (green) and between edges (blue) are shown as well as the mixing between cochains of different orders (red). The cellular Matérn kernel does not have this mixing property.

For $\boldsymbol{f} = (\boldsymbol{f}_0, \boldsymbol{f}_1)^\top$ in (21), the graphical structure of the k-th component $\boldsymbol{f}_k$ is summarised by the matrix $r\mathbf{I} + d\boldsymbol{\Delta}_k$ for $k \in \{0, 1\}$, and the dependence between $\boldsymbol{f}_0$ and $\boldsymbol{f}_1$ is represented by the incidence matrix $\boldsymbol{B}_1$ (dotted red in 4b). Since no communication between $\boldsymbol{f}_0$ and $\boldsymbol{f}_1$ occurs when $c = 0$, the Dirac term is essential for allowing information to propagate between cochains of different orders.

Let us now consider two special cases of the kernel (22). In the first case, taking $r = 2\nu/\ell^2$, $c = 0$, $d = 1$ and $\nu = \nu$, we see that (22) recovers the Matérn kernel (19). Since $c = 0$, there is no flow of information between cochains of different orders, resulting in independence between the random cochains $\boldsymbol{f}_0, \dots, \boldsymbol{f}_n$.

For $r = m^2$, $c = 1$, $d = 0$ and $\nu = 2$, we obtain

$$\mathbf{K} = \mathbf{U}(m\mathbf{I} - \boldsymbol{\Lambda})^{-2}\mathbf{U}^\top = (m\mathbf{I} - \boldsymbol{\mathcal{D}})^{-2}. \tag{26}$$

This kernel is considered by (Calmon et al., 2023) (in the form of a regulariser) for retrieving mixed topological signals supported on the k-cells for $k \leq 2$.

5. Results

In this section, we demonstrate the results of our GP model defined over cellular complexes (hereafter referred to as CC-GP) on two examples. First, we demonstrate that CC-GPs can make directed predictions on the edges of a graph by considering the problem of ocean current interpolation. In the second example, we investigate the effect of inter-signal mixing in the reaction-diffusion kernel. We provide details of the experimental setups in Appendix E.

5.1. Directed Edge Prediction

In the numerical simulation of fluids and especially in finite element methods (FEMs), it is common to treat vector fields as signals supported on the edges of a mesh (Arnold et al., 2006) to give them the flexibility for dealing with com-

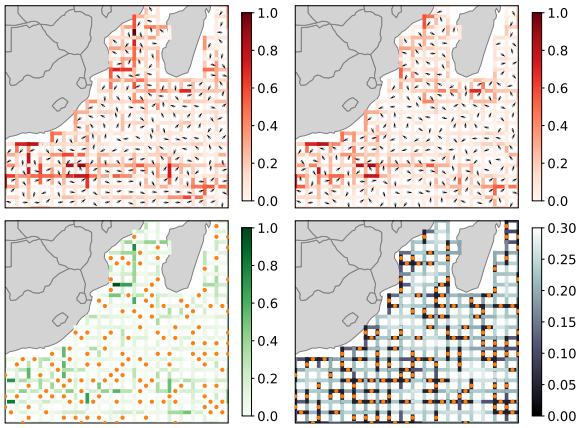


Figure 5. Prediction of geostrophic current around the Southern tip of Africa using the CC-Matérn GP on edges. (Top left) Ground truth. (Top right) Predicted mean. (Bottom left) Absolute error. (Bottom right) Standard deviation. Orange dots are observed edges.

plex geometries, e.g., arising from the coastlines in ocean modelling. As there is an increasing adoption of FEM for weather and climate modelling (for example, the UK Met Office's GungHo model uses FEM with cubical elements (Staniforth et al., 2013)), it is of interest to consider methods that propagate information from observations directly onto the finite element vertices and edges.

Our CC-GP model can naturally be applied in this setting: Consider the geostrophic current data from the NOAA CoastWatch (2023) database. First, we project the geostrophic current around the southern tip of Africa onto the oriented edges of a two-dimensional cubical complex by averaging the flow along each edge (Desbrun et al., 2006). This yields directed signals on the edges representing the vector field. We then train our edge-based Matérn GP (see Section 4.3.1 defined over a 1-cochain) on 30% of the data, selected randomly. The main objective of this experiment is to demonstrate that our approach can capture the directional information of the vector field, which would otherwise be difficult with existing approaches.

Figure 5 shows resulting predictions. For ease of visualisation, we display the magnitude of the predicted signals by colours on the edges; arrows indicate the predicted direction on each cell, computed by averaging the signals on its boundaries and then taking the resulting direction. We see that our CC-Matérn GP on edges captures the general characteristics of the ground-truth vector field with similar magnitudes and directions, indicating that it can correctly diffuse information onto neighbouring edges.

We compare the results with a graph Matérn GP baseline

	MSE (↓)	NLL (↓)
Graph Matérn	$\mathbf{0.030 \pm 0.000}$	-684.54 ± 4.20
Edge Matérn (ours)	$\mathbf{0.029 \pm 0.001}$	$\mathbf{-703.42 \pm 5.10}$

Table 2. Mean square error (MSE) and negative log-likelihood (NLL) of ocean current magnitude predictions using (a) a graph Matérn baseline, and (b) the edge Matérn GP. Mean and standard error are shown.

MSE (↓)	CC-Matérn	Reaction-diffusion
Vertices	0.165 ± 0.005	$\mathbf{0.076 \pm 0.004}$
Edges	0.335 ± 0.014	$\mathbf{0.200 \pm 0.010}$
Triangles	0.272 ± 0.005	$\mathbf{0.166 \pm 0.005}$
NLL (↓)		
Vertices	28.81 ± 1.75	$\mathbf{-9.31 \pm 1.61}$
Edges	136.77 ± 4.32	$\mathbf{71.07 \pm 6.29}$
Triangles	82.84 ± 1.56	$\mathbf{39.78 \pm 2.47}$

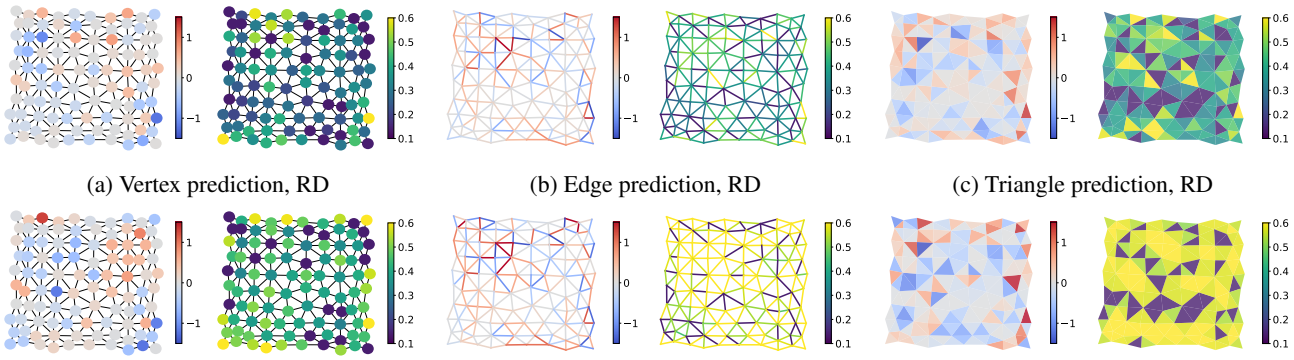
Table 3. Mean square error (MSE) and negative log-likelihood (NLL) of predictions on the synthetic data (mean and standard error across 20 random seeds). Overall, the performance of the reaction-diffusion GP is superior to that of the Matérn GP on the cellular complex, highlighting the benefits of mixing information across different cell types.

defined over the corresponding line graph[5]. Since this baseline cannot infer directions on the edge signals, we use it to only predict its magnitude. The results are shown in Table 2, where we report the mean square error (MSE) and negative log-likelihood (NLL) scores on the magnitude of the ocean current. MSE results for both models are comparable; however, our edge-based Matérn GP performs better than the graph Matérn GP on the NLL. This suggests that in addition to being able to infer the directions on edges, predictive uncertainties are better and the topological inductive bias contained in the edge-based Matérn GP also helps to improve predictions for the magnitudes.

5.2. Signal Mixing

We illustrate the benefits of mixing signals using the reaction-diffusion kernel on synthetic data, where we constructed a 2D-simplicial mesh consisting of 10×10 vertices. We generated artificial signals on the edges by considering a random 1-cochain $f = \sum_{i=k}^{K} \xi_i u_i$, where $\xi_i \sim \mathcal{N}(0, \lambda_i^{-1})$, $\{(\lambda_i, u_i)\}_i$ are the eigenpairs of the Hodge Laplacian Δ_1 and $0 < k < K$ are the minimal and maximal wavenumbers controlling the smoothness of the edge field. We then generated data supported on the triangles and vertices of the mesh by applying the coboundary operator d_1 and its adjoint d_1^* respectively to f (numerically, this corresponds to applying

[5]This is the graph constructed by treating the edges as vertices and connecting them if they share a vertex.



(a) Vertex prediction, RD (b) Edge prediction, RD (c) Triangle prediction, RD

(d) Vertex prediction, CC-Matérn (e) Edge prediction, CC-Matérn (f) Triangle prediction, CC-Matérn

Figure 6. Predictive distributions on vertices, edges, and triangles using the reaction-diffusion kernel (RD, top row) and the CC-Matérn kernel (bottom row). Left panels: differences between mean prediction and ground truth (values close to 0 are good); right panels: corresponding predictive standard deviations. By taking correlation properties into account, the RD kernel produces better predictions on vertices, edges, and triangles.

the matrices $\mathbf{B}_2^\top$ and $\mathbf{B}_1$). We randomly selected a third of the data supported on each type of cell (vertices, edges and triangles) and perturbed them by noise for training. The aim is to recover the signals on the remaining two thirds of the cells.

We compare the results of the Matérn GP on the cellular complex (CC-Matérn) and the reaction-diffusion (RD) GP, used to make predictions across various cell-types. Figure 6 shows an example result with $k = 20$ and $K = 100$, where we display the difference (left panels) from the ground truth and the predictive standard deviation (right panels) for vertex (left column), edge (center column), and triangle predictions (right column) for the RD kernel (top row) and the CC-Matérn kernel (bottom row). Overall, we see how the mixing of information across different cell types in the RD kernel helps to improve predictions, as evidenced by the overall smaller errors in each cell type (with small predictive uncertainty), compared to the predictions made by the CC-Matérn kernel. In particular, we see how data supported on one type of cell can be used to enhance the predictions on another cell type as we can infer from the standard deviation plots. The standard deviation of the CC-Matérn GP is more localised around the data points than that of the reaction-diffusion GP, suggesting that information on different cell types is being mixed in the latter.

This behaviour is also quantitatively reflected in Table 3, which reports the mean squared error (MSE) and the negative log-likelihood (NLL) scores of the predictions for both models, computed across 20 random seeds (mean and standard errors). We see how the results for the RD-GP are on average significantly better than that for the CC-Matérn kernel on both metrics, highlighting the benefits of mixing for both prediction and uncertainty quantification.

5.3. Modelling Electromagnetic Fields

Another potential application area of GPs defined over CCs is to model electromagnetic fields, which have natural representations as cochains. In particular, the scalar potential, electric field and magnetic field generated by point charges on a 2D plane can be modelled geometrically as scalar, vector and pseudovector fields, respectively. Upon discretisation, these can then be represented by $0, 1$ and 2-cochains, respectively, using the continuous-discrete correspondence between tensor fields (more precisely, differential forms) and cochains (Desbrun et al., 2006).

In this experiment, we use simulations of the scalar potential (V), electric field (E) and magnetic field (B) generated by 10 randomly sampled point charges. These are then projected onto $0, 1$ and 2-cochains, respectively, from which we extract noisy observations at a sixth of randomly selected cells in each skeleton of the generated CC. Then, similar to the experiment in Section 5.2, the objective is to recover the signals on the remaining cells. In contrast to the previous experiment, the correlation structure between signals supported on different cell types are more complex, provided indirectly through Maxwell's equations. Our goal is thus to see whether the RD kernel is still useful in this setting where the correlation between fields exist, but are not artificially imposed, as with the previous experiment. The electromagnetic simulation was performed using the Python package `PyCharge` (Filipovich & Hughes, 2022).

