Hodge-Compositional Edge Gaussian Processes

Maosheng Yang
TU Delft, Netherlands

Viacheslav Borovitskiy
ETH Zürich, Switzerland

Elvin Isufi
TU Delft, Netherlands

Abstract

We propose principled Gaussian processes (GPs) for modeling functions defined over the edge set of a simplicial 2-complex, a structure similar to a graph in which edges may form triangular faces. This approach is intended for learning flow-type data on networks where edge flows can be characterized by the discrete divergence and curl. Drawing upon the Hodge decomposition, we first develop classes of divergence-free and curl-free edge GPs, suitable for various applications. We then combine them to create *Hodge-compositional edge GPs* that are expressive enough to represent any edge function. These GPs facilitate direct and independent learning for the different Hodge components of edge functions, enabling us to capture their relevance during hyperparameter optimization. To highlight their practical potential, we apply them for flow data inference in currency exchange, ocean currents and water supply networks, comparing them to alternative models.

1 INTRODUCTION

Gaussian processes (GPs) are a widely used class of statistical models capable of quantifying uncertainty associated to their own predictions (Rasmussen & Williams, 2006). These models are determined by covariance kernels which encode prior knowledge about the unknown function. Choosing an appropriate kernel is often challenging, particularly when the input space is non-Euclidean (Duvenaud, 2014).

Developing GPs on graphs has been a subject of recent work, which requires structured kernels to encode the dependence between nodes (Venkitaraman et al., 2020;

Zhi et al., 2023), like the diffusion (Smola & Kondor, 2003) or random walk kernels (Vishwanathan et al., 2010). More recently, Borovitskiy et al. (2021) derived the more general family of Matérn kernels on graphs from stochastic partial differential equations (SPDEs) thereon, mirroring the continuous approaches on manifolds (Borovitskiy et al., 2020; Azangulov et al., 2022, 2023). Nikitin et al. (2022) incorporated the temporal factor in this framework to build temporal-graph kernels. However, GPs in these works are targeted for modeling functions on the nodes of networks.

We instead focus on functions defined on the *edges*, of particular interest for modeling edge-based dynamical processes in many complex networks, such as flows of energy, signal or mass (Schaub et al., 2014). For example, in water supply networks, we typically monitor the flow rates within pipes (edges) connecting tanks (nodes) (Zhou et al., 2022). Other examples include energy flows in power grids (Jia et al., 2019), synaptic signals between neurons in brain networks (Faskowitz et al., 2022), and exchange rates on trading paths (edges) of currencies (nodes) (Jiang et al., 2011).

While it might seem intuitive to use node-based methods for edge-based tasks using line-graphs (Godsil & Royle, 2001), this often yields sub-optimal solutions (Jia et al., 2019). Alternatively, recent successes in signal processing and neural networks for edge data have emerged from modeling flows on the edge set of a simplicial 2-complex (SC$_2$), including (Jia et al., 2019; Barbarossa & Sardellitti, 2020; Schaub et al., 2021; Yang et al., 2022a,b; Roddenberry et al., 2021; Yang & Isufi, 2023), among others. A SC$_2$ can be viewed as a graph with the additional set of triangular faces, encoding how edges are adjacent to each other via nodes or faces. A SC$_2$ also allows to characterize key properties of edge flows using discrete concepts of *divergence* (div) and *curl* (Lovász, 2004; Lim, 2020), measuring how they diverge at nodes and circulate along faces. For example, electric currents in circuit networks respecting the Kirchhoff's law are div-free (Grady & Polimeni, 2010), and arbitrage-free exchange rates are

Correspondence to: m.yang-2@tudelft.nl. Code available at: https://github.com/cookbook-ms/Hodge-Edge-GP.

curl-free along loops of trading paths (Jiang et al., 2011). Moreover, edge functions on a SC_2 admit the *Hodge decomposition* into three parts: gradient, curl and harmonic components, being either curl-free, div-free or both div- and curl-free (Lim, 2020). This provides unique insights in various applications including ranking (Jiang et al., 2011), gaming theory (Candogan et al., 2011), brain networks (Vijay Anand et al., 2022) and finance (Fujiwara & Islam, 2020). Nevertheless, existing works on edge-based learning remain deterministic and there is a lack of principled ways to define GP priors on the edge set of SCs, which is the central goal of this work.

Our main contribution lies in the proposal of *Hodge-compositional edge GPs*. We build them as combinations of three GPs, each modeling a specific part of the Hodge decomposition of an edge function, namely the gradient, curl and harmonic parts. With a focus on the Matérn GP family, we show that each of them can be linked to a SPDE, extending the framework used by Borovitskiy et al. (2020, 2021, 2023). Compared to a direct extension of graph GPs, they enable separate learning of the different Hodge components, which allows us to capture the practical behavior of edge flows. We also demonstrate their practical potential in edge-based learning tasks in foreign currency exchange markets, ocean current analysis and water supply networks.

2 BACKGROUND

A random function $f : X \to \mathbb{R}$ defined over a set X is a Gaussian process $f \sim \mathcal{GP}(\mu, k)$ with mean function $\mu(\cdot)$ and kernel $k(\cdot, \cdot)$ if, for any finite set of points $\boldsymbol{x} = (x_1, \ldots, x_n)^\top \in X^n$, the random vector $f(\boldsymbol{x}) = (f(x_1), \ldots, f(x_n))^\top$ is multivariate Gaussian with mean vector $\mu(\boldsymbol{x})$ and covariance matrix $k(\boldsymbol{x}, \boldsymbol{x})$.

The kernel k of a *prior* GP encodes prior knowledge about the unknown function while its mean μ is usually assumed to be zero. GP regression combines such a *prior* with training data $x_1, y_1, \ldots, x_n, y_n$ where $x_i \in X$, $y_i \in \mathbb{R}$ with $y_i = f(x_i) + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$. This results in a posterior $f_{|\boldsymbol{y}}$ which is another GP: $f_{|\boldsymbol{y}} \sim \mathcal{GP}(\mu_{|\boldsymbol{y}}, k_{|\boldsymbol{y}})$. For any new input $x^* \in X$, the mean $\mu_{|\boldsymbol{y}}(x^*)$ is the prediction and the posterior variance $k_{|\boldsymbol{y}}(x^*, x^*)$ quantifies the uncertainty. We refer the reader to Rasmussen & Williams (2006) for more details. Defining an appropriate kernel is one of the main challenges in GP modeling (Duvenaud, 2014).

2.1 GPs on Graphs

Let $G = (V, E)$ be an unweighted graph where $V = \{1, \ldots, N_0\}$ is the set of nodes and E is the set of N_1 edges such that if nodes i, j are connected, then $e = (i, j) \in E$. We can define real-valued functions on its node set $f_0 : V \to \mathbb{R}$, collected into a vector $\boldsymbol{f}_0 = (f_0(1), \ldots, f_0(N_0))^\top \in \mathbb{R}^{N_0}$. Denote the *oriented* node-to-edge incidence matrix by $\boldsymbol{B}_1$ of dimension $N_0 \times N_1$. It has entries $[\boldsymbol{B}_1]_{ie} = -1$ and $[\boldsymbol{B}_1]_{je} = 1$, if an edge $e = \{i, j\}$ exists, and zero otherwise. One can view oriented graphs as undirected graphs having an additional orientation structure on the edge set (Godsil & Royle, 2001, Sec 8.3). They are different from directed graphs, as discussed next subsection. The *graph Laplacian* is then given by $\boldsymbol{L}_0 = \boldsymbol{B}_1 \boldsymbol{B}_1^\top$, which is a positive semi-definite linear operator on the space $\mathbb{R}^{N_0}$ of node functions. It admits an eigendecomposition $\boldsymbol{L}_0 = \boldsymbol{U}_0 \boldsymbol{\Lambda}_0 \boldsymbol{U}_0^\top$ where $\boldsymbol{\Lambda}_0$ collects its eigenvalues on the diagonal and $\boldsymbol{U}_0$ collects the orthogonal eigenvectors of $\boldsymbol{L}_0$ (Chung, 1997).

A GP on graphs $\boldsymbol{f}_0 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_0)$ assumes $\boldsymbol{f}_0$ is a random function with zero mean and a graph kernel $\boldsymbol{K}_0$ which encodes the covariance between pairs of nodes. To construct principled graph GPs, Borovitskiy et al. (2021) extended the idea of deriving continuous GPs from SPDEs (Whittle, 1963; Lindgren et al., 2011) to the domain of graphs. Specifically, given the following SPDE on graphs with a Gaussian noise $\boldsymbol{w}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

$$\Phi(\boldsymbol{L}_0)\boldsymbol{f}_0 = \boldsymbol{w}_0, \text{ with } \Phi(\boldsymbol{L}_0) = \left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}_0\right)^{\frac{\nu}{2}}, \quad (1)$$

where $\Phi(\boldsymbol{L}_0) = \boldsymbol{U}_0 \Phi(\boldsymbol{\Lambda}_0) \boldsymbol{U}_0^\top$ and $\Phi(\cdot)$ applies to $\boldsymbol{\Lambda}_0$ element-wise, its solution is the Matérn graph GP

$$\boldsymbol{f}_0 \sim \mathcal{GP}\left(\boldsymbol{0}, \left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}_0\right)^{-\nu}\right) \quad (2)$$

with positive parameters κ, ν. When scaled properly, the Matérn kernel gives the graph diffusion kernel for $\nu \to \infty$, which in turn relates to the random walk kernel by Kondor & Lafferty (2002). This SPDE framework can be extended to spatial-temporal data yielding respective graph kernels (Nikitin et al., 2022).

2.2 Edge Functions on Simplicial Complexes

Simplicial 2-complexes represent discrete geometry more expressively than graphs. They are triples $SC_2 = (V, E, T)$ where V, E are the sets of nodes and edges, same as for graphs, and T is the set of triangular faces (shortened as triangles) such that if $(i, j), (j, k), (i, k)$ form a *closed* triangle, then $t = (i, j, k) \in T$ (Munkres, 2018). Note that *not* all three pairwise connected edges are necessarily closed and included in T. An example of a SC_2 is shown in Fig. 1a where the set $\{1, 3, 4\}$ is an open triangle and thus is not in T.

For each edge and each triangle, we assume the increasing order of their node labels as their reference

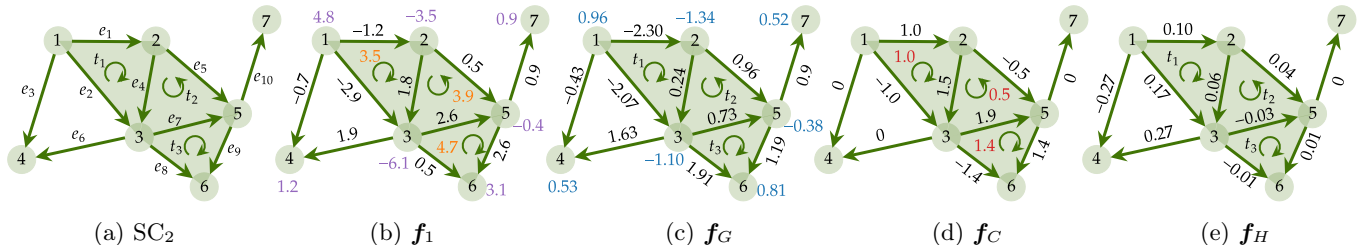Maosheng Yang, Viacheslav Borovitskiy, Elvin Isufi

Figure 1: (a) A SC_2 where we shade (closed) triangles in green and denote reference orientations of edges/triangles by arrows. (b) An edge function $\boldsymbol{f}_1$ with its divergence (purple values on nodes) and curl (orange values in triangles). (c-e) Hodge decomposition: (c) gradient flow $\boldsymbol{f}_G = \boldsymbol{B}_1^\top \boldsymbol{f}_0$, obtained as the gradient of some node function $\boldsymbol{f}_0$ (given in blue). It is curl-free: the net-circulation along each triangle is zero; (d) curl flow $\boldsymbol{f}_C = \boldsymbol{B}_2 \boldsymbol{f}_2$, induced by some circulating triangle signal $\boldsymbol{f}_2$ (given in red). It is div-free: the net-flow at each node is zero; and (e) harmonic flow $\boldsymbol{f}_H$, circulating around the 1-dimensional "hole" (open triangle $\{1, 3, 4\}$), where the net-flow on nodes and net-circulation in triangles are zero. All numbers are rounded to two decimal places.

orientation[1]. An oriented edge, denoted as $e = [i, j]$, is an ordering of $\{i, j\}$. This is not a directed edge allowing flow only from i to j, but rather an assignment of the sign of the flow: from i to j it is positive and the reverse is negative. Same goes for oriented triangles denoted as $t = [i, j, k]$ and we have $[i, j, k] = [j, k, i] = [k, i, j] = -[i, k, j] = -[k, j, i] = -[j, i, k]$.

In a SC_2, the functions, $f_1 : E \to \mathbb{R}$, on its edges E are required to be *alternating* (Lim, 2020), meaning that, we have $f_1(\bar{e}) = -f_1(e)$ if $\bar{e} = [j, i]$ is oriented opposite to the reference $e = [i, j]$. For example, in Fig. 1b, $f_1(1, 2) = -1.2$ means there is a 1.2 unit of flow from 2 to 1. This property keeps the flow unchanged with respect to the edge orientation. We collect the edge functions on E into $\boldsymbol{f}_1 = (f_1(e_1), \ldots, f_1(e_{N_1}))^\top \in \mathbb{R}^{N_1}$, as in Fig. 1b, which we also call as an *edge flow*.

We can also define alternating functions on triangles in T where $f_2(\bar{t}) = -f_2(t)$ if $\bar{t}$ is an odd permutation of reference $t = [i, j, k]$ (Lim, 2020). We collect them in $\boldsymbol{f}_2 \in \mathbb{R}^{N_2}$ where $N_2 = |T|$. In topology, functions f_0, f_1, f_2 are called *0-, 1-, 2-cochains*, which are discrete analogs of differential forms on manifolds (Grady & Polimeni, 2010). This motivates the use of subscripts $0, 1, 2$. Here we can view these functions as vectors of data on nodes, edges and triangles.

2.3 Hodge Laplacians

In the similar spirit as $\boldsymbol{L}_0$ operating on node functions, we can define the discrete *Hodge Laplacian* operating on the space $\mathbb{R}^{N_1}$ of edge functions

$$\boldsymbol{L}_1 = \boldsymbol{B}_1^\top \boldsymbol{B}_1 + \boldsymbol{B}_2 \boldsymbol{B}_2^\top := \boldsymbol{L}_d + \boldsymbol{L}_u \qquad (3)$$

where $\boldsymbol{B}_2$ is the edge-to-triangle incidence matrix. Its entries are $[\boldsymbol{B}_2]_{et} = 1$, for $e = [i, j]$ or $e = [j, k]$, and $[\boldsymbol{B}_2]_{et} = -1$ for $e = [i, k]$, if a triangle $t = [i, j, k]$ exists, and zero otherwise. Matrix $\boldsymbol{L}_1$ describes the connectivity of edges where the *down* part $\boldsymbol{L}_d$ and the *up* part $\boldsymbol{L}_u$ encode how edges are adjacent, respectively, through nodes and via triangles. For example, e_3 and e_6 are down neighbors sharing node 4 in Fig. 1a and e_1 and e_2 are up neighbors, collocated in t_1. Matrix $\boldsymbol{L}_1$ is positive semi-definite, admitting an eigendecomposition $\boldsymbol{L}_1 = \boldsymbol{U}_1 \boldsymbol{\Lambda}_1 \boldsymbol{U}_1^\top$ where diagonal matrix $\boldsymbol{\Lambda}_1 = \text{diag}(\lambda_1, \ldots, \lambda_{N_1})$ collects the eigenvalues and $\boldsymbol{U}_1$ is the eigenvector matrix. Likewise, one can define $\boldsymbol{L}_2 = \boldsymbol{B}_2^\top \boldsymbol{B}_2$ encoding the adjacency between triangles. Our discussion henceforth considers the unweighted $\boldsymbol{L}_1$ but it also holds for the weighted variants in Grady & Polimeni (2010); Schaub et al. (2020).

3 EDGE GAUSSIAN PROCESSES

We now define GPs on edges of a SC_2, specifically, $\boldsymbol{f}_1 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_1)$ with zero mean and edge kernel $\boldsymbol{K}_1$. Throughout this work, we refer to them as *edge GPs*, and call graph GPs in Section 2.1 as *node GPs* because they are both multivariate Gaussian but the former is indexed by $X = E$ and the latter by $X = V$. We start with deriving edge GPs from SPDEs on edges as a natural extension of Eq. (1). Then, by introducing basic notions from discrete calculus (Grady & Polimeni, 2010) and the Hodge decomposition theorem, we propose the divergence-free and curl-free GPs, combining them into Hodge-compositional GPs.

3.1 Edge GPs from SPDEs on Edges

The derivation of graph GPs in Eq. (2) as solutions of graph SPDE in Eq. (1) motivates the following SPDEs

---

[1]The orientation of a general simplex is an equivalence class of permutations of its labels—two orientations are equivalent (respectively, opposite) if they differ by an even (respectively, odd) permutation (Lim, 2020, Sec 4).

on edges, with edge Gaussian noise $\boldsymbol{w}_1 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$,

$$\Phi(\boldsymbol{L}_1)\boldsymbol{f}_1 = \boldsymbol{w}_1 \qquad (4)$$

where $\Phi(\boldsymbol{L}_1) = \boldsymbol{U}_1 \Phi(\boldsymbol{\Lambda}_1)\boldsymbol{U}_1^\top$ is a differential operator defined through $\boldsymbol{L}_1$. When we consider the operators

$$\Phi(\boldsymbol{L}_1) = \left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}_1\right)^{\frac{\nu}{2}} \text{ and } \Phi(\boldsymbol{L}_1) = e^{\frac{\kappa^2}{4}\boldsymbol{L}_1}, \qquad (5)$$

the solutions to Eq. (4) give two edge GPs

$$\begin{aligned}
\boldsymbol{f}_{1,\text{Matérn}} &\sim \mathcal{GP}\left(\boldsymbol{0}, \left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}_1\right)^{-\nu}\right), \\
\boldsymbol{f}_{1,\text{diffusion}} &\sim \mathcal{GP}\left(\boldsymbol{0}, e^{-\frac{\kappa^2}{2}\boldsymbol{L}_1}\right),
\end{aligned} \qquad (6)$$

which are the *edge Matérn* and *diffusion* GPs, respectively. These edge GPs impose structured prior covariance that encodes the dependence between edges. A related *Hodge Laplacian kernel* $(\boldsymbol{L}_1^\top \boldsymbol{L}_1)^\dagger$ can be obtained by setting $\Phi(\boldsymbol{L}_1) = \boldsymbol{L}_1$, i.e., $\boldsymbol{L}_1 \boldsymbol{f}_1 = \boldsymbol{w}_1$. This kernel was used to penalize the smoothness of edge functions in Schaub et al. (2021). The kernels of Eq. (6) are more flexible though and allow encoding non-local edge-to-edge adjacency while $\boldsymbol{L}_1$ instead encodes the local direct (one-hop) adjacency.

3.2 Div-free and Curl-free Edge GPs

The edge GPs in Section 3.1 define distributions over all edge functions. As opposed to this, here we seek to define GPs on the classes of divergence-free and curl-free edge functions. We start with defining the appropriate notions of discrete derivatives, expressed in terms of the incidence matrices.

Discrete Derivatives The *gradient* is a linear operator from the space of node functions to that of edge functions. At edge $e = [i, j]$, it is defined as

$$(\operatorname{grad} f_0)(e) = (\boldsymbol{B}_1^\top \boldsymbol{f}_0)_e = f_0(j) - f_0(i), \qquad (7)$$

which computes the difference between the values of a function on adjacent nodes, resulting in a flow on the connecting edge. We call $\boldsymbol{f}_G = \boldsymbol{B}_1^\top \boldsymbol{f}_0$ a gradient flow and $\boldsymbol{f}_0$ a node potential, as shown in Fig. 1c.

The *divergence*, the adjoint of gradient, is a linear operator from the space of edge functions to that of node functions. At node i, it is defined as

$$(\operatorname{div} f_1)(i) = (\boldsymbol{B}_1 \boldsymbol{f}_1)_i = -\sum_{j \in N(i)} f_1(i, j) \qquad (8)$$

with $N(i)$ the neighbors of i. Physically, it computes the net-flow of edge functions passing through node i, i.e., the in-flow minus the out-flow, as shown in Fig. 1b. A *divergence-free* flow has a zero net-flow everywhere.

Lastly, the *curl* operator is a linear operator from the space of edge functions to that of triangle functions. At triangle $t = [i, j, k]$, it is defined as

$$(\operatorname{curl} f_1)(t) = (\boldsymbol{B}_2^\top \boldsymbol{f}_1)_t = f_1(i, j) + f_1(j, k) - f_1(i, k) \quad (9)$$

which computes the *net-circulation* of edge functions along the edges of t, as a rotational measure of $\boldsymbol{f}_1$, as shown in Fig. 1b. A *curl-free* flow has zero curl over each triangle. As in calculus, we have the identity $\operatorname{curl} \operatorname{grad} = \boldsymbol{B}_2^\top \boldsymbol{B}_1^\top = \boldsymbol{0}$, i.e., gradient flow is curl-free.

Analogous to their continuous vector field counterparts, div-free and curl-free edge functions are ubiquitous, e.g., the electric currents and the exchange rates later in Section 4.1. We refer to Grady & Polimeni (2010); Lim (2020) for more examples. From this perspective, we can view the graph Laplacian as $\boldsymbol{L}_0 = \operatorname{div} \operatorname{grad} = \boldsymbol{B}_1 \boldsymbol{B}_1^\top$, which is a graph-theoretic analog of the Laplace-Beltrami operator Δ_0 on manifolds. Also, the SPDE on graphs in Eq. (1) is a discrete counterpart of the continuous one for scalar functions on manifolds. Moreover, the Hodge Laplacian $\boldsymbol{L}_1$ can be viewed as $\boldsymbol{L}_1 = \operatorname{grad} \operatorname{div} + \operatorname{curl}^* \operatorname{curl} = \boldsymbol{B}_1^\top \boldsymbol{B}_1 + \boldsymbol{B}_2 \boldsymbol{B}_2^\top$, which is a discrete analog of the vector Laplacian (or Helmholtzian) Δ_1 for vector fields.

Hodge Decomposition The following *Hodge decomposition theorem*, unfolding an edge function, will allow us to improve the edge GPs in Eq. (6).

Theorem 1 (Hodge (1989)). *The space $\mathbb{R}^{N_1}$ of edge functions is a direct sum of three subspaces*

$$\mathbb{R}^{N_1} = \operatorname{im}(\boldsymbol{B}_1^\top) \oplus \ker(\boldsymbol{L}_1) \oplus \operatorname{im}(\boldsymbol{B}_2), \qquad (10)$$

where $\operatorname{im}(\boldsymbol{B}_1^\top)$ is the gradient space, $\ker(\boldsymbol{L}_1)$ the harmonic space and $\operatorname{im}(\boldsymbol{B}_2)$ the curl space.

It states that any edge function $\boldsymbol{f}_1$ is composed of three orthogonal parts: gradient, curl, harmonic functions

$$\boldsymbol{f}_1 = \boldsymbol{f}_G + \boldsymbol{f}_H + \boldsymbol{f}_C \qquad (11)$$

where $\boldsymbol{f}_G = \boldsymbol{B}_1^\top \boldsymbol{f}_0$, being curl-free, is the gradient of some node function $\boldsymbol{f}_0$, and $\boldsymbol{f}_C = \boldsymbol{B}_2 \boldsymbol{f}_2$, being div-free, is the curl-adjoint of some triangle function $\boldsymbol{f}_2$. Lastly, $\boldsymbol{f}_H$ is harmonic (both div- and curl-free, $\boldsymbol{L}_1 \boldsymbol{f}_H = \boldsymbol{0}$). This decomposition is illustrated in Fig. 1. It provides a crucial tool for understanding edge functions and has been used in many applications as we discussed above.

Furthermore, the eigenspace $\boldsymbol{U}_1$ of $\boldsymbol{L}_1$ can be reorganized in terms of the three Hodge subspaces as

$$\boldsymbol{U}_1 = [\boldsymbol{U}_H \ \boldsymbol{U}_G \ \boldsymbol{U}_C] \qquad (12)$$

where $\boldsymbol{U}_H$ is the eigenvector matrix associated to zero eigenvalues $\boldsymbol{\Lambda}_H = \boldsymbol{0}$ of $\boldsymbol{L}_1$, $\boldsymbol{U}_G$ is associated to the

nonzero eigenvalues $\mathbf{\Lambda}_G$ of $\boldsymbol{L}_\mathrm{d}$, and $\boldsymbol{U}_C$ is associated to the nonzero eigenvalues $\mathbf{\Lambda}_C$ of $\boldsymbol{L}_\mathrm{u}$. Moreover, they span the Hodge subspaces:

$$\begin{aligned} \mathrm{span}(\boldsymbol{U}_H) = \ker(\boldsymbol{L}_1), \ \mathrm{span}(\boldsymbol{U}_G) = \mathrm{im}(\boldsymbol{B}_1^\top), \\ \mathrm{span}(\boldsymbol{U}_C) = \mathrm{im}(\boldsymbol{B}_2), \end{aligned} \quad (13)$$

where $\mathrm{span}(\bullet)$ denotes all possible linear combinations of columns of $\bullet$ (Yang et al., 2022b).

Div-free, Curl-free Edge GPs Given the eigendecomposition in Eq. (12), we can obtain special classes of edge GPs by only using a certain type of eigenvectors when building edge kernels of Eq. (6). Specifically, we define *gradient* and *curl edge GPs* as follows

$$\boldsymbol{f}_G \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_G), \quad \boldsymbol{f}_C \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_C) \quad (14)$$

where the gradient kernel and the curl kernel are

$$\boldsymbol{K}_G = \boldsymbol{U}_G \Psi_G(\mathbf{\Lambda}_G) \boldsymbol{U}_G^\top, \ \boldsymbol{K}_C = \boldsymbol{U}_C \Psi_C(\mathbf{\Lambda}_C) \boldsymbol{U}_C^\top. \quad (15)$$

We also define the *harmonic GPs* $\boldsymbol{f}_H \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_H)$ with the harmonic kernel $\boldsymbol{K}_H = \boldsymbol{U}_H \Psi_H(\mathbf{\Lambda}_H) \boldsymbol{U}_H^\top$.

Proposition 2. *Let $\boldsymbol{f}_G$ and $\boldsymbol{f}_C$ be the gradient and curl Gaussian processes, respectively. Then,* $\mathrm{curl}\, \boldsymbol{f}_G = \boldsymbol{0}$ *and* $\mathrm{div}\, \boldsymbol{f}_C = \boldsymbol{0}$ *with probability one. Moreover, a harmonic Gaussian process $\boldsymbol{f}_H$ follows* $\mathrm{curl}\, \boldsymbol{f}_H = \boldsymbol{0}$ *and* $\mathrm{div}\, \boldsymbol{f}_H = \boldsymbol{0}$ *with probability one.*

See proof in Appendix B.2. These Hodge GPs provide more targeted priors for special edge functions which are either div- or curl-free, capable of capturing these key properties. In the case of Matérn kernels, we set

$$\Psi_\square(\mathbf{\Lambda}_\square) = \sigma_\square^2 \Big(\frac{2\nu_\square}{\kappa_\square^2} \boldsymbol{I} + \mathbf{\Lambda}_\square \Big)^{-\nu_\square}, \quad (16)$$

for $\square \in \{H, G, C\}$, where $\sigma_\square^2$ controls the variance we assign to the function in the subspace, and $\nu_\square, \kappa_\square$ are the regular Matérn parameters. Note that since $\mathbf{\Lambda}_H = \boldsymbol{0}$, we consider a scaling function for $\boldsymbol{K}_H$ as $\Psi_H(\boldsymbol{0}) = \sigma_H^2$. We illustrate such a Matérn kernel function in Fig. 2 (left). These Hodge GPs can be derived from SPDEs on edges as well.

Proposition 3. *Given a scaled curl white noise $\boldsymbol{w}_C \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{W}_C)$ where $\boldsymbol{W}_C = \sigma_C^2 \boldsymbol{U}_C \boldsymbol{U}_C^\top$, consider the following SPDE on edges:*

$$\Phi_C(\boldsymbol{L}_\mathrm{u}) \boldsymbol{f}_C = \boldsymbol{w}_C, \quad (17)$$

with differential operators

$$\Phi_C(\boldsymbol{L}_\mathrm{u}) = \Big(\frac{2\nu_C}{\kappa_C^2} \boldsymbol{I} + \boldsymbol{L}_\mathrm{u} \Big)^{\frac{\nu_C}{2}}, \ \Phi_C(\boldsymbol{L}_\mathrm{u}) = e^{\frac{\kappa_C^2}{4} \boldsymbol{L}_\mathrm{u}}. \quad (18)$$

The respective solutions give the curl edge GPs with Matérn kernel in Eq. (16) and diffusion kernel

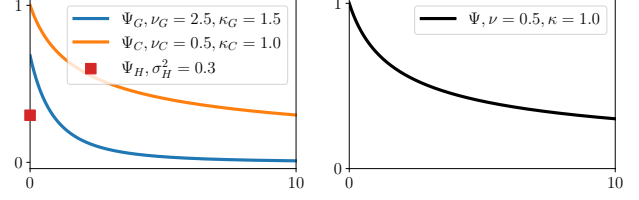$$\Psi_C(\mathbf{\Lambda}_C) = \sigma_C^2 e^{-\frac{\kappa_C^2}{2} \mathbf{\Lambda}_C}. \quad (19)$$

Figure 2: *(Left)* Matérn kernel functions $\Psi_\square(\lambda)$ for $\square = \{H, G, C\}$ in Eq. (16) of gradient, curl and harmonic GPs in the eigen-spectrum λ ranging in the min and man eigenvalues of $\boldsymbol{L}_1$. *(Right)* Matérn kernel function $\Psi(\lambda)$ of non-HC GP in Eq. (6).

Likewise, we can derive the gradient Matérn and diffusion GPs from the SPDEs as Eq. (17) but with operators $\Phi_G(\boldsymbol{L}_\mathrm{d})$ and a scaled gradient white noise.

See proof in Appendix B.3. We can draw the intuition of SPDE in Eq. (17) from the continuous analogy. In the case of $\boldsymbol{L}_\mathrm{u} \boldsymbol{f}_C = \boldsymbol{w}_C$, the equation $\mathrm{curl}^* \mathrm{curl}\, f_1(\boldsymbol{x}) = w_1(\boldsymbol{x})$ is a stochastic vector Laplace's equation of a div-free (solenoidal) vector field, where $w_1(\boldsymbol{x})$ the curl adjoint of some vector potential. In physics, this describes the static magnetic field from a magnetic vector potential, as well as an incompressible fluid.

3.3 Hodge-compositional Edge GPs

Many edge functions of interest are indeed div- or curl -free, but not all. In this section we combine the gradient, curl and harmonic GPs to define the Hodge-compositional (HC) edge GPs.

Definition 4. A Hodge-compositional edge Gaussian process $\boldsymbol{f}_1 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_1)$ is a sum of gradient, curl and harmonic GPs, i.e., $\boldsymbol{f}_1 = \boldsymbol{f}_G + \boldsymbol{f}_C + \boldsymbol{f}_H$ where

$$\boldsymbol{f}_\square \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_\square) \text{ with } \boldsymbol{K}_\square = \boldsymbol{U}_\square \Psi_\square(\mathbf{\Lambda}_\square) \boldsymbol{U}_\square^\top \quad (20)$$

for $\square = H, G, C$ where their kernels do not share hyperparameters.

Given this definition, we can obtain the following properties of HC edge GPs.