In Table 4, we compare the results of the CC-Matérn GP and the reaction-diffusion (RD) GP on this experiment. Generally, we observe that the RD-GP yields slightly better results than the CC-Matérn GP on both the MSE and the NLL, with the exception of the MSE on the scalar potential. For predictions of the electric and magnetic fields, the RD kernel

	Matérn MSE	RD MSE	Matérn NLL	RD NLL
V	**0.102** ± **0.008**	0.11 ± 0.011	45.75 ± 7.20	**45.41** ± **7.35**
E	0.130 ± 0.008	**0.128** ± **0.009**	104.3 ± 14.0	**100.7** ± **15.8**
B	0.190 ± 0.026	**0.178** ± **0.023**	124.7 ± 18.8	**117.3** ± **17.2**

Table 4. Comparison of independent Matérn GPs and the reaction-diffusion (RD) GP for joint modelling of the 2D potential (V), electric (E) and magnetic (B) fields. We compare the mean square error (MSE) and negative log-likelihood (NLL). For the E and B fields, RD ourperforms independent Matérn on 8/10 seeds.

outperformed the Matérn kernel on eight out of ten random seeds that were used to generate the results in Table 4. Our results indicate that interestingly, mixing of information between the different cell types using the RD kernel can still be useful in this setting. However, we also note that the improvements that we see here are much less pronounced than what we observed in the previous experiment, where correlations between signals on different cell types were imposed more directly.

6. Discussion

Our experiments demonstrate the benefits of incorporating the structure of cellular complexes to model data that are naturally supported on higher-order networks, enabling us to treat (a) directed signals and (b) mixed signals with ease, which a standard graph GP cannot handle. This opens up new possibilities for modelling data on non-Euclidean domains. For instance, as a promising direction, we believe that our approach could be useful for modelling vectorial or higher-order tensorial quantities supported on arbitrary manifolds, by relying on the structural parallels between CCs and differential forms, typically employed in numerical simulations of vectorial/tensorial quantities supported on manifolds (Arnold et al., 2006). This idea is explored in our experiments in Sections 5.1 and 5.3 on simple 2D domains. We aim to extend this to more complex domains in future work, possibly incorporating boundary conditions.

Indeed, vectorial GPs on manifolds have been considered before for example in (Hutchinson et al., 2021) and more recently in (Robert-Nicoud et al., 2024). However, the former construction relies on embedding the manifold in a larger ambient space in order to make use of scalar kernels, and the latter relies on the Helmholtz decomposition to make use of scalar kernels. On the other hand, by directly encoding the vector information on the edges of a CC, one can easily model vector fields over arbitrary manifolds, possibly with boundaries (see Figure 7). Adopting a CC perspective thus makes modelling with vectorial GPs easier, and can be further generalised to the tensorial setting naturally. Further, the reaction-diffusion GP provides a topologically consistent extension to multitask GPs to the CC setting, where the individual GPs can now live on different skeletons of the same CC. This will be useful in situations for modelling multiple fields (cochains) that are correlated.

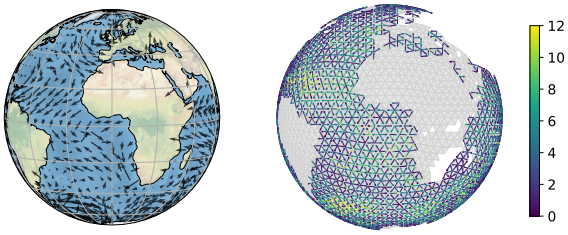


Figure 7. Left: Vector field on a manifold-with-boundaries; Right: Vector field's encoding on the edges of an oriented CC. Modelling such fields is straightforward with CC-GPs.

We note that at present, our reaction-diffusion kernel is controlled only by three hyperparameters r, c, d, which may be too restrictive for some modelling purposes. To make our model more flexible, one possible direction that we can consider is to make use of the Hodge decomposition to split the cochains into exact, co-exact and harmonic components. This will allow us to model each of these components separately, for added flexibility. In a recent work (Yang et al., 2024), the authors consider GPs to model edge signals on a graph and observed that modelling the exact, co-exact and harmonic signals separately can lead to improved results. For signals supported on a cellular complex, a natural analogue of the Hodge decomposition can be provided through the Dirac matrix (Calmon et al., 2023). Hence, a promising direction would be to combine this decomposition with our reaction-diffusion kernel to develop a more flexible extension that could fit more complex mixing of information between different cell-types.

Another current limitation of our model is the computational cost associated with computing the eigendecomposition of the Laplacian/Dirac matrix, which can grow very quickly with the size of the network. While this limits the size of networks we can work with, this expensive computation needs to be performed only once and not during training, so we expect one to be able to work with reasonably large networks, consisting of hundreds of thousands of cells.

7. Conclusion

We introduced Gaussian processes on cellular complexes as tools for probabilistic modelling on higher-order networks. We identify these as GPs defined over random chains or direct sums thereof, enabling inference on vertices, edges, and higher-order cells. We constructed kernels appropriate for practical modelling. In particular, we generalise the Matérn kernel to cellular complexes and propose the reaction-diffusion kernel, which allows for propagation of information between cells of different orders.

Acknowledgements

MA is supported by a Mathematical Sciences Doctoral Training Partnership held by Prof. Helen Wilson, funded by the Engineering and Physical Sciences Research Council (EPSRC), under Project Reference EP/W523835/1. ST is supported by a Department of Defense Vannevar Bush Faculty Fellowship held by Prof. Andrew Stuart, and by the SciAI Center, funded by the Office of Naval Research (ONR), under Grant Number N00014-23-1-2729.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning. However, due to its theoretical nature, we do not identify any potential societal consequences of our work.

References

Alvarez-Rodriguez, U., Battiston, F., de Arruda, G. F., Moreno, Y., Perc, M., and Latora, V. Evolutionary Dynamics of Higher-Order Interactions in Social Networks. *Nature Human Behaviour*, 2021. (Cited on page 1.)

Arnold, D. N., Falk, R. S., and Winther, R. Finite Element Exterior Calculus, Homological Techniques, and Applications. *Acta Numerica*, 2006. (Cited on pages 6 and 9.)

Barbarossa, S. and Sardellitti, S. Topological Signal Processing over Simplicial Complexes. *IEEE Transactions on Signal Processing*, 2020. (Cited on page 1.)

Battiloro, C., Testa, L., Giusti, L., Sardellitti, S., Lorenzo, P. D., and Barbarossa, S. Generalized Simplicial Attention Neural Networks. *arXiv preprint arXiv:2309.02138*, 2023. (Cited on page 2.)

Bianconi, G. The Topological Dirac Equation of Networks and Simplicial Complexes. *Journal of Physics: Complexity*, 2021. (Cited on page 2.)

Bodnar, C., Frasca, F., Otter, N., Wang, Y. G., Liò, P., Montúfar, G., and Bronstein, M. Weisfeiler and Lehman Go Cellular: CW Networks. In *Advances in Neural Information Processing Systems*, 2021. (Cited on page 1.)

Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth, M. Matérn Gaussian Processes on Riemannian Manifolds. In *Advances in Neural Information Processing Systems*, 2020. (Cited on pages 1, 5, and 23.)

Borovitskiy, V., Azangulov, I., Terenin, A., Mostowsky, P., Deisenroth, M. P., and Durrande, N. Matérn Gaussian Processes on Graphs. In *International Conference on Artificial Intelligence and Statistics*, 2021. (Cited on pages 2, 5, and 17.)

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: Composable Transformations of Python+NumPy Programs, 2018. URL http://github.com/google/jax. (Cited on page 23.)

Calmon, L., Schaub, M. T., and Bianconi, G. Dirac Signal Processing of Higher-Order Topological Signals. *New Journal of Physics*, 2023. (Cited on pages 2, 6, and 9.)

Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. In *International Conference on Learning Representations*, 2023. (Cited on page 1.)

Desbrun, M., Kanso, E., and Tong, Y. Discrete Differential Forms for Computational Modeling. In *ACM SIGGRAPH 2006 Courses*. 2006. (Cited on pages 7, 8, and 23.)

Fernandes, P., Allamanis, M., and Brockschmidt, M. Structured Neural Summarization. In *International Conference on Learning Representations*, 2019. (Cited on page 1.)

Filipovich, M. J. and Hughes, S. PyCharge: an Open-source Python Package for Self-Consistent Electrodynamics Simulations of Lorentz Oscillators and Moving Point Charges. *Computer Physics Communications*, 274: 108291, 2022. (Cited on page 8.)

Hajij, M., Istvan, K., and Zamzmi, G. Cell Complex Neural Networks. In *Workshop on Topological Data Analysis and Beyond at NeurIPS*, 2020. (Cited on page 1.)

Hatcher, A. *Algebraic Topology*. Cambridge University Press, 2001. (Cited on pages 1, 2, and 13.)

Hutchinson, M., Terenin, A., Borovitskiy, V., Takao, S., Teh, Y., and Deisenroth, M. Vector-Valued Gaussian Processes on Riemannian Manifolds via Gauge Independent Projected Kernels. In *Advances in Neural Information Processing Systems*, 2021. (Cited on pages 9 and 15.)

Kuzmin, E., VanderSluis, B., Wang, W., Tan, G., Deshpande, R., Chen, Y., Usaj, M., Balint, A., Usaj, M. M., van Leeuwen, J., Koch, E., Pons, C., Dagilis, A., Pryszlak, M., Wang, J. Z. Y., Hanchard, J., Riggi, M., Xu, K., Heydari, H., Luis, B.-J. S., Shuteriqi, E., Zhu, H., Dyk, N. V., Sharifpoor, S., Costanzo, M., Loewith, R., Caudy, A., Bolnick, D., Brown, G., Andrews, B., Boone, C., and Myers, C. Systematic Analysis of Complex Genetic Interactions. *Science*, 2018. (Cited on page 1.)

Nikitin, A., John, S., Solin, A., and Kaski, S. Non-separable Spatio-temporal Graph Kernels via SPDEs. In *International Conference on Artificial Intelligence and Statistics*, 2022. (Cited on page 1.)

NOAA CoastWatch. Sea Level Anomaly and Geostrophic Currents, Multi-mission, Global, Optimal Interpolation, Gridded. https://coastwatch.noaa.gov/cwn/products/sea-level-anomaly-and-geostrophic-currents-multi-mission-global-optimal-interpolation.html, 2023. Altimetry data are provided by the NOAA Laboratory for Satellite Altimetry. (Cited on pages 7 and 23.)

Opolka, F., Zhi, Y.-C., Liò, P., and Dong, X. Adaptive Gaussian Processes on Graphs via Spectral Graph Wavelets. In *International Conference on Artificial Intelligence and Statistics*, 2022. (Cited on page 1.)

Pinder, T. and Dodd, D. GPJax: a Gaussian Process Framework in JAX. *Journal of Open Source Software*, 2022. (Cited on page 23.)

Pinder, T., Turnbull, K., Nemeth, C., and Leslie, D. Gaussian Processes on Hypergraphs. *arXiv preprint arXiv:2106.01982*, 2021. (Cited on page 5.)

Robert-Nicoud, D., Krause, A., and Borovitskiy, V. Intrinsic Gaussian Vector Fields on Manifolds. In *International Conference on Artificial Intelligence and Statistics*, 2024. (Cited on page 9.)

Roddenberry, T. M., Schaub, M. T., and Hajij, M. Signal Processing on Cell Complexes. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022. (Cited on page 1.)

Rue, H. and Held, L. *Gaussian Markov Random Fields: Theory and Applications*. CRC Press, 2005. (Cited on pages 2 and 6.)

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 2008. (Cited on page 1.)

Smola, A. J. and Kondor, R. Kernels and Regularization on Graphs. In *Conference on Learning Theory*, 2003. (Cited on pages 1 and 2.)

Staniforth, A., Melvin, T., and Wood, N. Gungho! a New Dynamical Core for the Unified Model. In *ECMWF Workshop on Recent Developments in Numerical Methods for Atmosphere and Ocean Modelling*, 2013. (Cited on page 7.)

Yang, M., Borovitskiy, V., and Isufi, E. Hodge-Compositional Edge Gaussian Processes. In *International Conference on Artificial Intelligence and Statistics*, 2024. (Cited on pages 5 and 9.)

Yu, S., Yang, H., Nakahara, H., Santos, G., Nikolić, D., and Plenz, D. Higher-Order Interactions Characterized in Cortical Activity. *Journal of Neuroscience*, 2011. (Cited on page 1.)

Zhi, Y.-C., Ng, Y. C., and Dong, X. Gaussian Processes on Graphs via Spectral Kernel Learning. *IEEE Transactions on Signal and Information Processing over Networks*, 2023. (Cited on page 1.)