Lemma 5. *Let $\boldsymbol{f}_1 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_1)$ be an edge GP in Definition 4. Its realizations then give all possible edge functions[2]. It further holds that $\boldsymbol{K}_1 = \boldsymbol{K}_H + \boldsymbol{K}_G + \boldsymbol{K}_C$, and the three Hodge GPs are mutually independent, i.e.,* $Cov(\boldsymbol{f}_G, \boldsymbol{f}_C) = Cov(\boldsymbol{f}_G, \boldsymbol{f}_H) = Cov(\boldsymbol{f}_C, \boldsymbol{f}_H) = \boldsymbol{0}$.

See proof in Appendix B.4. Naturally, we can construct a Matérn HC GP as the sum of Matérn GPs

---

[2]Note that HC edge GPs do not represent all possible edge GPs. They are a particular GP family satisfying the independence hypothesis on the three Hodge GPs. This, however, does not contradict with that their realizations can represent all edge functions.

in the three subspaces with their kernels given by Eq. (16), and likewise for the diffusion HC GP by Eq. (19). Compared to the GPs in Eq. (6), referred to as non-HC GPs henceforth, HC GPs are more flexible and expressive, having more degrees of freedom. We discuss their practical advantages below.

Inductive GP Prior The HC GP encodes the prior covariance $\text{Cov}(f_1(e), f_1(e'))$ between edge functions over two edges e, e' as follows: (i) the covariance is the sum of three covariances $\text{Cov}_\square = \text{Cov}(f_\square(e), f_\square(e'))$ for $\square = H, G, C$; (ii) each $\text{Cov}_\square$ encodes the covariance between the corresponding Hodge parts of f_1 without affecting the others; and (iii) no covariance is imposed across different Hodge components, e.g., $\text{Cov}(f_G(e), f_C(e')) = 0$.

In the spatial/edge domain, this is related to separating the down and up adjacencies encoded in the SPDE operators $\Phi(\cdot)$. From an eigen-spectrum perspective, the eigenvalues $\Psi_\square$ of HC GP's kernels associated to the three Hodge subspaces have individual parameters. This enables capturing the different Hodge components of edge functions, as well as their relevance during hyperparameter optimization, further allowing us to directly recover the Hodge components in predictions, which we detail in Appendix B.5. Non-HC GPs instead require solving the Hodge decomposition in Eq. (11) (least squares problems) (Lim, 2020). Another implication is that, unlike for non-HC GPs, we do not require specific knowledge about the div or curl of the underlying function.

Comparison to non-HC GPs When we view non-HC GPs in terms of the Hodge decomposition, we notice that they put priors on the three Hodge GPs in a way that shares hyperparameters. This enforces learning the same hyperparameters for different Hodge components, resulting in a single function covering the entire edge spectrum, as shown in Fig. 2 (right), as opposed to the three individual functions of the HC one.

This raises issues when separate learning, say, different lengthscales, is required for the gradient and curl components. Non-HC GPs are strictly incapable of this practical need when an eigenvalue is associated to both gradient and curl spaces. We also delve into this in terms of *edge Fourier features* in Appendix B.6 where we compare the priors induced on the edge Fourier coefficients by HC and non-HC GPs.

Connection to Diffusion on Edges The HC diffusion kernel, given by $K_1 = \exp(-(\frac{\kappa_G^2}{2} L_\text{d} + \frac{\kappa_C^2}{2} L_\text{u}))$, when $\sigma_\square^2$s are one, is the Green's function for the edge diffusion of a function $\phi : [0, \infty) \times E \to \mathbb{R}$

$$\frac{\text{d}\phi(t)}{\text{d}t} = -(\mu L_\text{d} + \gamma L_\text{u})\phi(t), \text{ where } \mu, \gamma > 0 \quad (21)$$

with $\phi|_{t=\tau} = e^{-(\mu\tau L_\text{d} + \gamma\tau L_\text{u})}\phi(0)$. This equation describes the diffusion process on the edge space of SC_2 that was used for network analysis (Muhammad & Egerstedt, 2006; DeVille, 2021), often arising as the limit of random walks on edges (Schaub et al., 2020). The covariance K_1 within this context encodes the proportion of edge flow traveling from edge e to e' via down and up edge adjacencies. Its vector field counterpart was used for shape analysis (Zobel et al., 2011; Sharp et al., 2019). Compared to the graph (node) diffusion converging $(t \to \infty)$ to the state that is constant on all nodes as long as the graph is connected, the harmonic state of the edge diffusion can be non-constant, but lies in the span of U_H. We refer to Appendix B.7 for visualizations of the two harmonic states.

Complexity The kernels of HC edge GPs can be constructed in a scalable way by considering the l largest eigenvalues with off-the-shelf eigen-solvers, e.g., Lanczos algorithm. We refer to Appendix B.8 for more details on the complexity of implementing HC GPs, as well as sampling from them.

3.4 Node-Edge-Triangle GP Interactions

The gradient and curl components of edge functions are (co)derivatives of some node and triangle functions, specifically, $f_G = B_1^\top f_0$ and $f_C = B_2 f_2$ as in Eq. (11). Since the derivative of a GP is also a GP, we can then construct a gradient GP from node GPs.

Corollary 6. *Suppose a node function f_0 is a GP $f_0 \sim \mathcal{GP}(0, K_0)$ with $K_0 = \Psi_0(L_0) = U_0\Psi_0(\Lambda_0)U_0^\top$. Then, its gradient is an edge GP $f_G \sim \mathcal{GP}(0, K_G)$ where $K_G = B_1^\top K_0 B_1 = U_G\Psi_G(\Lambda_G)U_G^\top$ with*

$$\Psi_G(\Lambda_G) = \Lambda_G\Psi_0(\Lambda_G). \quad (22)$$

The proof follows from (i) derivatives preserving Gaussianity, and (ii) L_0 and L_d having the same nonzero eigenvalues. We can also obtain a curl edge GP from a GP on triangles likewise. In turn, for an edge GP, its div is a node GP and its curl is a GP on triangles. We refer to Appendix B.9 for the proof and more details.

Exploiting this interaction between GPs on nodes, edges and triangles can lead to new useful GPs, especially when functions on nodes, edges and triangles are intrinsically related by physical laws. For example, in water networks, water flowrates in pipes are often related to the gradient of hydraulic heads on nodes, as we will show in Section 4.3. This implies that given an appropriate node GP, say, node Matérn GP in Eq. (2), a good edge GP prior can be imposed as its gradient, as in Corollary 6. Furthermore, by leveraging this interaction, we can construct HC edge GPs as follows.

Maosheng Yang, Viacheslav Borovitskiy, Elvin Isufi

(a) Ground truth   (b) Mean, HC Matérn   (c) Standard deviation   (d) Mean, Non-HC   (e) Learned kernels
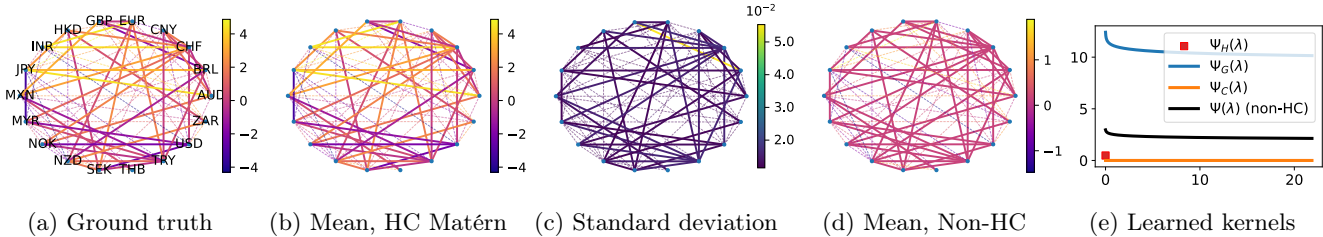
Figure 3: (a-d) Interpolating a smaller forex market (for better visibility) with train ratio 50% where dashed (solid) edges are used for training (testing). (e) Learned Matérn kernels in the spectrum of the Hodge Laplacian, $\Psi(\lambda)$ for non-HC GP and $\Psi_\square(\lambda)$ with $\square = \{H, G, C\}$ for HC GPs.

Proposition 7. *Let $\boldsymbol{f}_1$ be an edge function defined in Eq. (11) with harmonic component $\boldsymbol{f}_H$, node function $\boldsymbol{f}_0$ and triangle function $\boldsymbol{f}_2$. If we model $\boldsymbol{f}_0$ as a GP on nodes $\boldsymbol{f}_0 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_0)$, model $\boldsymbol{f}_2$ as a GP on triangles $\boldsymbol{f}_2 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_2)$, and $\boldsymbol{f}_H$ as a harmonic GP $\boldsymbol{f}_H \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_H)$, then we have GP $\boldsymbol{f}_1 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_1)$ with*

$$\boldsymbol{K}_1 = \boldsymbol{K}_H + \boldsymbol{B}_1^\top \boldsymbol{K}_0 \boldsymbol{B}_1 + \boldsymbol{B}_2 \boldsymbol{K}_2 \boldsymbol{B}_2^\top. \quad (23)$$

See proof in Appendix B.10. This alternative HC GP incorporates the Hodge theorem prior in a way that directly relates the node potential and the triangle function. It can be applicable when GP priors of node or triangle functions are more discernible. Similar ideas for general cellular complexes are studied in the concurrent paper by Alain et al. (2023).

Continuous Counterparts Edge functions can be viewed as discrete analogs of vector fields. Berlinghieri et al. (2023) studied the models similar to our HC edge GPs in Eq. (23) for Euclidean vector fields and the concurrent work by Robert-Nicoud et al. (2024) studies similar models for vector fields on manifolds.

4 EXPERIMENTS

We apply HC GPs for edge-based inference tasks in three applications: foreign currency exchange (forex), ocean current and water supply networks (WSNs). We showcase the structured prior on edges in these tasks by comparing them to baselines: (i) Euclidean GPs with RBF and Matérn kernels, and (ii) Node GPs on the line-graph—built by exchanging the nodes with edges in the original graph (Godsil & Royle, 2001). To highlight the prior of the Hodge decomposition, we also compare with non-HC GPs. For each of them, we consider Matérn and diffusion kernels. We perform GP regression with Gaussian likelihood for model fitting using the `GPyTorch` framework (Gardner et al., 2018). We use the root mean squared error (RMSE) to evaluate the predictive mean and the negative log predictive density (NLPD) for prediction uncertainty. We refer to Appendix C for full experimental details.

Table 1: Forex rates inference results.

Method	RMSE		NLPD	
	Diffusion	Matérn	Diffusion	Matérn
Euclidean	2.17 ± 0.13	2.19 ± 0.12	2.12 ± 0.07	2.20 ± 0.18
Line-Graph	2.43 ± 0.07	2.46 ± 0.07	2.28 ± 0.04	2.32 ± 0.03
Non-HC	2.48 ± 0.07	2.47 ± 0.08	2.36 ± 0.07	2.34 ± 0.04
HC	0.08 ± 0.12	0.06 ± 0.12	-3.52 ± 0.02	-3.52 ± 0.02

4.1 Foreign Currency Exchange

A forex market can be modeled as a network where nodes represent currencies and edges the exchangeable pairs (Jiang et al., 2011). Forex rates in a fair market ideally satisfy the *arbitrage-free* condition: for any currencies i, j, k, we have $r^{i/j} r^{j/k} = r^{i/k}$ with $r^{i/j}$ the rate between i and j. That is, the exchange path $i \rightarrow j \rightarrow k$ provides no gain or loss over a direct path $i \rightarrow k$. If we model forex rates as edge flows $f_1(i, j) = \log(r^{i/j})$, this condition can be translated into that $\boldsymbol{f}_1$ is a gradient flow, being curl-free, i.e., $f_1(i, j) + f_1(j, k) - f_1(i, k) = 0$. Here we consider real-world forex data on 2018/10/05 with 25 most traded currencies forming 210 exchangeable pairs and 710 triangles, formed by any three pairwise exchangeable currencies (Oanda, 2018; Jia et al., 2019). We randomly sample 20% of edges for training and test on the rest.

From Table 1, we see that HC GPs achieve significantly lower RMSEs with high certainty (small NLPDs), as visualized in Fig. 3. This shows their ability to automatically capture the curl-free nature of the forex rates. As shown in Fig. 3e, the HC Matérn GP learns that harmonic and curl components should vanish. In contrast, the other three give poor predictions, due to: (i) Euclidean GPs being oblivious of the structure of edge functions; (ii) line-graph GPs imposing inappropriate structure through node priors; and (iii) non-HC GPs being unable to induce the curl-free prior without removing the gradient. This results from sharing parameters in their kernels for different Hodge components. As shown in Fig. 3e, the non-HC Matérn GP

(a) Original ocean current

(b) Posterior mean

(c) Standard deviation

(d) Curl-free component

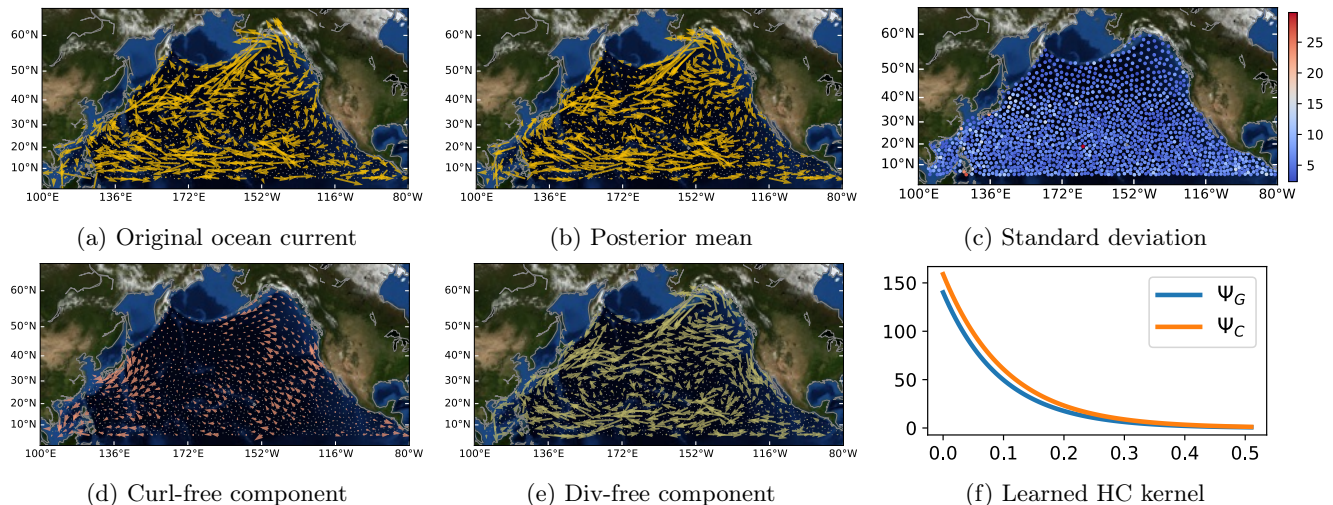(e) Div-free component

(f) Learned HC kernel

Figure 4: (a-b) Ground truth and interpolated ocean current in the vector field domain. (c) Standard deviation approximated by sampling 50 edge flows from the predictive posterior distribution and converted to the vector field domain. (d-e) The curl-free and div-free components directly obtained from the learned kernels. (f) Learned diffusion kernels $\Psi_G(\lambda)$ and $\Psi_C(\lambda)$ of the HC GP in the spectrum of the Hodge Laplacian.

Table 2: Ocean current inference results.

Method	RMSE		NLPD	
	Diffusion	Matérn	Diffusion	Matérn
Euclidean	1.00 ± 0.01	1.00 ± 0.00	1.42 ± 0.01	1.42 ± 0.10
Line-Graph	0.99 ± 0.00	0.99 ± 0.00	1.41 ± 0.00	1.41 ± 0.00
Non-HC	0.35 ± 0.00	0.35 ± 0.00	0.33 ± 0.00	0.36 ± 0.03
HC	0.34 ± 0.00	0.35 ± 0.00	0.33 ± 0.01	0.37 ± 0.04

learns a nonzero kernel in the whole spectrum, unable to remove the non-arbitrage-free part.

4.2 Ocean Current Analysis

We consider the edge-based ocean current learning following the setup in Chen et al. (2021). The ocean current velocity fields were converted using the linear integration approximation to edge flows within a SC_2 whose nodes are 1500 buoys sampled from North Pacific ocean drifter records in 2010-2019 (Lumpkin & Centurioni, 2019). We apply both non-HC and HC GPs to predict the converted edge flows. Given the large number of edges ($\sim$20k), we consider a truncated approximation of kernels with eigenpairs associated with the 500 largest eigenvalues (Knyazev, 2001). We randomly sample 20% of edges for training and test on the rest.

From Table 2, we see that HC and non-HC GPs exhibit similar performance. This arises from the comparable behavior of the gradient and curl components, as depicted in Fig. 4f, where the learned gradient and curl diffusion kernels display close patterns. In contrast, Euclidean and line-graph GPs give poor predictions

emphasizing the importance of structured edge priors.

We further convert the predicted edge flows into the vector field domain, as shown in Fig. 4b, based on Chen et al. (2021). We see that the predictions capture the pattern of the original velocity field. We approximate the predicted velocity field uncertainty by computing the average ℓ_2 distance per location from 50 posterior samples to the mean in the vector field domain. As shown in Fig. 4c, the standard deviation estimated this way is small in most locations except for a few exceptions (small islands at the bottom left) where the original field is more discontinuous. Moreover, since HC GPs enable the direct recovery of gradient and curl components, we show their corresponding vector fields in Figs. 4d and 4e, giving better insights into how ocean currents behave, of particular interest in oceanography. For example, we can observe the well-known North Pacific gyres including the North Equatorial, Kuroshio and Alaska currents in Fig. 4e.

4.3 Water Supply Networks

Network-based methods have been used in WSNs where tanks or reservoirs are represented by nodes, and pipes by edges (Zhou et al., 2022). By modeling the hydraulic heads as node functions $\boldsymbol{f}_0$ and the water flowrates as edge functions $\boldsymbol{f}_1$, the commonly used empirical equation connecting the two reads as $\boldsymbol{B}_1^\top \boldsymbol{f}_0 = \bar{\boldsymbol{f}}_1 := \text{diag}(\boldsymbol{r}) \boldsymbol{f}_1^{1.852}$ where r_e is the resistance of pipe e and the exponentiation is applied element-wise (Dini & Tabesh, 2014).

We consider the Zhi Jiang WSN with 114 tanks (in-

(a) Ground truth

(b) Mean, HC Matérn

(c) Mean, non-HC Matérn

(d) Std, HC Matérn

(e) Std, non-HC Matérn

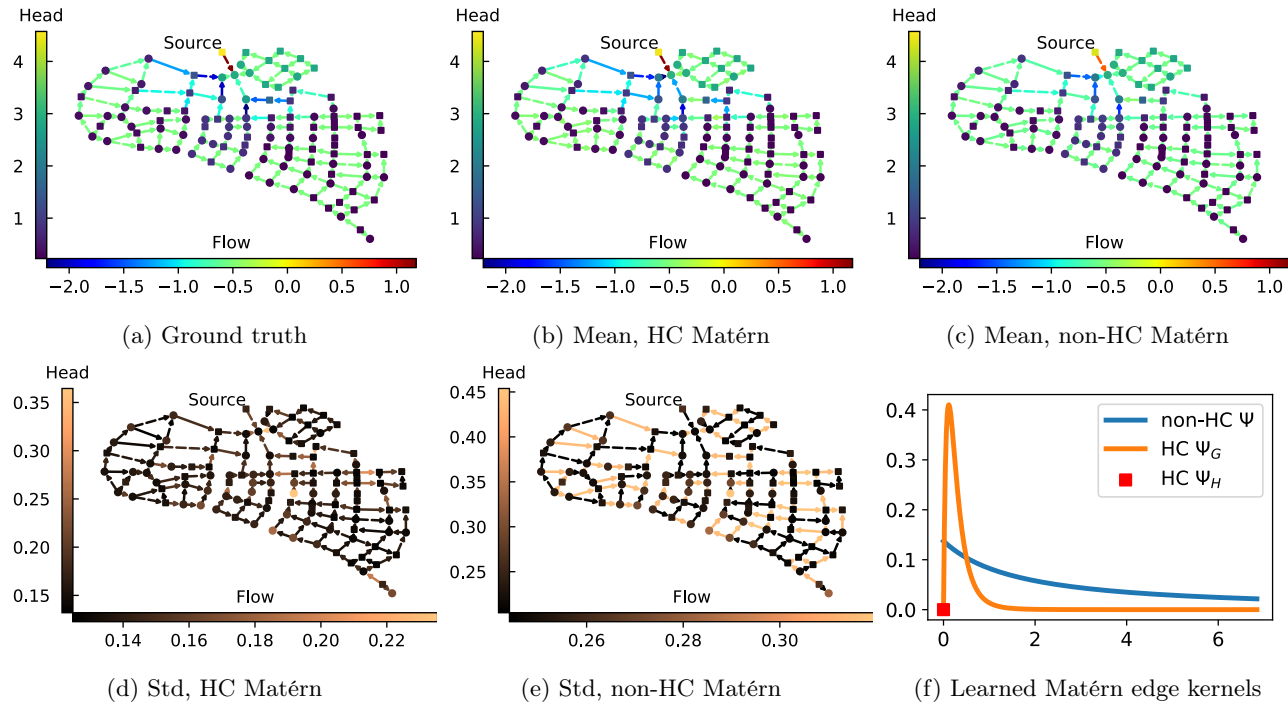(f) Learned Matérn edge kernels

Figure 5: (a-e) Posterior mean and standard deviation (std) based on the Matérn node GPs, and the HC and non-HC Matérn edge GPs. Squared (Circled) nodes represent the node samples for training (testing). Dashed (solid) edges denote the edge samples for training (testing). (f) The learned edge GP kernels in the spectrum, $\Psi(\lambda)$ for non-HC GP and $\Psi_H(0), \Psi(\lambda)$ for HC GPs.

Table 3: WSN inference results.

Method	Node Heads		Edge Flowrates	
	RMSE	NLPD	RMSE	NLPD
Diffusion, non-HC	0.16 ± 0.05	0.72 ± 2.06	0.32 ± 0.05	0.97 ± 1.80
Matérn, non-HC	0.16 ± 0.04	0.71 ± 2.39	0.26 ± 0.05	0.10 ± 0.13
Diffusion, HC	0.15 ± 0.04	-0.47 ± 0.14	0.22 ± 0.03	-0.20 ± 0.13
Matérn, HC	0.15 ± 0.04	-0.25 ± 0.48	0.23 ± 0.03	-0.45 ± 0.49

cluding one source) and 164 pipes (without triangles, Dandy (2016)) and simulate a scenario based on Klise et al. (2017). We perform joint state estimation of heads $\boldsymbol{f}_0$ and the adjusted flowrates $\bar{\boldsymbol{f}}_1$, by modeling them as GPs on nodes and edges, respectively. To compare HC and non-HC edge GPs, for a node GP with kernel $\boldsymbol{K}_0$, we consider the HC GP as its gradient, as discussed in Corollary 6. For the non-HC one, we consider a kernel $\boldsymbol{K}_1$ of the same type as $\boldsymbol{K}_0$. We randomly sample 50% of nodes and edges for training and test on the rest.

From Table 3, we see that while the mean predictions of heads remain similar whether we use HC or non-HC edge GPs, the former perform better for edge flows, particularly in the pipes around the source, as shown in Figs. 5b and 5c. Moreover, HC GPs have better prediction uncertainty with smaller average NLPDs for both heads and flowrates, as illustrated in Figs. 5d and 5e.

This is because HC GPs that we use share parameters with node GPs, helping to calibrate the uncertainty of head predictions. They also capture the physical prior of the pipe equation that assumes flowrates are a gradient flow. As shown in Fig. 5f, the HC Matérn GP learns a kernel with a trivial harmonic prior and a nonzero gradient prior in small eigenvalues, reflecting the gradient nature of the pipe flowrates. Note that due to the randomness of training samples, the WSN, having small edge connectivity, may become disconnected, causing the significant variance in NLPDs.

5 CONCLUSION

We introduced Hodge-compositional (HC) Gaussian processes (GPs) for modeling functions on the edges of simplicial 2-complexes. These HC GPs are constructed by combining three individual GPs, each designed to capture the gradient, curl and harmonic components of the Hodge decomposition of edge functions. This allows them to learn each component separately, making them more expressive and interpretable when compared to alternatives. Moreover, they can be extended to jointly model signals on nodes, edges and faces of the domain. We demonstrated their practical potential in learning real-world flow data.

References

Alain, M., Takao, S., Paige, B., and Deisenroth, M. P. Gaussian Processes on Cellular Complexes. *arXiv preprint arXiv:2311.01198*, 2023. Cited on page 7.

Azangulov, I., Smolensky, A., Terenin, A., and Borovitskiy, V. Stationary Kernels and Gaussian Processes on Lie Groups and their Homogeneous Spaces I: the Compact Case. *arXiv preprint arXiv:2208.14960*, 2022. Cited on page 1.

Azangulov, I., Smolensky, A., Terenin, A., and Borovitskiy, V. Stationary Kernels and Gaussian Processes on Lie Groups and their Homogeneous Spaces II: non-compact symmetric spaces. *arXiv preprint arXiv:2301.13088*, 2023. Cited on page 1.

Barbarossa, S. and Sardellitti, S. Topological Signal Processing Over Simplicial Complexes. *IEEE Transactions on Signal Processing*, 68:2992–3007, 2020. ISSN 1941-0476. doi: 10.1109/TSP.2020.2981920. Cited on pages 1 and 18.

Berlinghieri, R., Trippe, B. L., Burt, D. R., Giordano, R., Srinivasan, K., Özgökmen, T., Xia, J., and Broderick, T. Gaussian Processes at the Helm(holtz): A More Fluid Model for Ocean Currents. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023. Cited on page 7.

Borovitskiy, V., Terenin, A., Mostowsky, P., et al. Matérn Gaussian processes on Riemannian manifolds. *Advances in Neural Information Processing Systems*, 33:12426–12437, 2020. Cited on pages 1 and 2.

Borovitskiy, V., Azangulov, I., Terenin, A., Mostowsky, P., Deisenroth, M., and Durrande, N. Matérn Gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*, pp. 2593–2601. PMLR, 2021. Cited on pages 1, 2, 19, and 24.

Borovitskiy, V., Karimi, M. R., Somnath, V. R., and Krause, A. Isotropic Gaussian Processes on Finite Spaces of Graphs, February 2023. Cited on page 2.

Candogan, O., Menache, I., Ozdaglar, A., and Parrilo, P. A. Flows and Decompositions of Games: Harmonic and Potential Games. *Mathematics of Operations Research*, 36(3):474–503, August 2011. ISSN 0364-765X, 1526-5471. doi: 10.1287/moor.1110.0500. Cited on page 2.

Chen, Y.-C. and Meila, M. The decomposition of the higher-order homology embedding constructed from the *k*-Laplacian. *Advances in Neural Information Processing Systems*, 34:15695–15709, 2021. Cited on page 22.

Chen, Y.-C., Meilă, M., and Kevrekidis, I. G. Helmholtzian Eigenmap: Topological feature discovery & edge flow learning from point cloud data. *arXiv preprint arXiv:2103.07626*, 2021. Cited on pages 8 and 23.

Chung, F. R. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997. Cited on page 2.

Dandy, G. 06 Zhi Jiang. International Systems., 2016. https://uknowledge.uky.edu/wdst_international/6. Cited on pages 9 and 24.

DeVille, L. Consensus on Simplicial Complexes: Results on Stability and Synchronization. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31 (2):023137, February 2021. ISSN 1054-1500, 1089-7682. doi: 10.1063/5.0037433. Cited on page 6.

Dini, M. and Tabesh, M. A new method for simultaneous calibration of demand pattern and Hazen-Williams coefficients in water distribution systems. *Water resources management*, 28:2021–2034, 2014. Cited on pages 8 and 24.

Duvenaud, D. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014. Cited on pages 1, 2, and 17.

Erb, W. Krylov subspace methods to accelerate kernel machines on graphs. *arXiv preprint arXiv:2301.06384*, 2023. Cited on page 19.

Faskowitz, J., Betzel, R. F., and Sporns, O. Edges in brain networks: Contributions to models of structure and function. *Network Neuroscience*, 6(1):1–28, 2022. Cited on page 1.

Fujiwara, Y. and Islam, R. Hodge decomposition of bitcoin money flow. *Advanced Studies of Financial Technologies and Cryptocurrency Markets*, pp. 117–137, 2020. Cited on page 2.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018. Cited on pages 7 and 22.

Godsil, C. and Royle, G. *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*. Springer New York, New York, NY, 2001. ISBN 978-0-387-95220-8 978-1-4613-0163-9. doi: 10.1007/978-1-4613-0163-9. Cited on pages 1, 2, and 7.

Grady, L. J. and Polimeni, J. R. *Discrete Calculus*. Springer London, London, 2010. ISBN 978-1-84996-289-6 978-1-84996-290-2. doi: 10.1007/978-1-84996-290-2. Cited on pages 1, 3, and 4.

Higham, N. J. *Functions of matrices: theory and computation.* SIAM, 2008. Cited on page 20.

Hodge, W. V. D. *The theory and applications of harmonic integrals.* CUP Archive, 1989. Cited on page 4.

Jia, J., Schaub, M. T., Segarra, S., and Benson, A. R. Graph-Based Semi-Supervised & Active Learning for Edge Flows. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 761–771, Anchorage AK USA, July 2019. ACM. ISBN 978-1-4503-6201-6. doi: 10.1145/3292500.3330872. Cited on pages 1, 7, and 22.

Jiang, X., Lim, L.-H., Yao, Y., and Ye, Y. Statistical Ranking and Combinatorial Hodge Theory. *Mathematical Programming*, 127(1):203–244, March 2011. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-010-0419-x. Cited on pages 1, 2, and 7.

Klise, K. A., Hart, D., Moriarty, D. M., Bynum, M. L., Murray, R., Burkhardt, J., and Haxton, T. Water network tool for resilience (WNTR) user manual. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2017. Cited on pages 9 and 24.

Knyazev, A. V. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM journal on scientific computing*, 23(2):517–541, 2001. Cited on pages 8 and 23.

Kondor, R. I. and Lafferty, J. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th international conference on machine learning*, volume 2002, pp. 315–322, 2002. Cited on page 2.

Lim, L.-H. Hodge Laplacians on Graphs. *SIAM Review*, 62(3):685–715, January 2020. ISSN 0036-1445, 1095-7200. doi: 10.1137/18M1223101. Cited on pages 1, 2, 3, 4, and 6.

Lindgren, F., Rue, H., and Lindström, J. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(4):423–498, 2011. Cited on page 2.

Lovász, L. Discrete Analytic Functions: An Exposition. *Surveys in Differential Geometry*, 9(1):241–273, 2004. ISSN 10529233, 21644713. doi: 10.4310/SDG.2004.v9.n1.a7. Cited on page 1.

Lumpkin, R. and Centurioni, L. Global Drifter Program quality-controlled 6-hour interpolated data from ocean surface drifting buoys. NOAA National Centers for Environmental Information. Dataset, 2019. Cited on page 8.

McQueen, J., Meilă, M., VanderPlas, J., and Zhang, Z. Megaman: Scalable manifold learning in python. *The Journal of Machine Learning Research*, 17(1):5176–5180, 2016. Cited on page 23.

Muhammad, A. and Egerstedt, M. Control Using Higher Order Laplacians in Network Topologies. In *Proc. of 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto*, pp. 1024–1038, 2006. Cited on page 6.

Munkres, J. R. *Elements of Algebraic Topology.* The Advanced Book Program. CRC Press, Boca Raton London New York, 2018. ISBN 978-0-201-62728-2 978-0-201-04586-4. Cited on pages 2 and 19.

Nikitin, A. V., John, S., Solin, A., and Kaski, S. Non-Separable Spatio-temporal Graph Kernels via SPDEs. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 10640–10660. PMLR, May 2022. Cited on pages 1, 2, and 20.