A. Cell Orientation and Boundaries

In this appendix, we provide more details on the computation of cell boundaries. To this end, we first require a notion of orientation on cells. For low dimensional k-cells, we have the following intuitive definitions of what we mean by an orientation:

- $k = 0$ (i.e., a point): A choice of either "$+$" or "$-$".
- $k = 1$ (i.e., a line segment): A choice of direction from one endpoint to the other.
- $k = 2$ (i.e., a 2D-disk): A choice of rotation direction (clockwise or anticlockwise).

Generally, the orientation of a k-dimensional unit disk D^k is the choice of a continuously varying unit normal vector field $\hat{\boldsymbol{n}} : D^k \to \mathbb{R}^{k+1}$ on D^k, viewed as a surface embedded in $\mathbb{R}^{k+1}$. Here, a unit normal vector $\hat{\boldsymbol{n}}(s)$ for $s \in D^k$ is a vector in $\mathbb{R}^{k+1}$ such that $\|\hat{\boldsymbol{n}}(s)\| = 1$ and $\hat{\boldsymbol{n}}(s) \cdot \boldsymbol{v} = 0$ for any $\boldsymbol{v} \in T_s D^k \subset \mathbb{R}^{k+1}$, where $T_s D^k$ is the tangent space of D^k at point s (i.e., the space of all vectors in $\mathbb{R}^{k+1}$ that are tangential to D^k). Since orientation is a topological invariant, we can define the orientation on a k-cell generally to be the orientation of the k-dimensional disk that it is homeomorphic to.

A useful way of thinking about the orientation of D^k is in terms of how it is embedded in $\mathbb{R}^{k+1}$. In particular, we may identify a point $x \in D^k$ as a vector $(x_1, \ldots, x_k, 0)^\top \in \mathbb{R}^{k+1}$ such that $\sqrt{\sum_{i=1}^k x_i^2} < 1$. We can thus choose the unit normal field to be given by $\hat{\boldsymbol{n}} = (x_1, \ldots, x_k, 1)^\top \in \mathbb{R}^{k+1}$ for any $x_1, \ldots, x_k$ parameterising D^k. From this perspective, we can define an *induced orientation* $\partial\hat{\boldsymbol{n}}$ on its boundary $\partial D^k \cong \mathbb{S}^{k-1}$, by taking the unit normal field pointing outwards from the disk, with respect to this embedding.

(a) Orientation and induced orientation on a 2-cell

(b) Orientation and induced orientation on a 1-cell

Figure 8. Illustration of orientations (blue arrows) on (a) a 2-cell, and (b) a 1-cell, along with the induced orientation (red arrows) on the corresponding boundaries. The orientation can be understood as a choice of a continuous vector field in the ambient Euclidean space that points in the direction normal to the k-cell (in this case, pointing upwards). The induced orientation on the boundaries can be understood as the outward pointing normal field with respect to the embedding.

Example 15. *Consider a 2D-disk D^2 embedded in $\mathbb{R}^3$. The orientation on D^2 is then given by the unit normal field $\hat{\boldsymbol{n}}(x, y, 0) = (x, y, 1)$ for $\sqrt{x^2 + y^2} < 1$ and the induced orientation on its boundary $\mathbb{S}^1 \cong \partial D^2$ is given by the outward unit normal field $\partial\hat{\boldsymbol{n}}(x, y, 0) = (x, y, 0)$ for $\sqrt{x^2 + y^2} = 1$. Intuitively, the former can be thought of as anticlockwise rotation of the disk, deduced by aligning one's thumb with the unit normal direction and applying the right-hand rule. Now, aligning the thumb with $\hat{\boldsymbol{n}}$ and the index finger with $\partial\hat{\boldsymbol{n}}$, the middle finger, according to the right-hand rule, points in the anticlockwise direction around the boundary, giving us a more intuitive interpretation of the induced orientation (see Figure 8a).*

Example 16. *Consider a line-segment (i.e., a 1D disk) embedded in $\mathbb{R}^2$ with respect to the parameterisation $\{(x, 0) \in \mathbb{R}^2 : x \in (-1, 1)\}$. The orientation of the line segment is determined by the unit normal field $\hat{\boldsymbol{n}}(x, 0) = (x, 1)$ for $x \in (-1, 1)$ and the induced orientation is given by the outward normal field $\partial\hat{\boldsymbol{n}}(x, 0) = (x, 0)$ for $x \in \{-1, 1\}$. Again, we can use the right-hand rule to determine the "intuitive" interpretation of orientations here: pointing the thumb towards the page and the index finger aligned with $\hat{\boldsymbol{n}}$, the middle finger points in the direction going from left endpoint $x = -1$ to the right endpoint $x = 1$. Since the middle finger as a result aligns with $\partial\hat{\boldsymbol{n}}(x, 0)$ at $x = 1$, we think of the induced orientation at $(1, 0)$ as having the sign "$+$". However, it does not align with $\partial\hat{\boldsymbol{n}}(x, 0)$ at $x = -1$, hence the induced orientation at $(-1, 0)$ has the sign "$-$" (see Figure 8b).*

Next, the *boundary* of a k-cell e^k_α, is defined generally as a $(k-1)$-chain

$$\partial e^k_\alpha = \sum_{\beta=1}^{N_{k-1}} \deg(\chi^{\alpha\beta}) e^{k-1}_\beta, \tag{27}$$

where $\deg(\chi^{\alpha\beta})$ is the *Brouwer degree* of the surjection $\chi^{\alpha\beta} : \mathbb{S}^{k-1} \cong \partial e^k_\alpha \xrightarrow{\phi^k_\alpha} X^{k-1} \xrightarrow{/} X^{k-1}/(X^{k-1} - e^{k-1}_\beta) \cong \mathbb{S}^{k-1}$, mapping the boundary of e^k_α (homeomorphic to $\mathbb{S}^{k-1}$) to the $(k-1)$-cell e^{k-1}_β, identified with $\mathbb{S}^{k-1}$ by collapsing its boundary to a single point (Hatcher, 2001). For a regular cellular complex, $\deg(\chi^{\alpha\beta})$ is either 0 or ± 1 depending on whether or not e^{k-1}_β is a part of ∂e^k_α under the attaching map. That is, if e^{k-1}_β is not part of the boundary ∂e^k_α, then we take $\deg(\chi^{\alpha\beta}) = 0$. Otherwise, if the induced orientation of ∂e^k_α aligns with the orientation of e^{k-1}_β (i.e., take an embedding $X^{k-1} \hookrightarrow \mathbb{R}^k$ and see how the unit normal fields of ∂e^k_α and e^{k-1}_β align), then we take $\deg(\chi^{\alpha\beta}) = 1$ and if it has opposite orientations, we take $\deg(\chi^{\alpha\beta}) = -1$. We will also adopt the notation $\deg(e^k_\alpha \to e^{k-1}_\beta)$ to denote $\deg(\chi^{\alpha\beta})$ in order to make the cells involved more explicit.

The computation of the Brouwer degree can in general be challenging. However, in some special cases, the computation can be simplified, as we show in the following examples.

Example 17. *Consider a k-simplex, which can be identified as a collection of $k+1$ vertices, say $v_0, \dots, v_k$ without loss of generality. Its orientation is determined by the parity of the permutation of these $k+1$ vertices. Hence, we represent it by an equivalence class $[v_0, \dots, v_k]$, where equivalence is defined by the parity. We set $e^k_\alpha = [v_0, \dots, v_k]$. Now consider an oriented face of this simplex, represented by an equivalence class of k vertices $e^{k-1}_\beta = [w_1, \dots, w_k]$, such that $\{w_1, \dots, w_k\} = \{v_0, \dots, \cancel{v_\ell}, \dots, v_k\}$ for some $\ell \in \{0, \dots, k\}$. Naturally, this can be represented by a permutation $\sigma \in S_{k+1}$ on $\{0, \dots, k\}$ with $\sigma(0) = \ell$ and $w_i = v_{\sigma(i)}$ for $i = 1, \dots, k$. Then, we take $\deg(e^k_\alpha \to e^{k-1}_\beta) = \mathrm{sgn}(\sigma)$, the signature of the permutation σ. Thus, in general, we can write*

$$\partial[v_1, \dots, v_{k+1}] = \sum_{\ell=1}^{k+1} (-1)^\ell [v_1, \dots, \cancel{v_\ell}, \dots, v_{k+1}]. \tag{28}$$

Example 18. *Consider a 2D polygonal cell e^k_α with vertices $v_1, \dots, v_k$, and re-order them such that it revolves around the polygon in a clockwise or anticlockwise manner: i.e., $v_{\sigma(1)} \to v_{\sigma(2)} \to \cdots \to v_{\sigma(k-1)} \to v_{\sigma(k)} \to v_{\sigma(1)}$ for some permutation $\sigma \in S_k$. Whether the ordering here is clockwise or anticlockwise determines the orientation of the cell. Since the choice of the first vertex in the ordering is not important, we represent this polygon as the equivalence class $e^k_\alpha = [v_{\sigma(1)}, \dots, v_{\sigma(k)}]$, where equivalence is defined by permutation with respect to the cyclic group $C_k \subset S_k$. Now, an oriented edge of this polygon can be represented by an ordered tuple $e^{k-1}_\beta = (v_i, v_j)$ for $i < j$, such that either $(i, j) = (\sigma(\ell), \sigma(\ell+1))$ or $(i, j) = (\sigma(\ell+1), \sigma(\ell))$ for some $\ell \in \{1, \dots, k\}$ (we use the convention $k+1 \equiv 1$, i.e., assume the indices are elements of $\mathbb{Z}/k\mathbb{Z}$). Then, we set $\deg(e^k_\alpha \to e^{k-1}_\beta) = 1$ in the former case and $\deg(e^k_\alpha \to e^{k-1}_\beta) = -1$ in the latter. Hence, using the convention $(v_i, v_j) = -(v_j, v_i)$, we can check that*

$$\partial[v_{\sigma(1)}, \dots, v_{\sigma(k)}] = \sum_{\ell=1}^{k+1} (v_{\sigma(\ell)}, v_{\sigma(\ell+1)}). \tag{29}$$

In the general case, we can compute the boundary of an arbitrary cell by considering its *simplicial decomposition*. That is, we discretise the k-cell e^k_α by a collection of N k-simplices. This can be expressed as a k-chain

$$e^k_\alpha = \sum_{i=1}^{N} [v_{i_1}, \dots, v_{i_k}]. \tag{30}$$

We can compute its boundary by first taking

$$\partial e^k_\alpha = \sum_{i=1}^{N} \partial[v_{i_1}, \dots, v_{i_k}], \tag{31}$$

then using (28) to give a simplicial decomposition of ∂e^k_α, and finally collectivising with respect to the simplicial decomposition of e^{k-1}_β.

B. Numerical Representation of Cellular Complexes

In this appendix, we provide a full derivation of the numerical representation of key concepts on cellular complexes, as displayed in Table 1. To make this appendix self-contained, we first re-introduce some definitions.

Definition 19 (k-chains)**.** *A k-chain on X is a free Abelian group whose generator is the set of all k-cells comprising X. The space of all k-chains on X is denoted $C_k(X)$.*

Definition 20 (Boundary operators)**.** *For $k \in \{1, \dots, n\}$, the boundary operator is a group homomorphism $\partial_k : C_k(X) \to C_{k-1}(X)$ mapping k-chains to $(k-1)$-chains, i.e.,*

$$\partial_k \left(\sum_{\alpha=1}^{N_k} n_a e_\alpha^k \right) = \sum_{\alpha=1}^{N_k} n_a \partial e_\alpha^k, \tag{32}$$

where ∂e_α^k is the boundary of the cell e_k^α, viewed as a $(k-1)$-chain (see Appendix A). For convention, we also take $\partial_k c \equiv 0$ for $k \in \{0, n+1\}$.

Definition 21 (k-cochains)**.** *A k-cochain on X is a group homomorphism $f : C_k(X) \to \mathbb{R}$ assigning real numbers to k-chains, i.e.,*

$$f\Big(\sum_{\alpha=1}^{N_k} n_\alpha e_\alpha^k \Big) = \sum_{\alpha=1}^{N_k} n_\alpha f(e_\alpha^k), \tag{33}$$

where $f(e_\alpha^k) \in \mathbb{R}$ is the value of f at cell e_α^k. The space of all k-cochains on X is denoted $C^k(X)$ and forms a real vector space.