Oanda. Oanda Corporation. Foreign Exchange Data, 2018. https://www.oanda.com/ [Accessed: (2018-10-05)]. Cited on page 7.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning.* Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass, 2006. ISBN 978-0-262-18253-9. Cited on pages 1 and 2.

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 7.

Roddenberry, T. M., Glaze, N., and Segarra, S. Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning*, pp. 9020–9029. PMLR, 2021. Cited on page 1.

Schaub, M. T., Lehmann, J., Yaliraki, S. N., and Barahona, M. Structure of complex networks: Quantifying edge-to-edge relations by failure-induced flow redistribution. *Network Science*, 2(1):66–89, 2014. Cited on page 1.

Schaub, M. T., Benson, A. R., Horn, P., Lippner, G., and Jadbabaie, A. Random Walks on Simplicial Complexes and the Normalized Hodge 1-Laplacian. *SIAM Review*, 62(2):353–391, January 2020. ISSN 0036-1445, 1095-7200. doi: 10.1137/18M1201019. Cited on pages 3 and 6.

Schaub, M. T., Zhu, Y., Seby, J.-B., Roddenberry, T. M., and Segarra, S. Signal Processing on Higher-Order Networks: Livin' on the Edge ... and Beyond. *Signal Processing*, 187:108149, October 2021. ISSN 01651684. doi: 10.1016/j.sigpro.2021.108149. Cited on pages 1 and 4.

Sharp, N., Soliman, Y., and Crane, K. The vector heat method. *ACM Transactions on Graphics (TOG)*, 38 (3):1–19, 2019. Cited on page 6.

Smola, A. J. and Kondor, R. Kernels and Regularization on Graphs. In Goos, G., Hartmanis, J., Van Leeuwen, J., Schölkopf, B., and Warmuth, M. K. (eds.), *Learning Theory and Kernel Machines*, volume 2777, pp. 144–158. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-40720-1 978-3-540-45167-9. doi: 10.1007/978-3-540-45167-9_12. Cited on page 1.

Venkitaraman, A., Chatterjee, S., and Handel, P. Gaussian processes over graphs. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5640–5644. IEEE, 2020. Cited on page 1.

Vijay Anand, D., Das, S., and Chung, M. K. Hodge-Decomposition of Brain Networks. *arXiv e-prints*, pp. arXiv–2211, 2022. Cited on page 2.

Vishwanathan, S. V. N., Schraudolph, N. N., Kondor, R., and Borgwardt, K. M. Graph kernels. *Journal of Machine Learning Research*, 11:1201–1242, 2010. Cited on page 1.

Whittle, P. Stochastic-processes in several dimensions. *Bulletin of the International Statistical Institute*, 40 (2):974–994, 1963. Cited on page 2.

Yang, M. and Isufi, E. Convolutional Learning on Simplicial Complexes. *arXiv preprint arXiv:2301.11163*, 2023. Cited on page 1.

Yang, M., Isufi, E., Schaub, M. T., and Leus, G. Finite Impulse Response Filters for Simplicial Complexes. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 2005–2009, August 2021. doi: 10.23919/EUSIPCO54536.2021.9616185. Cited on page 18.

Yang, M., Isufi, E., and Leus, G. Simplicial Convolutional Neural Networks. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8847–8851, May 2022a. doi: 10.1109/ICASSP43922.2022.9746017. Cited on page 1.

Yang, M., Isufi, E., Schaub, M. T., and Leus, G. Simplicial Convolutional Filters. *IEEE Transactions on Signal Processing*, 70:4633–4648, 2022b. ISSN 1941-0476. doi: 10.1109/TSP.2022.3207045. Cited on pages 1, 5, and 18.

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*, 9:304–314, 2023. ISSN 2373-776X. doi: 10.1109/TSIPN.2023.3265160. Cited on page 1.

Zhou, X., Liu, S., Xu, W., Xin, K., Wu, Y., and Meng, F. Bridging hydraulics and graph signal processing: A new perspective to estimate water distribution network pressures. *Water Research*, 217:118416, 2022. Cited on pages 1 and 8.

Zobel, V., Reininghaus, J., and Hotz, I. Generalized Heat Kernel Signatures. *Journal of WSCG*, 2011. Cited on page 6.

Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] We include the problem setting for general Gaussian process in Background, which holds the same for our problem of edge Gaussian process.

   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] We provide the complexity about the edge kernel in Appendix.

   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:

   (a) Statements of the full set of assumptions of all theoretical results. [Yes]

   (b) Complete proofs of all theoretical results. [Yes] We provide all proofs and derivations in Appendix.

   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]

   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]

   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] All measures are defined and experiments are run for ten times.

   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Yes] We cite the works on all the dataset we use.

   (b) The license information of the assets, if applicable. [Not Applicable]

   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

   (d) Information about consent from data providers/curators. [Not Applicable]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials for Hodge-Compositional Edge Gaussian Processes

A BACKGROUND

A.1 Algebraic Representation of Simplicial 2-Complexes

For a SC_2 with N_0 nodes, N_1 edges and N_2 triangles in Section 2, the entries of $\boldsymbol{B}_1 \in \mathbb{R}^{N_0 \times N_1}$ and $\boldsymbol{B}_2 \in \mathbb{R}^{N_1 \times N_2}$ are given by

$$[\boldsymbol{B}_1]_{ie} = \begin{cases} -1, & \text{for } e = [i, \cdot] \\ 1, & \text{for } e = [\cdot, i] \\ 0, & \text{otherwise.} \end{cases} \quad [\boldsymbol{B}_2]_{et} = \begin{cases} 1, & \text{for } e = [i,j], t = [i,j,k] \\ -1, & \text{for } e = [i,k], t = [i,j,k] \\ 1, & \text{for } e = [j,k], t = [i,j,k] \\ 0, & \text{otherwise.} \end{cases} \tag{A.1}$$

In the following, we show the two incidence matrices of the SC in Fig. 1a.

$$\boldsymbol{B}_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{pmatrix} \begin{array}{cccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \end{pmatrix}, \quad \boldsymbol{B}_2 = \begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \\ e_{10} \end{array} \begin{pmatrix} \begin{array}{ccc} t_1 & t_2 & t_3 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array} \end{pmatrix} \tag{A.2}$$

B EDGE GAUSSIAN PROCESSES

Here we provide the additional details on Section 3 and the missing proofs.

B.1 Derivation of Edge GPs from SPDEs on Edges

Here we derive the edge Matérn and diffusion GPs in Eq. (6) from the two SPDEs in Eq. (4).

Proposition B.1. *Given the SPDE with a general differential operator $\Phi(\boldsymbol{L}_1) = \boldsymbol{U}_1 \Phi(\boldsymbol{\Lambda}_1) \boldsymbol{U}_1^\top$ and the stochastic Gaussian noise process $\boldsymbol{w}_1 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$*

$$\Phi(\boldsymbol{L}_1)\boldsymbol{f}_1 = \boldsymbol{w}_1, \tag{B.1}$$

its solution is an edge GP

$$\boldsymbol{f}_1 \sim \mathcal{GP}(\boldsymbol{0}, (\Phi^\top(\boldsymbol{L}_1)\Phi(\boldsymbol{L}_1))^\dagger) \tag{B.2}$$

Proof. By writing out its solution

$$\boldsymbol{f}_1 = \Phi^\dagger(\boldsymbol{L}_1)\boldsymbol{w}_1, \tag{B.3}$$

which is a random process, we can find its covariance as

$$\text{Cov}[\boldsymbol{f}_1] = \Phi^\dagger(\boldsymbol{L}_1)\text{Cov}[\boldsymbol{w}_1](\Phi^\dagger(\boldsymbol{L}_1))^\top = (\Phi^\top(\boldsymbol{L}_1)\Phi(\boldsymbol{L}_1))^\dagger \tag{B.4}$$

□

Corollary B.2. *Matérn and diffusion edge kernels in Eq. (6) given as follows*

$$\boldsymbol{f}_1 \sim \mathcal{GP}\left(\boldsymbol{0}, \left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}_1\right)^{-\nu}\right), \quad \boldsymbol{f}_1 \sim \mathcal{GP}\left(\boldsymbol{0}, e^{-\frac{\kappa^2}{2}\boldsymbol{L}_1}\right) \tag{B.5}$$

are the solutions of the following two SPDEs, respectively.

$$\left(\frac{2\nu}{\kappa^2}\boldsymbol{I} + \boldsymbol{L}_1\right)^{\frac{\nu}{2}}\boldsymbol{f}_1 = \boldsymbol{w}_1, \quad e^{\frac{\kappa^2}{4}\boldsymbol{L}_1}\boldsymbol{f}_1 = \boldsymbol{w}_1. \tag{B.6}$$

Proof. By following the procedure in Proposition B.1, the proof completes. ☐

B.2 Samples of Gradient and Curl Edge GPs

Here we discuss the div and curl properties of samples of gradient and curl GPs in Eq. (14), which completes the proof of Proposition 2 .

Proposition B.3. *Consider the gradient and curl GPs*

$$\boldsymbol{f}_G \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_G), \quad \boldsymbol{f}_C \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_C) \tag{B.7}$$

where the gradient kernel and the curl kernel are

$$\boldsymbol{K}_G = \boldsymbol{U}_G \Psi_G(\boldsymbol{\Lambda}_G)\boldsymbol{U}_G^\top, \quad \boldsymbol{K}_C = \boldsymbol{U}_C \Psi_C(\boldsymbol{\Lambda}_C)\boldsymbol{U}_C^\top. \tag{B.8}$$

Their prior samples are, respectively, curl-free and div-free.

Proof. We focus on the case of gradient GPs. First, we can decompose the gradient kernel in terms of $\boldsymbol{U}_1 = [\boldsymbol{U}_H\ \boldsymbol{U}_G\ \boldsymbol{U}_C]$ as

$$\boldsymbol{K}_G = \boldsymbol{U}_1 \begin{pmatrix} \boldsymbol{0} & & \\ & \Psi_G(\boldsymbol{\Lambda}_G) & \\ & & \boldsymbol{0} \end{pmatrix} \boldsymbol{U}_1^\top. \tag{B.9}$$

From a vector $\boldsymbol{v} = (v_1, \ldots, v_{N_1})^\top$ of variables following independent normal distribution, we can draw a random sample of gradient function as

$$\boldsymbol{f}_G = \boldsymbol{U}_1 \mathrm{diag}([\boldsymbol{0}, \Psi_G^{\frac{1}{2}}(\boldsymbol{\Lambda}_G), \boldsymbol{0}])\boldsymbol{v} \tag{B.10}$$

where $\mathrm{diag}([\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}])$ is the diagonal matrix with $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})^\top$ on its diagonal.

Therefore, their curls are

$$\mathrm{curl}\,\boldsymbol{f}_G = \boldsymbol{B}_2^\top \boldsymbol{U}_1 \mathrm{diag}([\boldsymbol{0}, \Psi_G^{\frac{1}{2}}(\boldsymbol{\Lambda}_G), \boldsymbol{0}]) = \boldsymbol{B}_2^\top \boldsymbol{U}_G \Psi_G^{\frac{1}{2}}(\boldsymbol{\Lambda}_G) = \boldsymbol{0}. \tag{B.11}$$

Likewise, we can show the samples of a curl GP are div-free.

Remark B.4. An alternative proof can follow by studying the curl of the gradient GP which is another GP on triangles as given later by Proposition B.11. The kernel $\boldsymbol{B}_2^\top \boldsymbol{K}_G \boldsymbol{B}_2$ is zero, due to the orthogonality $\boldsymbol{B}_2^\top \boldsymbol{U}_G = \boldsymbol{0}$. Thus, the curl of a gradient GP is a zero GP on triangles, as well as its samples. Similarly, one can show the div of a curl GP is a zero GP on nodes, thus, its samples are zero.

☐

B.3 Derivation of Gradient and Curl GPs from SPDEs

Here we provide proofs for Proposition 3, deriving Matérn and diffusion gradient/curl GPs from their SPDE representations.

Proposition B.5. *Given a scaled curl white noise $\boldsymbol{w}_C \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{W}_C)$ where $\boldsymbol{W}_C = \sigma_C^2 \boldsymbol{U}_C \boldsymbol{U}_C^\top$, consider the following SPDE on edges:*

$$\Phi_C(\boldsymbol{L}_\mathrm{u})\boldsymbol{f}_C = \boldsymbol{w}_C, \tag{B.12}$$

with differential operators

$$\Phi_C(\boldsymbol{L}_{\mathrm{u}}) = \left(\frac{2\nu_C}{\kappa_C^2}\boldsymbol{I} + \boldsymbol{L}_{\mathrm{u}}\right)^{\frac{\nu_C}{2}}, \quad \Phi_C(\boldsymbol{L}_{\mathrm{u}}) = e^{\frac{\kappa_C^2}{4}\boldsymbol{L}_{\mathrm{u}}}. \tag{B.13}$$

The respective solutions give the curl edge GPs with Matérn kernel and diffusion kernel

$$\boldsymbol{f}_C \sim \mathcal{GP}\left(\boldsymbol{0}, \sigma_C^2 \boldsymbol{U}_C \left(\frac{2\nu_C}{\kappa_C^2}\boldsymbol{I} + \boldsymbol{L}_{\mathrm{u}}\right)^{-\nu_C}\boldsymbol{U}_C^{\top}\right), \quad \boldsymbol{f}_C \sim \mathcal{GP}\left(\boldsymbol{0}, \sigma_C^2 \boldsymbol{U}_C e^{-\frac{\kappa_C^2}{2}}\boldsymbol{U}_C^{\top}\right). \tag{B.14}$$

Proof. First, consider the Matérn curl GP case. The corresponding SPDE has the form

$$\left(\frac{2\nu_C}{\kappa_C^2}\boldsymbol{I} + \boldsymbol{L}_{\mathrm{u}}\right)^{\frac{\nu_C}{2}}\boldsymbol{f}_C = \boldsymbol{w}_C, \tag{B.15}$$

with a solution $\boldsymbol{f}_C = \Phi_C^{\dagger}(\boldsymbol{L}_{\mathrm{u}})\boldsymbol{w}_C$.