Definition 22 (Coboundary operators)**.** *For $0 \leq k < n$, the coboundary operator $d_k : C^k(X) \to C^{k+1}(X)$ is a linear map defined via the relation*

$$d_k f(c) = f(\partial_{k+1} c) \tag{34}$$

for all $f \in C^k(X)$ and $c \in C_{k+1}(X)$. For convention, we also take $d_k f \equiv 0$ for $k \in \{-1, n\}$.

Definition 23 (L^2-inner product on cochains)**.** *For each k, let $\{w_\alpha^k\}_{\alpha=1}^{N_k}$ be a set of real, positive weights. Then, for any $f, g \in C^k(X)$, we define the weighted L^2-inner product as*

$$\langle f, g \rangle_{L^2(\boldsymbol{w}^k)} := \sum_{\alpha=1}^{N_k} w_\alpha^k \, f(e_\alpha^k) \, g(e_\alpha^k). \tag{35}$$

Definition 24 (L^2-adjoint of the coboundary operator)**.** *For each k, let $\{w_\alpha^k\}_{\alpha=1}^{N_k}$ be a set of real, positive weights. The L^2-adjoint of the coboundary operator, denoted $d_k^* : C^{k+1}(X) \to C^k(X)$ is defined by*

$$\langle d_k^* f, g \rangle_{L^2(\boldsymbol{w}^k)} = \langle f, d_k g \rangle_{L^2(\boldsymbol{w}^{k+1})}, \tag{36}$$

for any $f \in C^{k+1}(X)$ and $g \in C^k(X)$.

Definition 25 (Hodge Laplacian)**.** *The Hodge Laplacian $\Delta_k : C^k(X) \to C^k(X)$ on the space of k-cochains is defined as*

$$\Delta_k := d_{k-1} \circ d_{k-1}^* + d_k^* \circ d_k. \tag{37}$$

To make explicit computations with them, we wish to represent these using matrices and vectors. Fortunately, this is not difficult as the space of chains / cochains forms a free Abelian group / vector space, which is naturally isomorphic to $\mathbb{Z}^n$ / $\mathbb{R}^n$.

To this end, we fix a labelling $\alpha \mapsto e_\alpha^k$ of the k-cells comprising a cellular complex X, which forms an ordered basis $(e_1^k, \dots, e_{N_k}^k)$. Then, an arbitrary k-chain $c = \sum_{\alpha=1}^{N_k} n_\alpha e_\alpha^k \in C_k(X)$ may be represented by a vector $\boldsymbol{c} = (n_1, \dots, n_{N_k})^\top$ in $\mathbb{Z}^{N_k}$. Similarly, a k-cochain $f = \sum_{\alpha=1}^{N_k} f_\alpha (e_\alpha^k)^* \in C^k(X)$ can be represented by a vector $\boldsymbol{f} = (f_1, \dots, f_{N_k})^\top$ in $\mathbb{R}^{N_k}$. Under this representation, cochain evaluation (33) can simply be expressed as a dot product $f(c) = \boldsymbol{f}^\top \boldsymbol{c} \in \mathbb{R}$.

Next, we consider the boundary and coboundary operators. The boundary operator can be expressed as a signed incidence matrix $\mathbf{B}_k \in \mathbb{Z}^{N_{k-1} \times N_k}$, whose j-th column corresponds to the vector representation of the cell boundary ∂e_j^k, viewed as a $(k-1)$-chain (see Appendix A). That is,

$$[\mathbf{B}_k]_{ij} = \deg(e_j^k \to e_i^{k-1}) \tag{38}$$

for $k \in \{1, \dots, n\}$, and by convention, we take $\mathbf{B}_k \equiv \mathbf{0}$ for $k \in \{0, n+1\}$. Similarly, the coboundary operator can be represented by a matrix $\mathbf{D}_k \in \mathbb{R}^{N_{k+1} \times N_k}$. Using (34), we have

$$\boldsymbol{f}^\top \mathbf{D}_k^\top \boldsymbol{c} = \boldsymbol{f}^\top \mathbf{B}_{k+1} \boldsymbol{c}, \tag{39}$$

$$\Leftrightarrow \quad \mathbf{D}_k = \mathbf{B}_{k+1}^\top. \tag{40}$$

Thus, the coboundary operator is identified with the transpose of the signed incidence matrix. Finally, let $\mathbf{W}_k = \texttt{diag}(w_1^k, \dots, w_{N_k}^k)$ be the weight matrix defining the L^2-inner product (35). i.e.,

$$\langle f, g \rangle_{L^2(\boldsymbol{w}^k)} = \boldsymbol{f}^\top \mathbf{W}_k \boldsymbol{g}. \tag{41}$$

Then, letting $\mathbf{D}_k^* \in \mathbb{R}^{N_k \times N_{k+1}}$ be the matrix representation of the adjoint of the coboundary, (36) implies

$$\boldsymbol{f}^\top (\mathbf{D}_k^*)^\top \mathbf{W}_k \boldsymbol{g} = \boldsymbol{f}^\top \mathbf{W}_{k+1} \mathbf{B}_{k+1}^\top \boldsymbol{g} \tag{42}$$

$$\Leftrightarrow \quad \mathbf{D}_k^* = \mathbf{W}_k^{-1} \mathbf{B}_{k+1} \mathbf{W}_{k+1}. \tag{43}$$

Putting this together, we find the matrix expression $\boldsymbol{\Delta}_k \in \mathbb{R}^{N_k \times N_k}$ for the Hodge Laplacian operator using (37):

$$\boldsymbol{\Delta}_k = \mathbf{D}_k \mathbf{D}_k^* + \mathbf{D}_{k+1}^* \mathbf{D}_{k+1} \tag{44}$$

$$= \mathbf{B}_k^\top (\mathbf{W}_{k-1}^{-1} \mathbf{B}_k \mathbf{W}_k) + (\mathbf{W}_k^{-1} \mathbf{B}_{k+1} \mathbf{W}_{k+1}) \mathbf{B}_{k+1}^\top. \tag{45}$$

C. Characterisation of GPs on Cellular Complexes

Here, we prove the results in Section 4 characterising Gaussian random cochains and GPs on cellular complexes by a mean and a kernel.

C.1. Proof of Theorem 8

We restate Theorem 8 below for completeness.

Theorem 26. *A Gaussian random cochain $f : \Omega \to C^k(X)$ is fully characterised by a mean $\mu \in C^k(X)$ and a kernel $\kappa : C_k(X) \times C_k(X) \to \mathbb{R}$.*

Proof. The proof is almost identical to that of Lemmas 9–10 in (Hutchinson et al., 2021).

($\Rightarrow$) First, we show that given a Gaussian random cochain f, we can define a mean and a kernel object. For this, we simply set $\mu(c) := \mathbb{E}[f(c)]$ and $\kappa(c, c') := \mathrm{Cov}[f(c), f(c')]$. The former is clearly a cochain since f is a (random) cochain. The latter can be easily checked to be a kernel in the sense of Definition 7:

- (Symmetry) This follows from the symmetry of the covariance operator $\mathrm{Cov}[f(c), f(c')] = \mathrm{Cov}[f(c'), f(c)]$.
- (Group bi-homomorphism) This follows from the fact that by definition, f is a group homomorphism and using the bilinearity of the covariance operator.
- (Positive semi-definiteness) Fixing $c_1, \dots, c_m \in C_k(X)$, we have

$$\sum_{\alpha,\beta=1}^{m} \kappa(c_\alpha, c_\beta) = \sum_{\alpha,\beta=1}^{m} \mathrm{Cov}[f(c_\alpha), f(c_\beta)] = \mathbb{E}\left[\left(\sum_{\alpha=1}^{m} (f(c_\alpha) - \mathbb{E}[f(c_\alpha)])\right)^2\right] \geq 0. \tag{46}$$

($\Leftarrow$) Next, we show that given $\mu \in C^k(X)$ and a kernel $\kappa : C_k(X) \times C_k(X) \to \mathbb{R}$, we can construct a Gaussian random cochain. Take

$$\boldsymbol{\mu} = \begin{bmatrix} \mu(e_1^k) \\ \vdots \\ \mu(e_{N_k}^k) \end{bmatrix} \in \mathbb{R}^{N_k}, \qquad \mathbf{K} = \begin{bmatrix} k(e_1^k, e_1^k) & \cdots & k(e_1^k, e_{N_k}^k) \\ \vdots & \ddots & \vdots \\ k(e_{N_k}^k, e_1^k) & \cdots & k(e_{N_k}^k, e_{N_k}^k) \end{bmatrix} \in \mathbb{R}^{N_k \times N_k}. \tag{47}$$

This uniquely defines a multivariate Gaussian random variable $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$. Now let $\varphi : C^k(X) \xrightarrow{\sim} \mathbb{R}^{N_k}$ be the group isomorphism identifying k-cochains by a vector in $\mathbb{R}^{N_k}$ via the labelling $\alpha \mapsto e_\alpha^k$. We also take the group isomorphism $\psi : C_k(X) \xrightarrow{\sim} \mathbb{Z}^{N_k}$, defined by $f(c) = \varphi(f)^\top \psi(c)$ for any $f \in C^k(X)$ and $c \in C_k(X)$. Then, we set $f := \varphi^{-1}\boldsymbol{f}$, which defines a Gaussian random cochain, since for any $c \in C_k(X)$, we have $f(c) = [\varphi^{-1}\boldsymbol{f}](c) = \boldsymbol{f}^\top \psi(c)$, which is univariate Gaussian. □

C.2. Proof of Theorem 10

We restate Theorem 10 below for completeness.

Theorem 27. *A Gaussian process on a cellular complex X (abbreviated as CC-GP) is fully characterised by a mean $\mu \in \bigoplus_{k=0}^n C^k(X)$ and a kernel $\kappa : \bigoplus_{k=0}^n C_k(X) \times \bigoplus_{k=0}^n C_k(X) \to \mathbb{R}$.*

Proof. The proof is almost identical to the proof of Theorem 8 (see Appendix C.1), hence we will omit some details to avoid repetition.

($\Rightarrow$) For a CC-GP $f : \Omega \to \bigoplus_{k=0}^n C^k(X)$, we can define a mean and a kernel by setting $\mu(c) := \mathbb{E}[f(c)]$ and $\kappa(c, c') := \mathrm{Cov}[f(c), f(c')]$, for any $c, c' \in \bigoplus_{k=0}^n C^k(X)$.

($\Leftarrow$) Given $\mu \in \bigoplus_{k=0}^n C^k(X)$ and a kernel $\kappa : \bigoplus_{k=0}^n C_k(X) \times \bigoplus_{k=0}^n C_k(X) \to \mathbb{R}$, we take

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \vdots \\ \boldsymbol{\mu}_n \end{pmatrix}, \quad \mathbf{K} = \begin{pmatrix} \mathbf{K}_{11} & \cdots & \mathbf{K}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{n1} & \cdots & \mathbf{K}_{nn} \end{pmatrix} \tag{48}$$

with $[\boldsymbol{\mu}_i]_j = \mu(e_j^i)$ and $[\mathbf{K}_{nm}]_{ij} = \kappa(e_i^n, e_j^m)$, which uniquely defines a multivariate Gaussian random variable $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$ in $\mathbb{R}^{N_1 + \cdots + N_n}$. Now, consider the group isomorphism $\varphi : \bigoplus_{k=0}^n C^k(X) \xrightarrow{\sim} \mathbb{R}^{N_1 + \cdots + N_n}$ by fixing a labelling $\alpha \mapsto e_\alpha^k$ for each $k = 0, \ldots, n$. Then, $f := \varphi^{-1}\boldsymbol{f}$ defines a CC-GP. □

D. Kernels on Cellular Complexes

In this appendix, we provide further details on the kernels that we consider in this paper, namely the Matérn kernel on cellular complexes (CC-Matérn) and the reaction-diffusion kernel (RD). We will need the following definition to formalise our kernels.

Definition 28. *Consider the weighted L^2-inner product $\langle \cdot, \cdot \rangle_{L^2(\boldsymbol{w}^k)} : C^k(X) \times C^k(X) \to \mathbb{R}$ on the space of k-cochains (Definition 23). We define a group homomorphism $\flat : C_k(X) \to C^k(X)$ by*

$$f(c) = \left\langle f, c^\flat \right\rangle_{L^2(\boldsymbol{w}^k)}, \tag{49}$$

for any $c \in C_k(X)$ and $f \in C^k(X)$. The existence of $c^\flat$ follows from the Riesz representation theorem.

Next, we introduce the concept of a Gaussian white-noise cochain, defined as follows.