Given the scaled curl Gaussian noise process $\boldsymbol{w}_C \sim \mathcal{G}(\boldsymbol{0}, \boldsymbol{W}_C)$ with $\boldsymbol{W}_C = \sigma_C^2 \boldsymbol{U}_C \boldsymbol{U}_C^{\top}$, the solution $\boldsymbol{f}_C$ is an edge GP following $\boldsymbol{f}_C \sim \mathcal{GP}(\boldsymbol{0}, \mathrm{Cov}[\boldsymbol{f}_C])$ with the covariance of solution $\boldsymbol{f}_C$ as

$$\mathrm{Cov}[\boldsymbol{f}_C] = \left(\frac{2\nu_C}{\kappa_C^2}\boldsymbol{I} + \boldsymbol{L}_{\mathrm{u}}\right)^{-\frac{\nu_C}{2}}\boldsymbol{W}_C\left(\frac{2\nu_C}{\kappa_C^2}\boldsymbol{I} + \boldsymbol{L}_{\mathrm{u}}\right)^{-\frac{\nu_C}{2}}. \tag{B.16}$$

Note that we have

$$\boldsymbol{W}_C = \begin{pmatrix} \boldsymbol{U}_H & \boldsymbol{U}_G & \boldsymbol{U}_C \end{pmatrix} \begin{pmatrix} \boldsymbol{0} & & \\ & \boldsymbol{0} & \\ & & \sigma_C^2\boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{U}_H & \boldsymbol{U}_G & \boldsymbol{U}_C \end{pmatrix}^{\top}. \tag{B.17}$$

Moreover, $\boldsymbol{L}_{\mathrm{u}}$ can be decomposed by $\boldsymbol{U}_1$ as follows

$$\boldsymbol{L}_{\mathrm{u}} = \begin{pmatrix} \boldsymbol{U}_H & \boldsymbol{U}_G & \boldsymbol{U}_C \end{pmatrix} \begin{pmatrix} \boldsymbol{0} & & \\ & \boldsymbol{0} & \\ & & \boldsymbol{\Lambda}_C \end{pmatrix} \begin{pmatrix} \boldsymbol{U}_H & \boldsymbol{U}_G & \boldsymbol{U}_C \end{pmatrix}^{\top}, \tag{B.18}$$

which follows that

$$\left(\frac{2\nu_C}{\kappa_C^2}\boldsymbol{I} + \boldsymbol{L}_{\mathrm{u}}\right)^{-\frac{\nu_C}{2}} = \begin{pmatrix} \boldsymbol{U}_H & \boldsymbol{U}_G & \boldsymbol{U}_C \end{pmatrix} \begin{pmatrix} \left(\frac{2\nu_C}{\kappa_C^2}\boldsymbol{I}\right)^{-\frac{\nu_C}{2}} & & \\ & \left(\frac{2\nu_C}{\kappa_C^2}\boldsymbol{I}\right)^{-\frac{\nu_C}{2}} & \\ & & \left(\frac{2\nu_C}{\kappa_C^2}\boldsymbol{I} + \boldsymbol{\Lambda}_C\right)^{-\frac{\nu_C}{2}} \end{pmatrix} \begin{pmatrix} \boldsymbol{U}_H & \boldsymbol{U}_G & \boldsymbol{U}_C \end{pmatrix}^{\top}. \tag{B.19}$$

By plugging Eq. (B.17) and Eq. (B.19) into Eq. (B.16), we can then express the covariance as

$$\begin{aligned} \mathrm{Cov}[\boldsymbol{f}_C] &= \begin{pmatrix} \boldsymbol{U}_H & \boldsymbol{U}_G & \boldsymbol{U}_C \end{pmatrix} \begin{pmatrix} \boldsymbol{0} & & \\ & \boldsymbol{0} & \\ & & \sigma_C^2\left(\frac{2\nu_C}{\kappa_C^2}\boldsymbol{I} + \boldsymbol{\Lambda}_C\right)^{-\nu_C} \end{pmatrix} \begin{pmatrix} \boldsymbol{U}_H & \boldsymbol{U}_G & \boldsymbol{U}_C \end{pmatrix}^{\top} \\ &= \boldsymbol{U}_C\sigma_C^2\left(\frac{2\nu_C}{\kappa_C^2}\boldsymbol{I} + \boldsymbol{\Lambda}_C\right)^{-\nu_C}\boldsymbol{U}_C^{\top} \end{aligned} \tag{B.20}$$

which returns the Matérn curl GP $\boldsymbol{f}_C \sim \mathcal{GP}\left(\boldsymbol{0}, \sigma_C^2 \boldsymbol{U}_C\left(\frac{2\nu_C}{\kappa_C^2}\boldsymbol{I} + \boldsymbol{L}_{\mathrm{u}}\right)^{-\nu_C}\boldsymbol{U}_C^{\top}\right)$.

Second, consider the following SPDE

$$e^{\frac{\kappa_C^2}{4}\boldsymbol{L}_{\mathrm{u}}}\boldsymbol{f}_C = \boldsymbol{w}_C. \tag{B.21}$$

Following the same procedure as above, we have its solution as

$$\boldsymbol{f}_C \sim \mathcal{GP}\left(\boldsymbol{0}, \sigma_C^2 \boldsymbol{U}_C e^{-\frac{\kappa_C^2}{2}}\boldsymbol{U}_C^{\top}\right) \tag{B.22}$$

which is the diffusion curl GP. $\qquad\square$

Proposition B.6. *Given a scaled gradient white noise $\boldsymbol{w}_G \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{W}_G)$ where $\boldsymbol{W}_G = \sigma_G^2 \boldsymbol{U}_G \boldsymbol{U}_G^\top$, consider the following SPDE on edges:*

$$\Phi_G(\boldsymbol{L}_{\mathrm{d}})\boldsymbol{f}_G = \boldsymbol{w}_G, \tag{B.23}$$

with differential operators

$$\Phi_G(\boldsymbol{L}_{\mathrm{d}}) = \left(\frac{2\nu_G}{\kappa_G^2}\boldsymbol{I} + \boldsymbol{L}_{\mathrm{d}}\right)^{\frac{\nu_G}{2}}, \quad \Phi_G(\boldsymbol{L}_{\mathrm{d}}) = e^{\frac{\kappa_G^2}{4}\boldsymbol{L}_{\mathrm{d}}}. \tag{B.24}$$

The respective solutions give the curl edge GPs with Matérn kernel and diffusion kernel

$$\boldsymbol{f}_G \sim \mathcal{GP}\left(\boldsymbol{0}, \sigma_G^2 \boldsymbol{U}_G \left(\frac{2\nu_G}{\kappa_G^2}\boldsymbol{I} + \boldsymbol{L}_{\mathrm{d}}\right)^{-\nu_G} \boldsymbol{U}_G^\top\right) \quad \boldsymbol{f}_G \sim \mathcal{GP}\left(\boldsymbol{0}, \sigma_G^2 \boldsymbol{U}_G e^{-\frac{\kappa_G^2}{2}} \boldsymbol{U}_G^\top\right). \tag{B.25}$$

Proof. The proof follows Proposition B.5 likewise. $\square$

B.4 Proof of Properties of HC Edge GPs

Here we provide proofs for Lemma 5, which directly follow from Definition 4.

Proof. For an edge GP $\boldsymbol{f}_1$ with covariance kernel $\boldsymbol{K}_1$, due to the fact that the HC edge kernel $\boldsymbol{K}_1$ is built using all the orthonormal basis of the edge function space $\boldsymbol{U}_1$, its realizations give all possible edge functions. This is analogous to Karhunen-Loève theorem for GPs with Mercer kernels. For the second point that $\boldsymbol{K}_1 = \boldsymbol{K}_H + \boldsymbol{K}_G + \boldsymbol{K}_C$ and the three Hodge GPs mutually independent, this results from the construction of $\boldsymbol{f}_1$ and the orthogonality of the three Hodge GPs. $\square$

B.5 Posterior Distributions of Hodge Components

Here we discuss the posterior distribution of the three Hodge components from the posterior prediction of the edge function. As the construction of our HC edge GPs is essentially a sum of three independent functions, we can follow Duvenaud (2014, Section 2.4) modeling the sums of Euclidean functions. Denote $\boldsymbol{f}_1(\boldsymbol{x})$ and $\boldsymbol{f}_1(\boldsymbol{x}^*)$ the function values, respectively, at training locations $\boldsymbol{x} = [x_1, \ldots, x_n]^\top$ and query locations $\boldsymbol{x}^* = [x_1^*, \ldots, x_n^*]^\top$. We first write down the joint prior distribution over the three Hodge components and the edge function.

$$
\begin{bmatrix}
\boldsymbol{f}_H(\boldsymbol{x}) \\
\boldsymbol{f}_H(\boldsymbol{x}^*) \\
\boldsymbol{f}_G(\boldsymbol{x}) \\
\boldsymbol{f}_G(\boldsymbol{x}^*) \\
\boldsymbol{f}_C(\boldsymbol{x}) \\
\boldsymbol{f}_C(\boldsymbol{x}^*) \\
\boldsymbol{f}_1(\boldsymbol{x}) \\
\boldsymbol{f}_1(\boldsymbol{x}^*)
\end{bmatrix}
\sim \mathcal{N} \left(\boldsymbol{0},
\begin{bmatrix}
\boldsymbol{K}_H & \boldsymbol{K}_H^* & & & & & \boldsymbol{K}_H & \boldsymbol{K}_H^* \\
\boldsymbol{K}_H^{*\top} & \boldsymbol{K}_H^{**} & & & & & \boldsymbol{K}_H^* & \boldsymbol{K}_H^{**} \\
& & \boldsymbol{K}_G & \boldsymbol{K}_G^* & & & \boldsymbol{K}_G & \boldsymbol{K}_G^* \\
& & \boldsymbol{K}_G^{*\top} & \boldsymbol{K}_G^{**} & & & \boldsymbol{K}_G^* & \boldsymbol{K}_G^{**} \\
& & & & \boldsymbol{K}_C & \boldsymbol{K}_C^* & \boldsymbol{K}_C & \boldsymbol{K}_C^* \\
& & & & \boldsymbol{K}_C^{*\top} & \boldsymbol{K}_C^{**} & \boldsymbol{K}_C^* & \boldsymbol{K}_C^{**} \\
\boldsymbol{K}_H & \boldsymbol{K}_H^{*\top} & \boldsymbol{K}_G & \boldsymbol{K}_G^{*\top} & \boldsymbol{K}_C & \boldsymbol{K}_C^{*\top} & \boldsymbol{K}_1 & \boldsymbol{K}_1^* \\
\boldsymbol{K}_H^{*\top} & \boldsymbol{K}_H^{**} & \boldsymbol{K}_G^{*\top} & \boldsymbol{K}_G^{**} & \boldsymbol{K}_C^{*\top} & \boldsymbol{K}_C^{**} & \boldsymbol{K}_1^{*\top} & \boldsymbol{K}_1^{**}
\end{bmatrix}
\right) \tag{B.26}
$$

where we represent the kernel matrices by $\boldsymbol{K}_1 = k_1(\boldsymbol{x}, \boldsymbol{x}), \boldsymbol{K}_1^* = k_1(\boldsymbol{x}, \boldsymbol{x}^*)$ and $\boldsymbol{K}_1^{**} = k_1(\boldsymbol{x}^*, \boldsymbol{x}^*)$, and likewise for the other kernel matrices. Given this joint distribution, we can obtain the posterior distributions of the three Hodge components as follows

$$\boldsymbol{f}_H(\boldsymbol{x}^*)|\boldsymbol{f}_1(\boldsymbol{x}) \sim \mathcal{N}\left(\boldsymbol{K}_H^{*\top}\boldsymbol{K}_1^{-1}\boldsymbol{f}_1(\boldsymbol{x}), \boldsymbol{K}_H^{**} - \boldsymbol{K}_H^{*\top}\boldsymbol{K}_1^{-1}\boldsymbol{K}_H^*\right) \tag{B.27a}$$

$$\boldsymbol{f}_G(\boldsymbol{x}^*)|\boldsymbol{f}_1(\boldsymbol{x}) \sim \mathcal{N}\left(\boldsymbol{K}_G^{*\top}\boldsymbol{K}_1^{-1}\boldsymbol{f}_1(\boldsymbol{x}), \boldsymbol{K}_G^{**} - \boldsymbol{K}_G^{*\top}\boldsymbol{K}_1^{-1}\boldsymbol{K}_G^*\right) \tag{B.27b}$$

$$\boldsymbol{f}_C(\boldsymbol{x}^*)|\boldsymbol{f}_1(\boldsymbol{x}) \sim \mathcal{N}\left(\boldsymbol{K}_C^{*\top}\boldsymbol{K}_1^{-1}\boldsymbol{f}_1(\boldsymbol{x}), \boldsymbol{K}_C^{**} - \boldsymbol{K}_C^{*\top}\boldsymbol{K}_1^{-1}\boldsymbol{K}_C^*\right) \tag{B.27c}$$

From these posterior distributions, we can directly obtain the means and the uncertainties of the Hodge components of the predicted edge function.

B.6 Edge Fourier Feature Perspective

Edge Fourier transform From the edge eigen-feature perspective, any edge function can be viewed as a linear combination of eigenvectors in $\boldsymbol{U}$, that is,

$$\boldsymbol{f}_1 = \sum_{i=1}^{N_1} \tilde{f}_{1,i}\boldsymbol{u}_i = \boldsymbol{U}_1 \tilde{\boldsymbol{f}}_1 \text{ with } \tilde{\boldsymbol{f}}_1 = \boldsymbol{U}^\top \boldsymbol{f}_1 \tag{B.28}$$

where $\tilde{\boldsymbol{f}}_1$ is known as the (edge) Fourier feature of $\boldsymbol{f}_1$ and $\tilde{f}_{1,i}$ is the i-th Fourier coefficient at eigenvalue λ_i. These eigenvalues carry the notion of frequency (Barbarossa & Sardellitti, 2020). Particularly, based on the reorganized eigenvector matrix $\boldsymbol{U}_1 = [\boldsymbol{U}_H\ \boldsymbol{U}_G\ \boldsymbol{U}_C]$ and the associated eigenvalues $\boldsymbol{\Lambda}_1 = \text{diag}(\boldsymbol{\Lambda}_H, \boldsymbol{\Lambda}_G, \boldsymbol{\Lambda}_C)$, we have that any λ_G measures the squared ℓ_2-norm of the divergence while λ_C measures the squared ℓ_2-norm of the curl: $\lambda_G = \boldsymbol{u}_G^\top \boldsymbol{L}_1 \boldsymbol{u}_G = \boldsymbol{u}_G^\top \boldsymbol{L}_\text{d} \boldsymbol{u}_G = \|\boldsymbol{B}_1 \boldsymbol{u}_G\|_2^2$, and $\lambda_C = \boldsymbol{u}_C^\top \boldsymbol{L}_1 \boldsymbol{u}_C = \boldsymbol{u}_C^\top \boldsymbol{L}_\text{u} \boldsymbol{u}_C = \|\boldsymbol{B}_2^\top \boldsymbol{u}_C\|_2^2$, and a zero eigenvalue $\lambda_H = 0$ corresponding to harmonic eigenvector μ_H has zero total divergence and curl, as discussed by (Yang et al., 2021, 2022b). Therefore, the Fourier coefficients at eigenvalues in different Hodge subspaces measure the weights of the corresponding Fourier basis in $\boldsymbol{f}$, each basis associated with different total divergence or total curl. That is, we have the edge Fourier representation as

$$\tilde{\boldsymbol{f}}_1 = \boldsymbol{U}_1^\top \boldsymbol{f}_1 = [\tilde{\boldsymbol{f}}_H^\top, \tilde{\boldsymbol{f}}_G^\top, \tilde{\boldsymbol{f}}_C^\top]^\top \text{ with } \tilde{\boldsymbol{f}}_H = \boldsymbol{U}_H^\top \boldsymbol{f}_1,\ \tilde{\boldsymbol{f}}_G = \boldsymbol{U}_G^\top \boldsymbol{f}_1,\ \tilde{\boldsymbol{f}}_C = \boldsymbol{U}_C^\top \boldsymbol{f}_1. \tag{B.29}$$

Remark B.7. This provides as a spectral tool to understand the edge functions. The harmonic Fourier feature $\tilde{\boldsymbol{f}}_H$ measures the extent of harmonic Fourier basis $\boldsymbol{U}_H$ in $\boldsymbol{f}$, reflecting how harmonic $\boldsymbol{f}_1$ is. The gradient Fourier feature $\tilde{\boldsymbol{f}}_G$ measures the extent of gradient Fourier basis $\boldsymbol{U}_G$ in $\boldsymbol{f}_1$, reflecting how divergent $\boldsymbol{f}_1$ is, where each basis in $\boldsymbol{U}_G$ has different total divergence. The curl Fourier feature $\tilde{\boldsymbol{f}}_C$ measures the extent of curl Fourier basis $\boldsymbol{U}_C$ in $\boldsymbol{f}_1$, reflecting how rotational $\boldsymbol{f}_1$ is, where each basis in $\boldsymbol{U}_C$ has different total curl.

Corollary B.8 (Fourier feature perspective of edge GPs)**.** *Let $\boldsymbol{f}_1 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_1)$ be an edge Gaussian process with kernel diagonalizable by $\boldsymbol{U}_1$. Then, given the edge Fourier transform $\tilde{\boldsymbol{f}}_1 = \boldsymbol{U}_1^\top \boldsymbol{f}_1$, its Fourier coefficients $\{\tilde{f}_{1,i}\}_{i=1}^N$ are independently distributed Gaussian variables*

$$\tilde{f}_{1,i} \sim \mathcal{N}(0, \boldsymbol{u}_i^\top \boldsymbol{K}_1 \boldsymbol{u}_i),\ \text{for } i = 1, \ldots, N. \tag{B.30}$$

Proof. Using the affine transformation preserving Gaussian, we have

$$\tilde{\boldsymbol{f}}_1 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{U}_1^\top \boldsymbol{K}_1 \boldsymbol{U}_1). \tag{B.31}$$

Since the kernel $\boldsymbol{K}_1$ can be diagonalized by $\boldsymbol{U}_1$, the kernel $\boldsymbol{U}_1^\top \boldsymbol{K}_1 \boldsymbol{U}_1$ is a diagonal matrix, implying the independence between variables in $\tilde{\boldsymbol{f}}_1$. Thus, a variable $\tilde{f}_{1,i}$ follows normal distribution $\mathcal{N}(0, \boldsymbol{u}_i^\top \boldsymbol{K}_1 \boldsymbol{u}_i)$. □

This corollary indicates that an edge GP can be viewed as an affine transformation by $\boldsymbol{U}_1$ of a collection of independent Gaussian variables, $\tilde{\boldsymbol{f}}_1 = [\tilde{f}_{1,1}, \ldots, \tilde{f}_{1,N_1}]^\top$, which are the Fourier coefficients of $\boldsymbol{f}$. The prior distribution of certain Fourier coefficient is the prior imposed on the corresponding divergent or rotational part of the function $\boldsymbol{f}$. This allows us to compare HC and non-HC edge GPs from the following perspective.

Proposition B.9. *Suppose the Hodge Laplacian $\boldsymbol{L}_1$ has eigenpairs $(\lambda, \boldsymbol{u}_G)$ and $(\lambda, \boldsymbol{u}_C)$, i.e., λ is associated to both gradient and curl subspaces. Let $\boldsymbol{f}_1 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_1)$ be an edge Gaussian process. Denote the Fourier coefficients of $\boldsymbol{f}_1$ at λ_G and λ_C as $\tilde{f}_G$ and $\tilde{f}_C$, respectively. Then, a non-Hodge-compositional GP with $\boldsymbol{K}_1 = \Psi(\boldsymbol{L}_1)$ imposes the same prior variance on $\tilde{f}_G$ and $\tilde{f}_C$, i.e.,*

$$\text{Var}[\tilde{f}_G] = \text{Var}[\tilde{f}_C] = \Psi(\lambda). \tag{B.32}$$

Instead, a Hodge-compositional GP with $\boldsymbol{K}_1$ in Eq. (20) imposes different variances on two coefficients

$$\text{Var}[\tilde{f}_G] = \Psi_G(\lambda)\ \text{and}\ \text{Var}[\tilde{f}_C] = \Psi_C(\lambda). \tag{B.33}$$

Proof. For a non-HC edge GP with kernel $\Psi(\boldsymbol{L}_1)$, its Fourier coefficients $\tilde{f}_G$ and $\tilde{f}_C$ at a common λ follows the normal distribution with a variance $\Psi(\lambda)$, which follows from the nature of kernel function Ψ mapping each λ to exactly one value $\Psi(\lambda)$. However, for a HC edge GP, we have

$$\tilde{f}_G \sim \mathcal{N}(0, \boldsymbol{u}_G^\top \boldsymbol{K}_1 \boldsymbol{u}_G),\quad \tilde{f}_C \sim \mathcal{N}(0, \boldsymbol{u}_C^\top \boldsymbol{K}_1 \boldsymbol{u}_C). \tag{B.34}$$

Using Definition 4 [cf. Eq. (20)], we have $\boldsymbol{u}_G^\top \boldsymbol{K}_1 \boldsymbol{u}_G = \Psi_G(\lambda)$ and $\boldsymbol{u}_C^\top \boldsymbol{K}_1 \boldsymbol{u}_C = \Psi_C(\lambda)$, which are two different values, arising from the individually parametrized kernels $\boldsymbol{K}_G$ and $\boldsymbol{K}_C$. $\qquad\square$

This edge Fourier feature perspective directly shows that non-HC GPs impose the same prior on two Fourier coefficients, which are however associated with two different Hodge subspaces. This prohibits individual learning for the gradient and curl parts of edge functions particularly associated to the same eigenvalue. Instead, HC edge GPs do not have this limitation.

B.7 Diffusion on Edges

Here we provide the details on the connection of diffusion HC edge GPs to edge diffusion equations, as well as an illustration of diffusion process on edges. Consider the diffusion equation on the edge space

$$\frac{\mathrm{d}\boldsymbol{\phi}(t)}{\mathrm{d}t} = -(\mu \boldsymbol{L}_\mathrm{d} + \gamma \boldsymbol{L}_\mathrm{u})\boldsymbol{\phi}(t) \qquad (\mathrm{B.35})$$

where $\mu, \gamma > 0$. Given an initial value $\boldsymbol{\phi}(0)$, we obtain a solution

$$\boldsymbol{\phi}|_{t=\tau} = e^{-(\mu\tau \boldsymbol{L}_\mathrm{d} + \gamma\tau \boldsymbol{L}_\mathrm{u})}\boldsymbol{\phi}(0), \qquad (\mathrm{B.36})$$

When $\sigma_G^2 = \sigma_C^2 = \sigma_H^2 = 1$, the diffusion kernel can be written as

$$\boldsymbol{K}_1 = e^{-(\frac{\kappa_G^2}{2}\boldsymbol{L}_\mathrm{d} + \frac{\kappa_C^2}{2}\boldsymbol{L}_\mathrm{u})} \qquad (\mathrm{B.37})$$

which is the Green's function of above diffusion equation. In Fig. B.1, we illustrate the diffusion processes on nodes and on edges, started at a random location. When the graph is connected, the node diffusion converges to the harmonic state where all nodes are constant. Instead, the harmonic state of the edge diffusion gives an edge flow which is div- and curl-free, cycling around the 1-dimensional "hole" of the SC_2 (Munkres, 2018).
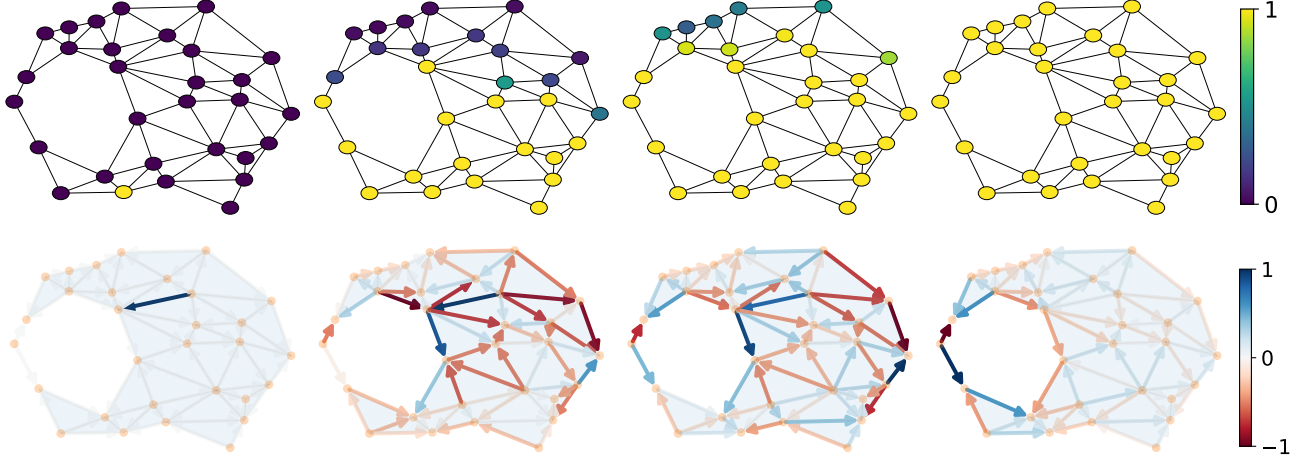

Figure B.1: Node *(Top)* and edge *(Bottom)* diffusion processes (started at one random location *(Left)*, then two intermediate states *(Center)* and harmonic state *(Right)*).

B.8 Complexity of Edge GPs

Here we discuss their complexity when training, e.g., in Gaussian process regression, and the complexity of sampling from them. Note that the complexity of graph GPs naturally apply to edge GPs.

Complexity when Training The Matérn and diffusion kernels can be trained in a scalable way. Due to their decreasing eigenvalues, we can consider the l largest eigenvalues of the kernel matrices with off-the-shelf eigen-solvers, e.g., Lanczos algorithm. The recent work on Krylov subspace methods to accelerate graph kernels by Erb (2023) can be extended to edge kernels. Moreover, other computational techniques applicable for graph GPs in Borovitskiy et al. (2021, Section 3.1) can be adopted as well.

Complexity when Sampling from Edge GPs Given an edge GP, as well as the eigenpairs for constructing the edge kernel, we can follow the procedure in Appendix B.2 to sample an edge function. That is, from a vector $\boldsymbol{v} = (v_1, \dots, v_{N_1})^\top$ of variables following independent normal distribution, a sample of the edge function can be given by

$$\boldsymbol{f}_1 = [\boldsymbol{U}_H\ \boldsymbol{U}_G\ \boldsymbol{U}_C]\operatorname{diag}([\Psi_H^{\frac{1}{2}}(\boldsymbol{\Lambda}_H), \Psi_G^{\frac{1}{2}}(\boldsymbol{\Lambda}_G), \Psi_C^{\frac{1}{2}}(\boldsymbol{\Lambda}_C)])\boldsymbol{v} \tag{B.38}$$

which has a complexity of $\mathcal{O}(N_1^2)$ (matrix-vector multiplication). Furthermore, the discussion on improving sampling efficiency in graph GP models by Nikitin et al. (2022, Section 4.7) naturally applies to our proposed edge GPs as well.

B.9 Interaction between Node, Edge and Triangle GPs

Here we provide the proof for Corollary 6, showing the gradient of a node GP is an edge GP.

Proof. Given a node GP $\boldsymbol{f}_0 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_0)$, using the derivative of a GP is also a GP, its gradient $\boldsymbol{f}_G = \boldsymbol{B}_1^\top \boldsymbol{f}_0$ is an edge GP whose kernel can be found as

$$\boldsymbol{K}_G = \operatorname{Cov}[\boldsymbol{f}_G] = \boldsymbol{B}_1^\top \operatorname{Cov}[\boldsymbol{f}_0]\boldsymbol{B}_1 = \boldsymbol{B}_1^\top \boldsymbol{K}_0 \boldsymbol{B}_1. \tag{B.39}$$

By definition, $\boldsymbol{L}_0 = \boldsymbol{B}_1\boldsymbol{B}_1^\top$ and $\boldsymbol{L}_\mathrm{d} = \boldsymbol{B}_1^\top \boldsymbol{B}_1$ are isospectral, having the same nonzero eigenvalues. Furthermore, using $\boldsymbol{K}_0 = \Psi_0(\boldsymbol{L}_0)$, we can write above covariance as

$$\boldsymbol{K}_G = \boldsymbol{B}_1^\top \Psi(\boldsymbol{B}_1\boldsymbol{B}_1^\top)\boldsymbol{B}_1 = \boldsymbol{B}_1^\top \boldsymbol{B}_1 \Psi_0(\boldsymbol{B}_1^\top \boldsymbol{B}_1) = \boldsymbol{L}_\mathrm{d}\Psi_0(\boldsymbol{L}_\mathrm{d}) \tag{B.40}$$

where the second equality can be shown by using the definition of analytic functions of matrix (Higham, 2008, Corollary 1.34). Furthermore, relying on the eigendecomposition

$$\boldsymbol{L}_\mathrm{d} = \begin{pmatrix} \boldsymbol{U}_H & \boldsymbol{U}_G & \boldsymbol{U}_C \end{pmatrix} \begin{pmatrix} \boldsymbol{0} & & \\ & \boldsymbol{\Lambda}_G & \\ & & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{U}_H & \boldsymbol{U}_G & \boldsymbol{U}_C \end{pmatrix}^\top, \tag{B.41}$$

we can obtain

$$\boldsymbol{K}_G = \boldsymbol{U}_G \boldsymbol{\Lambda}_G \Psi_0(\boldsymbol{\Lambda})\boldsymbol{U}_G^\top, \tag{B.42}$$

which gives the gradient kernel function $\Psi_G(\boldsymbol{\Lambda}_G) = \boldsymbol{\Lambda}_G \Psi_0(\boldsymbol{\Lambda}_G)$. $\square$

In the following we provide the respective corollaries for other derivative operations of interest, where the proofs can directly follow from the fact that derivatives preserve Gaussianity.

Corollary B.10 (Curl of a triangle GP). *Suppose a triangle function $\boldsymbol{f}_2$ is a GP $\boldsymbol{f}_2 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_2)$ with $\boldsymbol{K}_2 = \Psi_2(\boldsymbol{L}_2) = \boldsymbol{U}_2 \Psi_2(\boldsymbol{\Lambda}_2)\boldsymbol{U}_2^\top$ given the eigendecomposition $\boldsymbol{L}_2 = \boldsymbol{U}_2 \boldsymbol{\Lambda}_2 \boldsymbol{U}_2^\top$. Then, its curl is an edge GP $\boldsymbol{f}_C \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_C)$ where $\boldsymbol{K}_C = \boldsymbol{U}_C \Psi_C(\boldsymbol{\Lambda}_C)\boldsymbol{U}_C^\top$ with*

$$\Psi_C(\boldsymbol{\Lambda}_C) = \boldsymbol{\Lambda}_C \Psi_2(\boldsymbol{\Lambda}_C). \tag{B.43}$$

Proposition B.11 (Div and Curl of edge GPs). *Let $\boldsymbol{f}_1 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K})$ be a Hodge-compositional edge Gaussian process in Definition 4. Then, its divergence and curl are Gaussian processes on nodes and triangles, respectively, as follows*

$$\boldsymbol{B}_1\boldsymbol{f} \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{B}_1\boldsymbol{K}_G\boldsymbol{B}_1^\top), \quad \boldsymbol{B}_2^\top \boldsymbol{f} \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{B}_2^\top \boldsymbol{K}_C\boldsymbol{B}_2). \tag{B.44}$$

Remark B.12. These interactions between GPs on nodes, edges and triangles provide us alternative ways to construct gradient and curl edge GPs [cf. Corollaries 6 and B.10], as well as construct appropriate node GPs and triangle GPs. They are more applicable when the underlying physical relationships exist between the corresponding functions and the GP priors on the original simplices are easier to construct.

B.10 Alternative Hodge-compositional Edge GPs

Here we provide the proof for Proposition 7 giving an alternative way to build HC edge GPs.

Proof. From the Hodge decomposition, we can write an edge function as

$$\boldsymbol{f}_1 = \boldsymbol{f}_H + \boldsymbol{B}_1^\top \boldsymbol{f}_0 + \boldsymbol{B}_2 \boldsymbol{f}_2. \tag{B.45}$$

where $\boldsymbol{f}_0$ and $\boldsymbol{f}_2$ are some node and triangle functions. Then, the proof can be completed by using the results from Corollaries 6 and B.10. □

B.11 Alternative HC Edge GPs from SPDEs on Edges

While gradient and curl edge GPs in Definition 4 can be linked to their SPDEs as discussed by Proposition 3, we can also obtain the alternatively constructed counterparts in Corollaries 6 and B.10 from SPDEs. Again, we consider the Matérn family.