Definition 29 (Gaussian white-noise cochain)**.** *We define a zero-mean Gaussian white noise cochain as a Gaussian random cochain $\mathcal{W} : \Omega \to C^k(X)$ satisfying*

- $\mathbb{E}[\langle \mathcal{W}, f \rangle_{L^2(\boldsymbol{w}^k)}] = 0$ *for any* $f \in C^k(X)$.
- $\mathbb{E}[\langle \mathcal{W}, f \rangle_{L^2(\boldsymbol{w}^k)} \langle \mathcal{W}, g \rangle_{L^2(\boldsymbol{w}^k)}] = \langle f, g \rangle_{L^2(\boldsymbol{w}^k)}$ *for any* $f, g \in C^k(X)$

Now, we are ready to define our notion of the Matérn kernel on cellular complexes.

D.1. Matérn Kernel

For simplicity, let us assume for now that the cell weights are $w_\alpha^k = 1$ for all k, α. In this special case, we will use the shorthand notation $\langle \cdot, \cdot \rangle_{L^2} \equiv \langle \cdot, \cdot \rangle_{L^2(\boldsymbol{w}^k)}$. We will deal with the more general case in Appendix D.3.1.

Following the construction in (Borovitskiy et al., 2021), we define a Matérn Gaussian random k-cochain as a solution to the stochastic system

$$\left(\frac{2\nu}{\ell^2} + \Delta_k\right)^{\nu/2} f = \mathcal{W}, \tag{50}$$

where $f \in C^k(X)$ and $\mathcal{W} : \Omega \to C^k(X)$ is the Gaussian white-noise cochain (Definition 29).

The operator $\left(\frac{2\nu}{\ell^2} + \Delta_k\right)^{\nu/2}$ is to be understood as an operation in frequency space, by the following construction. Let $\{(\lambda_i^2, u_i)\}_{i=1}^{N_k}$, be solutions to the eigenproblem $\Delta_k u_i = \lambda_i^2 u_i$ such that the eigencochains $\{u_i\}_{i=1}^{N_k}$ are orthonormal in L^2. Representing f as $f = \sum_i \langle f, u_i \rangle_{L^2(w^k)} u_i$, we define

$$\left(\frac{2\nu}{\ell^2} + \Delta_k\right)^{\nu/2} f := \sum_{i=1}^{N_k} \left(\frac{2\nu}{\ell^2} + \lambda_i^2\right)^{\nu/2} \langle f, u_i \rangle_{L^2} u_i, \tag{51}$$

which is a linear operator on the space of k-cochains. We define the Matérn kernel $\kappa : C_k(X) \times C_k(X) \to \mathbb{R}$ as as a solution to the linear system

$$\left(\frac{2\nu}{\ell^2} + \Delta_k\right)^{\nu} \kappa(c, \cdot) = c^\flat, \quad \forall c \in C_k(X). \tag{52}$$

D.1.1. Proof of Proposition 12

We restate Proposition 12 below for completeness.

Proposition 30. *The solution to* (52) *is related to the solution f of the system* (50) *as:*

$$\kappa(c, c') = \mathbb{E}[f(c) f(c')], \quad \forall c, c' \in C_k(X) \tag{53}$$

Thus, κ is the kernel corresponding to the Gaussian random cochain f.

Proof. We first claim that the unique solution f to (50) can be represented as

$$f = \sum_i \left(\frac{2\nu}{\ell^2} + \lambda_i^2\right)^{-\nu/2} \langle \mathcal{W}, u_i \rangle_{L^2} u_i. \tag{54}$$

This can be checked by simply substituting this expression inside (51) and using the L^2-orthonormality of the eigencochains $\{u_i\}_{i=1}^{N_k}$. The uniqueness can be checked by the linearity of the operator $\left(\frac{2\nu}{\ell^2} + \Delta_k\right)^{\nu/2}$ and the fact that the solution to the system $\left(\frac{2\nu}{\ell^2} + \Delta_k\right)^{\nu/2} f = 0$ is satisfied only by $f \equiv 0$.

Similarly, the solution to (52) is given by

$$\kappa(c, \cdot) = \sum_i \left(\frac{2\nu}{\ell^2} + \lambda_i^2\right)^{-\nu} \left\langle c^\flat, u_i \right\rangle_{L^2} u_i(\cdot) \tag{55}$$

$$\overset{(49)}{=} \sum_i \left(\frac{2\nu}{\ell^2} + \lambda_i^2\right)^{-\nu} u_i(c) u_i(\cdot) \tag{56}$$

Next, we show that for arbitrary $c, c' \in C_k(X)$, we have

$$\mathbb{E}[f(c)f(c')] = \mathbb{E}\left[\sum_i \sum_j \left(\frac{2\nu}{\ell^2} + \lambda_i^2\right)^{-\nu/2} \left(\frac{2\nu}{\ell^2} + \lambda_j^2\right)^{-\nu/2} \langle \mathcal{W}, u_i \rangle_{L^2} \langle \mathcal{W}, u_j \rangle_{L^2} u_i(c) u_j(c')\right] \tag{57}$$

$$= \sum_i \sum_j \left(\frac{2\nu}{\ell^2} + \lambda_i^2\right)^{-\nu/2} \left(\frac{2\nu}{\ell^2} + \lambda_j^2\right)^{-\nu/2} \mathbb{E}\left[\langle \mathcal{W}, u_i \rangle_{L^2} \langle \mathcal{W}, u_j \rangle_{L^2}\right] u_i(c) u_j(c') \tag{58}$$

$$= \sum_i \sum_j \left(\frac{2\nu}{\ell^2} + \lambda_i^2\right)^{-\nu/2} \left(\frac{2\nu}{\ell^2} + \lambda_j^2\right)^{-\nu/2} \underbrace{\langle u_i, u_j \rangle_{L^2}}_{=\delta_{ij}} u_i(c) u_j(c') \tag{59}$$

$$= \sum_i \left(\frac{2\nu}{\ell^2} + \lambda_i^2\right)^{-\nu} u_i(c) u_i(c') \tag{60}$$

$$\overset{(56)}{=} \kappa(c, c'), \tag{61}$$

verifying property (53). Finally, we show that κ is indeed a kernel. The symmetry of κ can be easily verified from the explicit expression (56), that is, $\kappa(c, c') = \kappa(c', c)$ for any $c, c' \in C_k(X)$. Checking that κ is a group bi-homomorphism also follows easily from expression (56) using the fact that u_i is a group homomorphism by the definition of cochains. Fixing $c_1, \ldots, c_m \in C_k(X)$ such that $c_\alpha \neq 0$ for some α, we also have

$$\sum_{\alpha,\beta=1}^{m} \kappa(c_\alpha, c_\beta) = \sum_{\alpha,\beta=1}^{m} \sum_i \left(\frac{2\nu}{\ell^2} + \lambda_i^2\right)^{-\nu} u_i(c_\alpha) u_i(c_\beta) \tag{62}$$

$$= \sum_i \left(\frac{2\nu}{\ell^2} + \lambda_i^2\right)^{-\nu} \sum_{\alpha=1}^{m} u_i(c_\alpha) \sum_{\beta=1}^{m} u_i(c_\beta) \tag{63}$$

$$= \sum_i \left(\frac{2\nu}{\ell^2} + \lambda_i^2\right)^{-\nu} \left(\sum_{\alpha=1}^{m} u_i(c_\alpha)\right)^2 \tag{64}$$

$$> 0, \tag{65}$$

verifying the positive-definiteness of κ. Hence, κ is a kernel on $C_k(X)$, as expected. □

D.1.2. Matrix representation

From (56), we can deduce the matrix representation of the Matérn kernel as

$$\mathbf{K} = \mathbf{U}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \mathbf{\Lambda}^2\right)^{-\nu} \mathbf{U}^\top. \tag{66}$$

Another way to derive this representation is by directly considering the numerical representation of system (50):

$$\mathbf{L}\boldsymbol{f} = \boldsymbol{w}, \tag{67}$$

where

$$\mathbf{L} := \mathbf{U}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \mathbf{\Lambda}^2\right)^{\nu/2} \mathbf{U}^\top \tag{68}$$

is the numerical representation of the linear operator (51), and $\boldsymbol{w} \sim \mathcal{N}(0, \mathbf{I})$. Then, we have

$$\boldsymbol{f} = \mathbf{L}^{-1}\boldsymbol{w} \sim \mathcal{N}(0, \mathbf{K}), \tag{69}$$

where

$$\mathbf{K} = \mathbf{L}^{-1}\mathbf{L}^{-\top}. \tag{70}$$

We claim that

$$\mathbf{L}^{-1} = \mathbf{U}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \mathbf{\Lambda}^2\right)^{-\nu/2}\mathbf{U}^\top. \tag{71}$$

This can be verified by computing

$$\begin{aligned}
\mathbf{L}^{-1}\mathbf{L} &= \mathbf{U}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \mathbf{\Lambda}^2\right)^{-\nu/2}\underbrace{\mathbf{U}^\top\mathbf{U}}_{=\mathbf{I}}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \mathbf{\Lambda}^2\right)^{\nu/2}\mathbf{U}^\top && (72)\\
&= \mathbf{U}\underbrace{\left(\frac{2\nu}{\ell^2}\mathbf{I} + \mathbf{\Lambda}^2\right)^{-\nu/2}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \mathbf{\Lambda}^2\right)^{\nu/2}}_{=\mathbf{I}}\mathbf{U}^\top && (73)\\
&= \mathbf{U}\mathbf{U}^\top && (74)\\
&= \mathbf{I}, && (75)
\end{aligned}$$

and similarly, $\mathbf{L}\mathbf{L}^{-1} = \mathbf{I}$. Then, we can check that indeed we have

$$\begin{aligned}
\mathbf{K} &= \mathbf{L}^{-1}\mathbf{L}^{-\top} && (76)\\
&= \mathbf{U}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \mathbf{\Lambda}^2\right)^{-\nu/2}\underbrace{\mathbf{U}^\top\mathbf{U}}_{=\mathbf{I}}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \mathbf{\Lambda}^2\right)^{-\nu/2}\mathbf{U}^\top && (77)\\
&= \mathbf{U}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \mathbf{\Lambda}^2\right)^{-\nu}\mathbf{U}^\top. && (78)
\end{aligned}$$

D.2. Reaction-diffusion Kernel

Here, we provide further details on the reaction-diffusion kernel, presented in Section 4.3.2. Since many of the ideas are similar to the Matérn kernel (Appendix D.1), we omit some details. We first lift the Dirac operator (10) to the direct sum space $\bigoplus_{k=1}^n C^k(X)$, where it is more natural as it defines a group homomorphism to itself (i.e., an endomorphism) $\mathcal{D} : \bigoplus_{k=1}^n C^k(X) \to \bigoplus_{k=1}^n C^k(X)$. This is given explicitly as

$$\mathcal{D}f = \sum_k \delta_k f_k = \begin{pmatrix} d_0^* f_1 \\ d_0 f_0 + d_1^* f_2 \\ \vdots \\ d_{n-2} f_{n-2} + d_{n-1}^* f_n \\ d_{n-1} f_{n-1} \end{pmatrix}. \tag{79}$$

Using the property $d_{k+1} \circ d_k = 0$ (equivalently, $d_k^* \circ d_{k+1}^* = 0$) of the coboundary operator, one can check that $\mathcal{D}^2 = \mathcal{L}$ (the super-Laplacian operator) holds. We also extend the L^2 inner-product to the direct sum space $\bigoplus_{k=1}^n C^k(X)$, which we define by

$$\langle f, g\rangle_{L^2(\boldsymbol{w})} := \sum_{k=1}^n \sum_{\alpha=1}^{N_k} w_\alpha^k\, f(e_\alpha^k)\, g(e_\alpha^k). \tag{80}$$

This trivially lifts Definition 23 and Definition 29 to the direct sum setting, defining a group homomorphism $\flat : \bigoplus_{k=1}^n C_k(X) \to \bigoplus_{k=1}^n C^k(X)$ and a Gaussian white noise process $\mathcal{W} : \Omega \to \bigoplus_{k=1}^n C^k(X)$, respectively.

Now let $\mathcal{W} : \Omega \to \bigoplus_{k=1}^n C^k(X)$ be the white noise process on the direct sum space. We define the reaction-diffusion GP to be the solution to the stochastic system

$$(r + c\mathcal{D} + d\mathcal{L})^{\nu/2} f = \mathcal{W}. \tag{81}$$

As before, the operator $(r + c\mathcal{D} + d\mathcal{L})^{\nu/2}$ is to be understood as an operation in frequency space, as follows. Let $\{(\lambda_i, u_i)\}_{i=1}^{N_1+\cdots+N_n}$, be solutions to the eigenproblem $\mathcal{D}u_i = \lambda_i u_i$ such that $\{u_i\}_{i=1}^{N_1+\cdots+N_n}$ are orthonormal in L^2. Since

$\mathcal{D}^2 = \mathcal{L}$, we have that $\mathcal{D}$ and $\mathcal{L}$ share the same eigenfunction u_i with eigenvalues λ_i and λ_i^2, respectively. Thus, we define

$$(r + c\mathcal{D} + d\mathcal{L})^{\nu/2} f := \sum_{k=1}^{n} \sum_{i=1}^{N_k} \left(r + c\lambda_i + d\lambda_i^2\right)^{\nu/2} \langle f, u_i \rangle_{L^2} u_i, \tag{82}$$

which is a linear operator on the direct sum space $\bigoplus_{k=1}^{n} C^k(X)$. We then define the reaction-diffusion kernel to be the solution to the system

$$(r + c\mathcal{D} + d\mathcal{L})^{\nu} k(c, \cdot) = c^\flat, \tag{83}$$

for any $c \in \bigoplus_{k=1}^{n} C^k(X)$. The solutions to (81) and (83) are given explicitly by

$$f(\cdot) = \sum_{k=1}^{n} \sum_{i=1}^{N_k} \left(r + c\lambda_i + d\lambda_i^2\right)^{-\nu/2} \langle \mathcal{W}, u_i \rangle_{L^2} u_i(\cdot), \tag{84}$$

$$\kappa(c, \cdot) = \sum_{k=1}^{n} \sum_{i=1}^{N_k} \left(r + c\lambda_i + d\lambda_i^2\right)^{-\nu} u_i(c) u_i(\cdot). \tag{85}$$

Then, following the proof in Appendix D.1.1 line-by-line, one can verify that κ is indeed a kernel for the GP f, that is, $\kappa(c, c') = \mathbb{E}[f(c)f(c')]$. Below, we present a more explicit proof under the numerical representation of (84) and (85), which can be written in the form

$$\boldsymbol{f} = \mathbf{U} \left(r\mathbf{I} + c\boldsymbol{\Lambda} + d\boldsymbol{\Lambda}^2\right)^{-\nu/2} \mathbf{U}^\top \boldsymbol{w} \tag{86}$$

$$\mathbf{K} = \mathbf{U} \left(r\mathbf{I} + c\boldsymbol{\Lambda} + d\boldsymbol{\Lambda}^2\right)^{-\nu} \mathbf{U}^\top. \tag{87}$$

Here, we denoted by $\boldsymbol{\Lambda} = \texttt{diag}(\lambda_1, \ldots, \lambda_{N_1 + \cdots + N_n})$ the diagonal matrix of eigenvalues, $\boldsymbol{w} \in \mathcal{N}(0, \mathbf{I})$ is the numerical representation of $\mathcal{W}$, and $\mathbf{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{N_1 + \cdots + N_n})$ is the matrix of eigenvectors.

D.2.1. Proof of Proposition 13

Proposition 31. *The kernel defined by* (87) *is related to* $\boldsymbol{f}$ *given by* (86)*, as*

$$[\mathbf{K}]_{ij} = \mathbb{E}[f_i f_j], \quad \forall i, j = 1, \ldots, N_1 + \cdots + N_n. \tag{88}$$

Proof. We have

$$\mathbb{E}[\boldsymbol{f}\boldsymbol{f}^\top] = \mathbf{U} \left(r\mathbf{I} - c\boldsymbol{\Lambda} + d\boldsymbol{\Lambda}^2\right)^{-\nu/2} \mathbf{U}^\top \underbrace{\mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\top]}_{=\mathbf{I}} \mathbf{U} \left(r\mathbf{I} - c\boldsymbol{\Lambda} + d\boldsymbol{\Lambda}^2\right)^{-\nu/2} \mathbf{U}^\top \tag{89}$$

$$= \mathbf{U} \left(r\mathbf{I} - c\boldsymbol{\Lambda} + d\boldsymbol{\Lambda}^2\right)^{-\nu/2} \underbrace{\mathbf{U}^\top \mathbf{U}}_{=\mathbf{I}} \left(r\mathbf{I} - c\boldsymbol{\Lambda} + d\boldsymbol{\Lambda}^2\right)^{-\nu/2} \mathbf{U}^\top \tag{90}$$

$$= \mathbf{U} \left(r\mathbf{I} - c\boldsymbol{\Lambda} + d\boldsymbol{\Lambda}^2\right)^{-\nu} \mathbf{U}^\top \tag{91}$$

$$= \mathbf{K}, \tag{92}$$

which proves the claim. □

D.3. Generalisation to Arbitrary Cell Weights

Here, we consider the case of general cell weights, extending the results in Appendix D.1 and D.2. In particular, we demonstrate how we arrive at identical expressions for the kernels, only the eigenbasis must be orthonormal with respect to the weighted L^2 inner product instead of the standard one.

D.3.1. MATÉRN KERNEL

Let $\{(\lambda_i^2, u_i)\}_{i=1}^{N_k}$ be the eigenpairs of the Hodge Laplacian operator Δ_k, defined with respect to the weighted L^2-inner product $\langle \cdot, \cdot \rangle_{L^2(\boldsymbol{w}^k)}$. We claim that in this case, the eigencochains $\{u_i\}_{i=1}^{N_k}$ can be set to be orthonormal under the weighted L^2-inner product. To see this, we first show that Δ_k is self-adjoint with respect to the weighted L^2-inner product:

$$\begin{aligned}
\langle f, \Delta_k g \rangle_{L^2(\boldsymbol{w}^k)} &= \langle f,\, d_{k-1} d_{k-1}^* g \rangle_{L^2(\boldsymbol{w}^k)} + \langle f,\, d_k^* d_k g \rangle_{L^2(\boldsymbol{w}^k)} && (93)\\
&= \langle d_{k-1}^* f,\, d_{k-1}^* g \rangle_{L^2(\boldsymbol{w}^{k-1})} + \langle d_k f,\, d_k g \rangle_{L^2(\boldsymbol{w}^{k+1})} && (94)\\
&= \langle d_{k-1} d_{k-1}^* f,\, g \rangle_{L^2(\boldsymbol{w}^k)} + \langle d_k^* d_k f,\, g \rangle_{L^2(\boldsymbol{w}^k)} && (95)\\
&= \langle \Delta_k f, g \rangle_{L^2(\boldsymbol{w}^k)}. && (96)
\end{aligned}$$

Now consider

$$\langle u_i, \Delta_k u_j \rangle_{L^2(\boldsymbol{w}^k)} = \lambda_j \langle u_i, u_j \rangle_{L^2(\boldsymbol{w}^k)}, \qquad \langle \Delta_k u_i, u_j \rangle_{L^2(\boldsymbol{w}^k)} = \lambda_i \langle u_i, u_j \rangle_{L^2(\boldsymbol{w}^k)}. \tag{97}$$

Since $\langle u_i, \Delta_k u_j \rangle_{L^2(\boldsymbol{w}^k)} = \langle \Delta_k u_i, u_j \rangle_{L^2(\boldsymbol{w}^k)}$ owing to the self-adjointness of Δ_k, we have $\lambda_j \langle u_i, u_j \rangle_{L^2(\boldsymbol{w}^k)} = \lambda_i \langle u_i, u_j \rangle_{L^2(\boldsymbol{w}^k)}$ by (97). For $\lambda_i \neq \lambda_j$, this relation is true if and only if $\langle u_i, u_j \rangle_{L^2(\boldsymbol{w}^k)} = 0$. In the case $\lambda_i = \lambda_j = \lambda$, letting $E(\lambda)$ denote the corresponding eigenspace, we can simply take the orthonormal basis of $E(\lambda)$ to be the elements of $\{u_i\}_{i=1}^{N_k}$ corresponding to the eigenvalue λ. Thus, we have a choice of $\{u_i\}_{i=1}^{N_k}$ such that $\langle u_i, u_j \rangle_{L^2(\boldsymbol{w}^k)} = \delta_{ij}$ for all i, j. Now, with this choice of the eigenbasis, we can follow the arguments in Appendix (D.1) almost identically to show that Proposition 12 still holds in the weighted setting, under the same definition for the Matérn GP (50) and the Matérn kernel (52). The only difference is that the $\flat$ operator and the Gaussian white noise $\mathcal{W}$ must take into consideration the cell weights, according to Definitions 23 and 29.

To illustrate this better, we consider its explicit representation in terms of a matrix-vector system. We first represent the weighted orthonormality condition of the eigenbasis by

$$\mathbf{U}^\top \mathbf{W} \mathbf{U} = \mathbf{I}, \tag{98}$$

where $\mathbf{U}$ is the matrix of eigenvectors of $\boldsymbol{\Delta}_k$ and $\mathbf{W} = \texttt{diag}(w_1^k, \ldots, w_{N_k}^k)$. Due to the orthonormality, we also have that for any $f \in C^k(X)$, we have the expression $f = \sum_i \langle f, u_i \rangle_{L^2(\boldsymbol{w}^k)} u_i$. This has the vector expression

$$\mathbf{U}(\mathbf{U}^\top \mathbf{W} \boldsymbol{f}) = \boldsymbol{f}, \tag{99}$$

which implies

$$\mathbf{U}\mathbf{U}^\top \mathbf{W} = \mathbf{I}, \tag{100}$$

since $\boldsymbol{f}$ is arbitrary. We can check that the operator (51) takes the form (contrast this with (68) in the non-weighted case):

$$\mathbf{L} := \mathbf{U} \left(\frac{2\nu}{\ell^2} \mathbf{I} + \boldsymbol{\Lambda}^2 \right)^{\nu/2} \mathbf{U}^\top \mathbf{W}. \tag{101}$$

Again, we can check that its inverse reads

$$\mathbf{L}^{-1} := \mathbf{U} \left(\frac{2\nu}{\ell^2} \mathbf{I} + \boldsymbol{\Lambda}^2 \right)^{-\nu/2} \mathbf{U}^\top \mathbf{W}, \tag{102}$$

by verifying

$$\begin{aligned}
\mathbf{L}^{-1}\mathbf{L} &= \mathbf{U} \left(\frac{2\nu}{\ell^2} \mathbf{I} + \boldsymbol{\Lambda}^2 \right)^{-\nu/2} \underbrace{\mathbf{U}^\top \mathbf{W} \mathbf{U}}_{=\mathbf{I}} \left(\frac{2\nu}{\ell^2} \mathbf{I} + \boldsymbol{\Lambda}^2 \right)^{\nu/2} \mathbf{U}^\top \mathbf{W} && (103)\\
&= \mathbf{U} \underbrace{\left(\frac{2\nu}{\ell^2} \mathbf{I} + \boldsymbol{\Lambda}^2 \right)^{-\nu/2} \left(\frac{2\nu}{\ell^2} \mathbf{I} + \boldsymbol{\Lambda}^2 \right)^{\nu/2}}_{=\mathbf{I}} \mathbf{U}^\top \mathbf{W} && (104)\\
&= \mathbf{U}\mathbf{U}^\top \mathbf{W} && (105)\\
&= \mathbf{I}, && (106)
\end{aligned}$$

and similarly, $\mathbf{L}\mathbf{L}^{-1} = \mathbf{I}$. Now, we consider the numerical representation $\boldsymbol{w}$ of the white noise process $\mathcal{W}$ in the weighted setting. We claim that

$$\boldsymbol{w} \sim \mathcal{N}(0, \mathbf{W}^{-1}). \tag{107}$$

This can be checked by using its definition (Definition 29), we have

$$\mathbb{E}[\langle \mathcal{W}, f\rangle_{L^2(\boldsymbol{w}^k)} \langle \mathcal{W}, g\rangle_{L^2(\boldsymbol{w}^k)}] = \langle f, g\rangle_{L^2(\boldsymbol{w}^k)} \tag{108}$$

$$\Leftrightarrow (\boldsymbol{f}^\top \mathbf{W})\mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\top](\mathbf{W}\boldsymbol{g}) = \boldsymbol{f}^\top \mathbf{W}\boldsymbol{g} \tag{109}$$

$$\Leftrightarrow \mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\top] = \mathbf{W}^{-1}. \tag{110}$$

Hence by (67), we have

$$\boldsymbol{f} = \mathbf{L}^{-1}\boldsymbol{w} \sim \mathcal{N}(0, \mathbf{K}), \tag{111}$$

where

$$\mathbf{K} = \mathbf{L}^{-1}\mathbf{W}^{-1}\mathbf{L}^{-\top} \tag{112}$$

$$= \mathbf{U}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \boldsymbol{\Lambda}^2\right)^{-\nu/2} \mathbf{U}^\top \underbrace{\mathbf{W}\mathbf{W}^{-1}\mathbf{W}}_{\mathbf{W}} \mathbf{U}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \boldsymbol{\Lambda}^2\right)^{-\nu/2} \mathbf{U}^\top \tag{113}$$

$$= \mathbf{U}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \boldsymbol{\Lambda}^2\right)^{-\nu/2} \underbrace{\mathbf{U}^\top\mathbf{W}\mathbf{U}}_{=\mathbf{I}} \left(\frac{2\nu}{\ell^2}\mathbf{I} + \boldsymbol{\Lambda}^2\right)^{-\nu/2} \mathbf{U}^\top \tag{114}$$

$$= \mathbf{U}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \boldsymbol{\Lambda}^2\right)^{-\nu} \mathbf{U}^\top. \tag{115}$$

The expression for the kernel is the same as in the unweighted case, except that now $\boldsymbol{\Lambda}, \mathbf{U}$ are the eigenpairs of the weighted Hodge Laplacian (45), the latter being orthonormal with respect to the weighted L^2-inner product instead of the standard one.

D.3.2. Reaction-diffusion Kernel

The same approach also applies to extending the reaction-diffusion kernel to the weighted setting. In this case, the matrix expression for the Dirac operator (79) reads

$$\boldsymbol{\mathcal{D}} = \begin{pmatrix} \mathbf{0} & \mathbf{D}_0^* & \cdots & \mathbf{0} \\ \mathbf{D}_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{D}_{n-1}^* \\ \mathbf{0} & \dots & \mathbf{D}_{n-1} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{W}_0^{-1}\mathbf{B}_1\mathbf{W}_1 & \cdots & \mathbf{0} \\ \mathbf{B}_1^\top & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{W}_{n-1}^{-1}\mathbf{B}_n\mathbf{W}_n \\ \mathbf{0} & \dots & \mathbf{B}_n^\top & \mathbf{0} \end{pmatrix}. \tag{116}$$

We can check that $\boldsymbol{\mathcal{D}}$ is self-adjoint under the weighted L^2-inner product, that is

$$\boldsymbol{f}^\top \mathbf{W}\boldsymbol{\mathcal{D}}\boldsymbol{g} = \boldsymbol{g}^\top \mathbf{W}\boldsymbol{\mathcal{D}}\boldsymbol{f}. \tag{117}$$

Hence, we can choose an eigenbasis $\mathbf{U}$ of $\boldsymbol{\mathcal{D}}$ that is orthonormal under the weighted L^2-inner product, with eigenvalues given by $\boldsymbol{\Lambda} = \texttt{diag}(\lambda_1, \dots, \lambda_{N_1+\dots+N_n})$. The operator (82) in this case can be expressed as

$$(r\mathbf{I} - c\boldsymbol{\mathcal{D}} + d\boldsymbol{\mathcal{L}})^{\nu/2}\boldsymbol{f} = \mathbf{U}\left(r\mathbf{I} - c\boldsymbol{\Lambda} + d\boldsymbol{\Lambda}^2\right)^{\nu/2}(\mathbf{U}^{-1}\mathbf{U})\mathbf{U}^\top\mathbf{W}\boldsymbol{f} \tag{118}$$

$$= \mathbf{U}\left(r\mathbf{I} - c\boldsymbol{\Lambda} + d\boldsymbol{\Lambda}^2\right)^{\nu/2}\mathbf{U}^\top\mathbf{W}\boldsymbol{f}. \tag{119}$$

Then as before, taking

$$\mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\top] = \mathbf{W}^{-1} \tag{120}$$

and following the remaining steps in Appendix D.3.1, we arrive at the expression for the weighted reaction-diffusion kernel

$$\mathbf{K} = \mathbf{U}\left(r\mathbf{I} - c\boldsymbol{\Lambda} + d\boldsymbol{\Lambda}^2\right)^{-\nu}\mathbf{U}^\top. \tag{121}$$

This is essentially the same as in the non-weighted case, except $\boldsymbol{\Lambda}, \mathbf{U}$ are now the eigenpairs of the weighted Dirac operator (116), the latter being orthonormal under the weighted L^2-inner product.

D.3.3. AMPLITUDE OF THE PROCESS

As a special case, given a positive constant $\sigma > 0$, let us consider the weights $w_\alpha^k = \sigma^{-2}$ for all k, α. That is,

$$\mathbf{W} = \sigma^{-2}\mathbf{I}. \tag{122}$$

In this case, notice that conditions (98) and (100) become

$$\mathbf{U}^\top\mathbf{U} = \mathbf{U}\mathbf{U}^\top = \sigma^2\mathbf{I}. \tag{123}$$

Thus, the normalised eigenbasis $\widehat{\mathbf{U}} := \sigma^{-1}\mathbf{U}$ is orthonormal under the standard L^2-inner product, i.e.,

$$\widehat{\mathbf{U}}^\top\widehat{\mathbf{U}} = \widehat{\mathbf{U}}\widehat{\mathbf{U}}^\top = \mathbf{I}. \tag{124}$$

Under this basis, the expressions for the Matérn and reaction-diffusion kernels read

$$\mathbf{K}_{\text{Matérn}} = \sigma^2\,\widehat{\mathbf{U}}\left(\frac{2\nu}{\ell^2}\mathbf{I} + \mathbf{\Lambda}^2\right)^{-\nu}\widehat{\mathbf{U}}^\top, \tag{125}$$

$$\mathbf{K}_{\text{r.d.}} = \sigma^2\,\widehat{\mathbf{U}}\left(r\mathbf{I} - c\mathbf{\Lambda} + d\mathbf{\Lambda}^2\right)^{-\nu}\widehat{\mathbf{U}}^\top. \tag{126}$$

The extra parameter σ controls the *amplitude* of the process

$$\mathrm{Var}(f_i) = [\mathbf{K}]_{ii} = \sigma^2 c_i, \tag{127}$$

for $c_i = [\widehat{\mathbf{U}}\Phi(\mathbf{\Lambda})\widehat{\mathbf{U}}^\top]_{ii}$, where $\Phi(\mathbf{\Lambda}) = \left(\frac{2\nu}{\ell^2}\mathbf{I} + \mathbf{\Lambda}^2\right)^{-\nu}$ in the case of the Matérn kernel and $\Phi(\mathbf{\Lambda}) = \left(r\mathbf{I} - c\mathbf{\Lambda} + d\mathbf{\Lambda}^2\right)^{-\nu}$ in the case of the reaction-diffusion kernel. This can be introduced as an extra hyperparameter in the model to fit the data more appropriately, which is recommended to obtain better results.

E. Experimental Details

In this paper, all Gaussian processes (graph Matérn GP, CC-GP and RD-GP) are implemented using the GPJax library (Pinder & Dodd, 2022). The objective function is the conjugate marginal log-likelihood and the optimiser is an implementation of Adam from Optax (Bradbury et al., 2018) with a learning rate set at 0.1.

E.1. Directed Edge Prediction

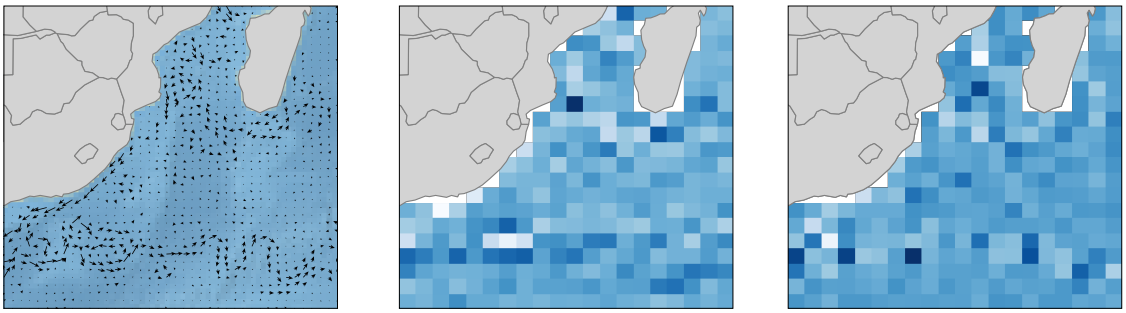
This experiment compares our CC-GP on edges (Matérn-CC kernel) and the graph Matérn kernel (Borovitskiy et al., 2020). The task is to predict the edge flow constructed from the geostrophic current around the southern tip of Africa. Here, the geostrophic current refers to the dominant component of the ocean current derived by balancing the pressure gradient with the Coriolis effect. The unprocessed data is retrieved from the (NOAA CoastWatch, 2023) database, which comes in the form of two scalar fields: one representing the x-component and the other the y-component of the geostrophic current vector field. The current around the southern tip of Africa is then extracted (lat = $[-45.0, -15.0]$, lon = $[20.0, 53.1]$) and its components are rescaled to a 2D grid of dimension 20×20 (see Figure 9).

The next step in the pre-processing is to transform this data into edge signals of a cubical 1-complex. We adopt the method in (Desbrun et al., 2006) to generate these signals. To do so, a cubical mesh of the same resolution as the data (20×20) is first generated, where each edge e in the mesh is assigned an orientation. Here, the orientation is represented by a unit vector $\hat{\boldsymbol{t}}_e$ pointing from one endpoint to the other. Then for each edge e, we compute how much of the geostrophic current flows along e in the direction specified by its orientation.

More precisely, the value f_0 on the edge e (illustrated in Figure 10) is computed according to

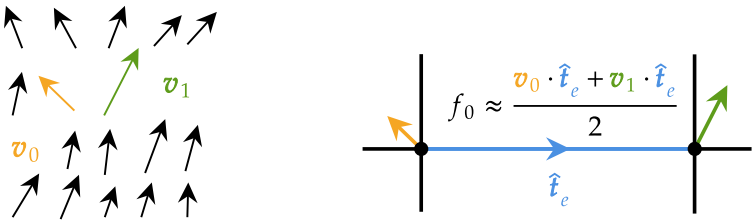
$$f_0 = \int_e \boldsymbol{v}(s)\cdot\hat{\boldsymbol{t}}_e\,\mathrm{d}s \approx \frac{\boldsymbol{v}_0\cdot\hat{\boldsymbol{t}}_e + \boldsymbol{v}_1\cdot\hat{\boldsymbol{t}}_e}{2}, \tag{128}$$

where $s : [0, 1] \to e$ is a parameterisation of the edge e and $\boldsymbol{v}$ is the geostrophic current. This yields directed edge signals on a cubical mesh, where we use the usual rule of setting the direction to be aligned with the orientation of e if f_0 is positive



(a) Geostrophic current (b) x-component (coarsened) (c) y-component (coarsened)

Figure 9. The geostrophic current around the southern tip of Africa. (Left) Quiver plot of the geostrophic current, (Middle) The x-component of the geostrophic current coarsened to a 20×20 grid, (Right) The y-component of the geostrophic current coarsened to a 20×20 grid.



(a) Two vectors of a vector field. (b) Directed edge signals

Figure 10. The construction of the edge signals.

and opposite to it if f_0 is negative. The training data is obtained by randomly selecting 30% of the generated edge signals and adding i.i.d. noise from a Gaussian $\mathcal{N}(0, 10^{-4})$.

For the training of the graph Matérn GP, the smoothness hyperparameter ν is fixed at 2. The amplitude and lengthscale hyperparameters σ^2, ℓ are both initialised at 1.0 and optimised for 1000 iterations using Adam. The training took less than 30 seconds on a MacBook Pro with M1 chip. In a similar way, when training CC-Matérn GP on edges, the smoothness hyperparameter ν is set to 2, and the amplitude and lengthscale hyperparameters σ^2, ℓ are initialised at 1.0, before optimising them for 1000 iterations using Adam. The training for this model also takes less than 30 seconds.

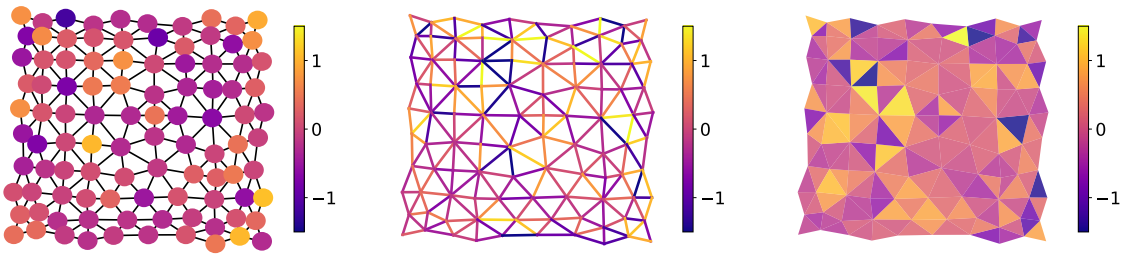
E.2. Signal Mixing

This experiment compares the performance of the RD-GP and the Matérn CC-GP in the task of predicting signals on the vertices, edges and triangles of a 2D simplicial mesh. The mesh is constructed by first defining a 10×10 grid, then subdividing this grid into triangles to transform it into a 2D simplicial mesh. The resulting complex is composed of 523 simplices: 100 vertices, 261 edges and 162 triangles.

The signals on the edges are created by taking inspiration from the Karhunen-Loève theorem, which states that a stochastic process can be expressed as a linear combination of L^2-orthogonal basis functions with random coefficients (one may view this as a stochastic analogue of the Fourier expansion). Here, the orthogonal basis functions are the set of eigenfunctions $\{u_i\}_i$ of the Hodge Laplacian Δ_1. The orthogonality of the eigenfunctions is ensured by the symmetry of the operator Δ_1. This forms a basis for edge signals (i.e. 1-cochains) that encodes the topology of the mesh through the information contained in Δ_1. For the coefficients in the basis expansion, we use i.i.d. Gaussians $\xi_i \sim \mathcal{N}(0, \lambda_i^{-1})$, where λ_i is the eigenvalue of Δ_1 corresponding to u_i. This expansion is truncated to lie between $0 < k < K$, which represent the minimal and maximal wavenumbers controlling the smoothness of the edge field. Putting this together yields the random 1-cochain

$$f = \sum_{i=k}^{K} \xi_i u_i, \quad \xi_i \sim \mathcal{N}(0, \lambda_i^{-1}). \tag{129}$$

Once the signals on the edges are obtained, the signals on the vertices and the triangles are computed by applying the coboundary operator d_1 and its adjoint d_1^* to f, respectively. Using the numerical representation of cochains, $\{\boldsymbol{u}_i\}_i$ becomes the set of eigenvectors of the Hodge Laplacian matrix $\mathbf{\Delta}_1$, the coboundary operator becomes the matrix $\mathbf{B}_1$ and its adjoint becomes $\mathbf{B}_2^\top$ (see Appendix B). The vertex signals and the triangle signals are obtained by computing $\mathbf{B}_1\boldsymbol{f}$ and $\mathbf{B}_2^\top\boldsymbol{f}$, respectively. An example signal for $k = 20$ and $K = 100$ is displayed in Figure 11, which we use as the ground truth in our experiment.



(a) Vertex signals (b) Edge signals (c) Triangle signals

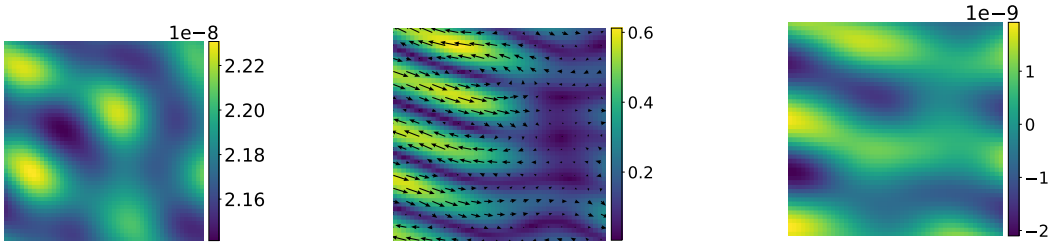
Figure 11. An example synthetic signal on the vertices, edges and triangles.

The training data is generated by randomly selecting a third of the vertices, a third of the edges and a third of the triangles from this ground truth field, and corrupting them by i.i.d. noise from a Gaussian $\mathcal{N}(0, 10^{-2})$.

For training the CC-Matérn GP, the smoothness hyperparameter ν is fixed at 2, and the amplitude and lengthscale hyperparameters σ^2, ℓ are both initialised at 1.5, before optimising them for 1000 iterations using Adam. The training takes less than a minute on a MacBook Pro equipped with a M1 Pro chip. The training of RD-GP is similar: The smoothness hyperparameter ν is fixed at 2, and the amplitude hyperparameter σ^2, the reaction coefficient r, the diffusion coefficient d, and the cross-diffusion coefficient c are all initialised at 1.5. They are then optimised for 1000 iterations using Adam, again taking less than a minute to run.

E.3. Modelling Electromagnetism

In this experiment, we compare the performance of the RD-GP and the Matérn GP on imputing signals on the vertices, edges and faces of a 20×20 square lattice. The signals come from simulations of electromagnetic fields. In particular, we used the Python package `PyCharge` to generate 2D electromagnetic fields on a square domain, generated by 10 oscillating point charges at randomly generated locations. The fields that were computed were the scalar potential (V), electric field (E) and the magnetic field (B). Physically, these are a scalar field, a vector field and a two-form / pseudovector field (i.e., a field of vectors whose sign depends on the orientation of the manifold), respectively. An example of such fields is displayed in Figure 12.

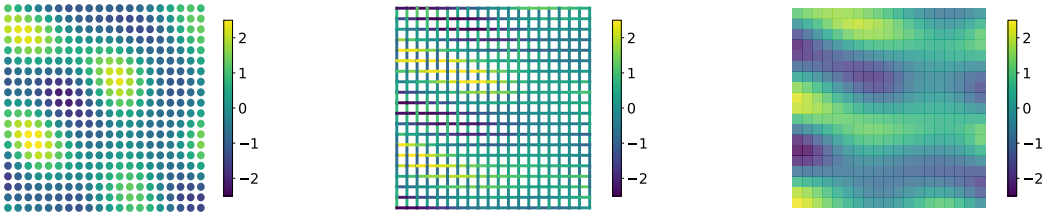


(a) Scalar potential (b) Electric field (c) Magnetic flux into page

Figure 12. We plot an example scalar potential, electric field and magnetic field generated from ten randomly sampled oscillating point charges. For the electric field, we display the amplitudes of the vectors in colour in the background. For the magnetic field, we plot the magnetic flux going into the page, which becomes a scalar field.

The next step involves projecting these fields onto a cellular complex of dimension two, given by a 20×20 square lattice. Projecting the scalar potential on the vertices of a square lattice involves just extracting the point values of the field at

the vertex locations. To project the electric field onto the edges of the lattice, the procedure is similar to that described in Appendix E.1. Finally, projecting the magnetic field onto the square faces of the lattice involves averaging the magnetic flux (i.e., $B \cdot \hat{n}$) over the square cells, where the unit normal $\hat{n}$ is given by the normal vector determining the orientation of the cell (see Appendix A). We also normalise the projected values, due to the large discrepancies of magnitudes between the different fields. The final projections of the fields in Figure 12 onto the cells of a square lattice are displayed in Figure 13.



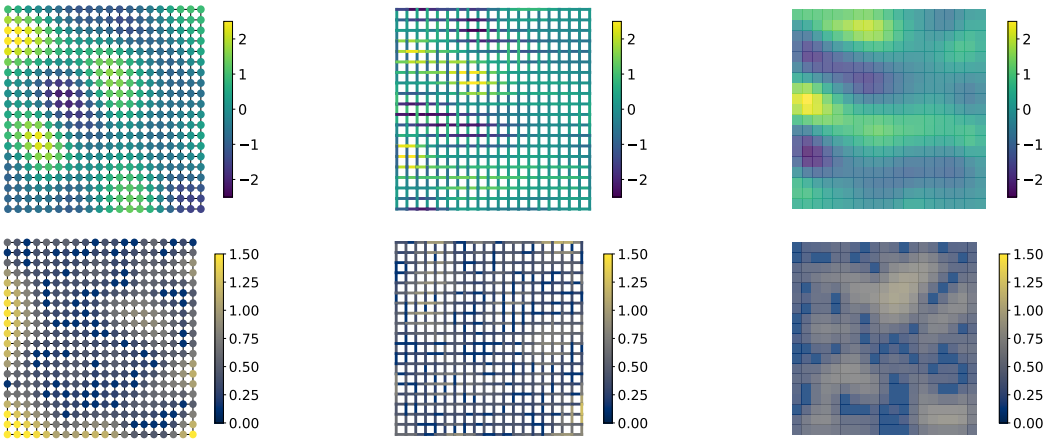
(a) Scalar potential (b) Electric field (c) Magnetic field

Figure 13. Discrete representations of the scalar potential, electric field and magnetic field as 0, 1 and 2-cochains of a square lattice respectively. Each square cell is assigned clockwise orientation.

The training data is generated by randomly selecting a sixth of the vertices, edges and square faces from the projected fields and adding i.i.d Gaussian noise with standard deviation of 10^{-2}.

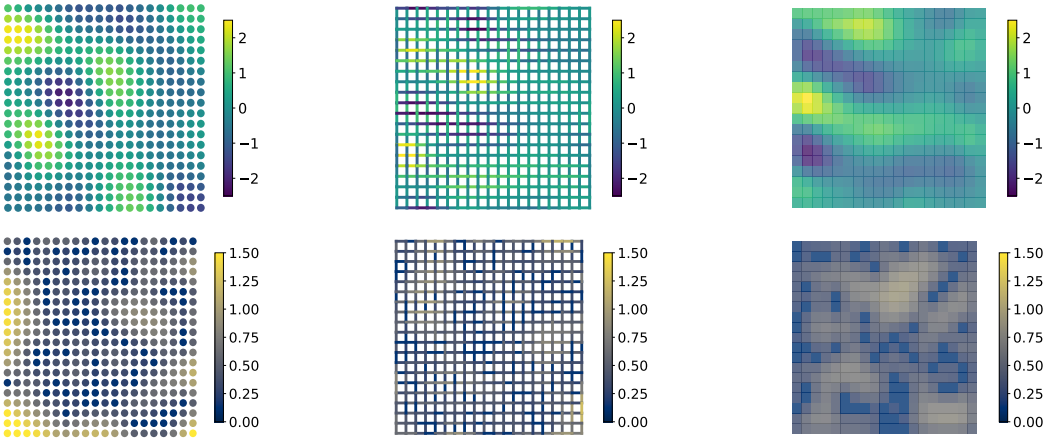
For training the CC-Matérn GP, the smoothness hyperparameter ν is fixed at 2, and the amplitude and lengthscale hyperparameters σ^2, ℓ are both initialised at 1.5, before optimising them for 1000 iterations using Adam. The training takes less than a minute on a MacBook Pro equipped with a M1 Pro chip. The training of RD-GP is similar: The smoothness hyperparameter ν is fixed at 2, the amplitude hyperparameter σ^2, the reaction coefficient r and the diffusion coefficient d are initialised at 1.5. The cross-diffusion coefficient c is initialised at 2.5. They are then optimised for 1000 iterations using Adam, again taking less than a minute to run.

The predictions made by the RD-GP is displayed in Figure 14 and those made by the CC-Matérn GP is displayed in Figure 15. We see that both GPs recover the ground truth field (Figure 13) fairly accurately from the observations. While the metrics indicate that the RD-GP output is slightly better than those of CC-Matérn (Table 5), perceptually, the differences are too small to see.



(a) Scalar potential (b) Electric field (c) Magnetic field

Figure 14. Predictions of the scalar potential, electric field and magnetic field made from the reaction-diffusion GP. The top row displays the predictive mean and the bottom row displays the standard deviations.



(a) Scalar potential (b) Electric field (c) Magnetic field

Figure 15. Predictions of the scalar potential, electric field and magnetic field made from the CC-Matérn GP. The top row displays the predictive mean and the bottom row displays the standard deviations.

MSE (↓)	CC-Matérn	Reaction-diffusion
Scalar potential	0.113	**0.110**
Electric field	0.125	**0.108**
Magnetic field	0.151	**0.137**
NLL (↓)		
Scalar potential	70.8	**68.5**
Electric field	110.3	**92.3**
Magnetic field	118.4	**110.1**

Table 5. Mean square error (MSE) and negative log-likelihood (NLL) of predictions of the electromagnetic fields in Figure 14 (RD-GP) and Figure 15 (CC-Matérn GP). The performance of the reaction-diffusion GP is slightly better than the Matérn GP on the cellular complex, suggesting that mixing on this example has some positive impact on the predictions.