Corollary B.13. *Suppose a node function* $\boldsymbol{f}_0$ *is a graph (node) Matérn GP* $\boldsymbol{f}_0 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_0)$ *with*

$$\boldsymbol{K}_0 = \Psi_0(\boldsymbol{L}_0) = \left(\frac{2\nu_0}{\kappa_0^2}\boldsymbol{I} + \boldsymbol{L}_0\right)^{-\nu_0}. \tag{B.46}$$

Then, Corollary 6 gives us its gradient as a gradient edge GP $\boldsymbol{f}_G \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_G)$ *with*

$$\boldsymbol{K}_G = \boldsymbol{L}_\mathrm{d}\left(\frac{2\nu_0}{\kappa_0^2}\boldsymbol{I} + \boldsymbol{L}_\mathrm{d}\right)^{-\nu_0}. \tag{B.47}$$

Furthermore, the gradient GP $\boldsymbol{f}_G$ *is the solution of the following SPDE*

$$\left(\frac{2\nu_0}{\kappa_0^2}\boldsymbol{I} + \boldsymbol{L}_\mathrm{d}\right)^{\frac{\nu_0}{2}} \boldsymbol{f}_G = \boldsymbol{B}_1^\top \boldsymbol{w}_0 \tag{B.48}$$

where $\boldsymbol{w}_0$ *is a standard Gaussian noise on nodes following* $\boldsymbol{f}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

Proof. First, we can solve the SPDE with the following solution

$$\boldsymbol{f}_G = \left(\frac{2\nu_0}{\kappa_0^2}\boldsymbol{I} + \boldsymbol{L}_\mathrm{d}\right)^{-\frac{\nu_0}{2}} \boldsymbol{B}_1^\top \boldsymbol{w}_0 = \boldsymbol{B}_1^\top \left(\frac{2\nu_0}{\kappa_0^2}\boldsymbol{I} + \boldsymbol{L}_0\right)^{-\frac{\nu_0}{2}} \boldsymbol{w}_0 \tag{B.49}$$

where the second equality follows from the definition of $\boldsymbol{L}_0$ and $\boldsymbol{L}_\mathrm{d}$. Given that $\boldsymbol{w}_0$ is a GP, so is $\boldsymbol{f}_G$ and we can study its covariance as

$$\begin{aligned}
\mathrm{Cov}[\boldsymbol{f}_G] &= \boldsymbol{B}_1^\top \left(\frac{2\nu_0}{\kappa_0^2}\boldsymbol{I} + \boldsymbol{L}_0\right)^{-\frac{\nu_0}{2}} \mathrm{Cov}[\boldsymbol{w}_0]\left(\frac{2\nu_0}{\kappa_0^2}\boldsymbol{I} + \boldsymbol{L}_0\right)^{-\frac{\nu_0}{2}} \boldsymbol{B}_1 \\
&= \boldsymbol{B}_1^\top \left(\frac{2\nu_0}{\kappa_0^2}\boldsymbol{I} + \boldsymbol{L}_0\right)^{-\nu_0} \boldsymbol{B}_1 \\
&= \boldsymbol{L}_\mathrm{d}\left(\frac{2\nu_0}{\kappa_0^2}\boldsymbol{I} + \boldsymbol{L}_\mathrm{d}\right)^{-\nu_0}
\end{aligned} \tag{B.50}$$

which completes the proof. □

For completeness, we give the corollary relating the curl Matérn edge GP obtained from some triangle GP to its SPDE representation.

Corollary B.14. *Suppose a triangle function* $\boldsymbol{f}_2$ *is a triangle Matérn GP* $\boldsymbol{f}_2 \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_2)$ *with*

$$\boldsymbol{K}_2 = \Psi_2(\boldsymbol{L}_2) = \left(\frac{2\nu_2}{\kappa_2^2}\boldsymbol{I} + \boldsymbol{L}_2\right)^{-\nu_2}. \tag{B.51}$$

Then, Corollary B.10 gives us its curl adjoint as a curl edge GP $\boldsymbol{f}_C \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}_C)$ with

$$\boldsymbol{K}_C = \boldsymbol{L}_\mathrm{u}\Big(\frac{2\nu_2}{\kappa_2^2}\boldsymbol{I} + \boldsymbol{L}_\mathrm{u}\Big)^{-\nu_2}. \tag{B.52}$$

Furthermore, the curl GP $\boldsymbol{f}_C$ is the solution of the following SPDE

$$\Big(\frac{2\nu_2}{\kappa_2^2}\boldsymbol{I} + \boldsymbol{L}_\mathrm{u}\Big)^{\frac{\nu_2}{2}} \boldsymbol{f}_C = \boldsymbol{B}_2\boldsymbol{w}_2 \tag{B.53}$$

where $\boldsymbol{w}_2$ is a standard Gaussian noise on triangles following $\boldsymbol{f}_2 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

Proof. The proof can follow the same procedure as above for Corollary B.13. □

C EXPERIMENTS

Here we provide additional details on the three experiments presented in the main text.

Experimental Setup   In our three experiments we consider the regression tasks and implement GP regression using the GPyTorch library (Gardner et al., 2018). We optimize the marginal log likelihood loss for 1000 iterations with the ADAM optimizer where the learning rate is set to the default value of 0.001. We run each experiment 10 times with hyperparameters randomly initialized. We report evaluation metrics averaged over 10 experiments and the respective standard deviations. All experiments are run on a NVIDIA GeForce RTX 3080 GPU with 10GB of memory.

Line-graph Construction   Given the incidence matrix $\boldsymbol{B}_1$ of the original graph, the adjacency matrix and the corresponding graph Laplacian of the line-graph can be found as $\boldsymbol{A}_{lg} := |\boldsymbol{B}_1^\top \boldsymbol{B}_1 - 2\boldsymbol{I}|$ and $\boldsymbol{L}_{lg} = \mathrm{diag}(\boldsymbol{A}_{lg}\boldsymbol{1}) - \boldsymbol{A}_{lg}$.

C.1   Additional Details for the Forex Experiment

In the forex experiment, we obtain the data from *Foreign Exchange Data* by *Oanda Corporation*[3]. The data was collected at 2018/20/05 17:00 UTC by Jia et al. (2019). It includes the pairwise exchange rates between the 25 most traded currencies, which form 210 exchangeable pairs. With them as nodes and edges, we then construct an unweighted SC_2 by including the triangles formed by any three pairwise exchangeable currencies. For an edge $\{i,j\}$ connecting currencies i, j, we assign its orientation based on an alphabetical order of their currency names, and likewise for a triangle. For each exchangeable pair, we consider the underlying edge flow as $f_1(i,j) = \log r^{i/j}$, translating the arbitrage-free condition to curl-free condition, where $r^{i/j}$ is the midpoint between ask and bid prices. Fig. C.1 shows the prediction RMSEs using different GP models with respect to training ratios from 0.1 to 0.5 with a step 0.05, as well as the learned Matérn kernels.

For visualizing the predictions using different models, we consider a smaller market for better visibility where we first randomly removed seven currencies then half of the exchangeable pairs, resulting 18 currencies and 77 pairs, as shown in Fig. C.2.

C.2   Additional Details for Ocean Current Analysis

In the second experiment, we consider the ocean drifter data, also known as *Global Lagrangian Drifter Data*, which was collected by NOAA Atlantic Oceanographic and Meteorological Laboratory[4]. Each point in the dataset is a buoy at a specific time, with buoy ID, location (in latitude and longitude), date/time, velocity and water temperature. We consider the buoys that were in the North Pacific ocean dated from 2010 to 2019 with a size of around three million. The dataset itself is a 3D point cloud after converting the location to the *earth-centered, earth-fixed* (ECEF) coordinate system. We follow the procedure in Chen & Meila (2021) to first sample 1,500 buoys furthest from each other, then construct a weighted SC_2 as a Vietoris-Rips (VR) complex with N_1 around 20k and N_2 around 90k. We then convert the velocity field into flows on the edges of SC_2 by using the linear

---

[3]https://www.oanda.com/.
[4]http://www.aoml.noaa.gov/envids/gld/.

(a) RMSEs versus training ratios

(b) RMSEs versus training ratios
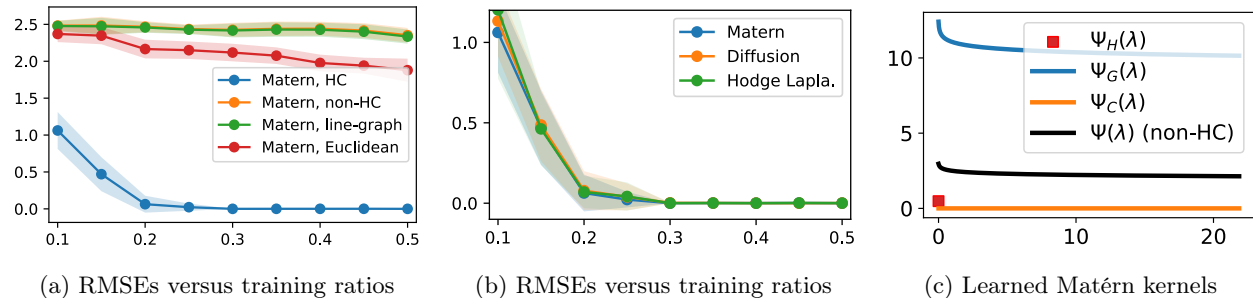
(c) Learned Matérn kernels

Figure C.1: (a) Forex prediction RMSEs of different GPs using Matérn kernels with respect to training ratios. (b) Forex prediction RMSEs of HC GPs using different edge kernels with respect to training ratios. (c) Learned HC and non-HC Matérn kernels in the spectrum for a training ratio of 0.2.

(a) Forex market SC_2

(b) Ground truth

(c) HC Matérn, posterior mean

(d) non-HC Matérn, posterior mean

(e) HC Matérn, posterior std
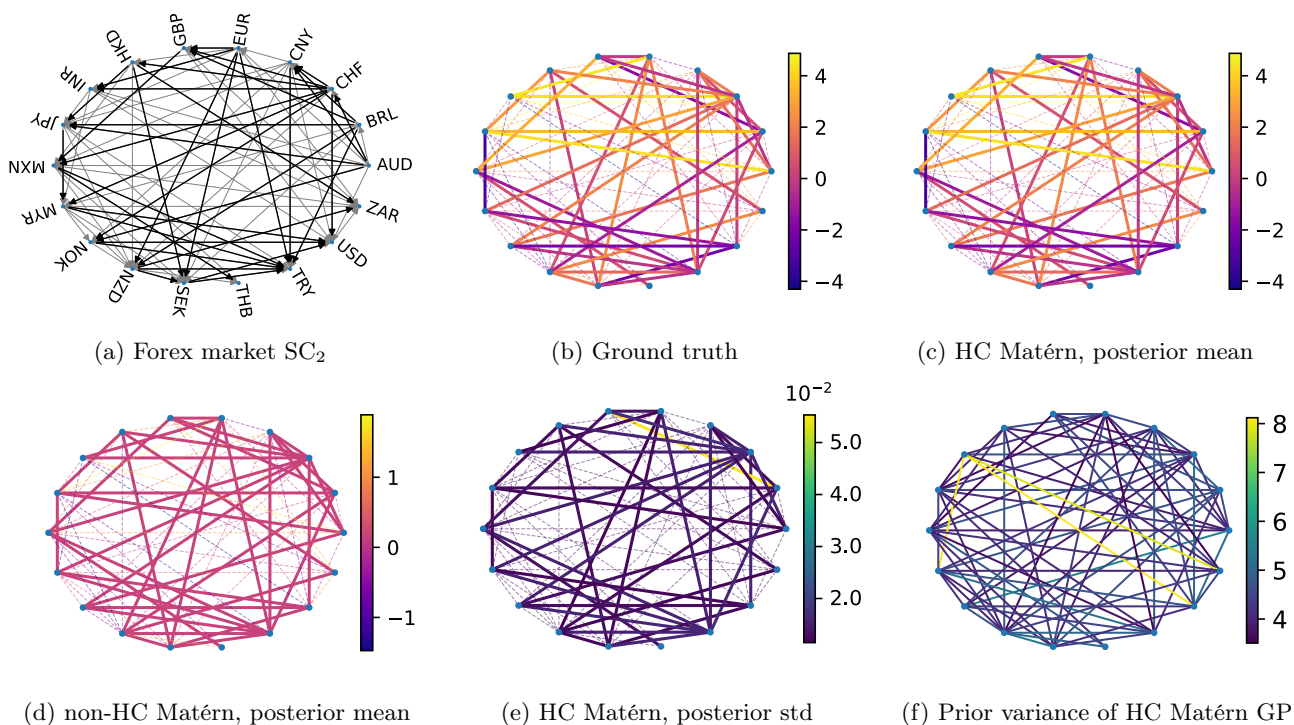
(f) Prior variance of HC Matérn GP

Figure C.2: (a-e): Visualization of forex rates predictions in a smaller market. (f): Prior variance of the learned HC Matérn GP. Note that (b-d) and (f) are the same as the ones in the main content [cf. Fig. 3]. Here we show them with a better resolution.

integration approximation (Chen et al., 2021). We randomly sample 20% of the edges for training and test on the rest. To efficiently construct the edge kernels, we use eigensolver in Knyazev (2001), implemented using the `megaman` library (McQueen et al., 2016), to compute the eigenpairs associated to the 500 largest eigenvalues. We evaluate the prediction mean and uncertainty in the edge flow domain, reported in Table C.1. Furthermore, we obtain the gradient and curl components of the edge flow of the prediction as in Appendix B.5. We visualize the predictions in the edge flow domain in Fig. C.3. We see that both HC and non-HC edge diffusion GPs give close performance and they capture the general pattern of the edge flow. Moreover, the standard deviation is small in most of the locations except few locations (small islands around the lower left corner) where the edge flows (velocity fields) exhibit more discontinuities due to the boundary.

We further convert the edge flows back into vector fields, as shown in Fig. C.4. We refer to Chen et al. (2021) for this procedure. We also approximate the standard deviation of the velocity field prediction by sampling 50 edge flows from the posterior distribution and converting them to the vector field domain, followed by computing the

Table C.1: Ocean current inference results.

Method	RMSE			NLPD		
	Diffusion	Matérn	Hodge Laplacian	Diffusion	Matérn	Hodge Laplacian
Euclidean	1.00 ± 0.01	1.00 ± 0.00	—	1.42 ± 0.01	1.42 ± 0.10	—
Line-Graph	0.99 ± 0.00	0.99 ± 0.00	—	1.41 ± 0.00	1.41 ± 0.00	—
Non-HC	0.35 ± 0.00	0.35 ± 0.00	0.35 ± 0.00	0.33 ± 0.00	0.36 ± 0.03	0.33 ± 0.01
HC	0.34 ± 0.00	0.35 ± 0.00	0.35 ± 0.00	0.33 ± 0.01	0.37 ± 0.04	0.33 ± 0.01

average ℓ_2 distance between the samples and the mean per location, as shown in Fig. C.4d.

C.3 Additional Details for Water Supply Networks

We obtain the Zhi Jiang WSN from Dandy (2016) which contains 114 nodes (113 tanks and 1 source reservoir) and 164 edges (water pipes), no triangles considered. We build an unweighted graph based on the topology of this WSN. We model the hydraulic heads as functions on nodes $\boldsymbol{f}_0$ and water flowrates as functions on edges $\boldsymbol{f}_1$. A WSN is often governed by the following equations

$$\text{mass conservation}: \boldsymbol{B}_1\boldsymbol{f}_1 = \boldsymbol{q}, \text{ and Hazen-Williams equation}: [\boldsymbol{B}_1^\top \boldsymbol{f}_0](e) = \bar{\boldsymbol{f}}_1(e) := r_e f_1(e)^{1.852} \qquad (\text{C.1})$$

for a pipe e, where $\boldsymbol{q} \in \mathbb{R}^{N_0}$ is the demand on nodes, r_e is the roughness of pipe e (Dini & Tabesh, 2014). We then use the WNTR library (Klise et al., 2017) to simulate a scenario generating the states of node heads and edge flowrates given the pipe roughnesses and the node demands. The latter are sampled uniformly from 0 to 10 (unit liter/s), modeling the read-world demand.

We consider the joint state estimation of both heads (using node GPs) and the adjusted flowrates $\bar{\boldsymbol{f}}_1$ (using edge GPs). Specifically, our GP models are

$$\begin{pmatrix} \boldsymbol{f}_0 \\ \bar{\boldsymbol{f}}_1 \end{pmatrix} \sim \mathcal{GP}\left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_0 & \\ & \boldsymbol{K}_1 \end{pmatrix} \right). \qquad (\text{C.2})$$

We choose the Matérn and diffusion node GPs (Borovitskiy et al., 2021). For HC edge GPs, we leverage the physical prior to model $\boldsymbol{K}_1 = \boldsymbol{B}_1^\top \boldsymbol{K}_0 \boldsymbol{B}_1$ as discussed in Corollary 6, while for non-HC edge GPs, we choose them as in Eq. (6), of the same type as node GPs. We randomly sample 50% of the nodes and edges for training and use the rest for test. Note that the WSN has small edge connectivity. The randomness of the training set may disconnect the graph, which may deteriorate the performance, causing the large variance in the metrics.

(a) Original ocean current

(b) Predicted ocean current

(c) Standard deviation

(d) Predicted ocean current, non-HC diffusion

(e) Original gradient ocean current

(f) Original curl ocean current

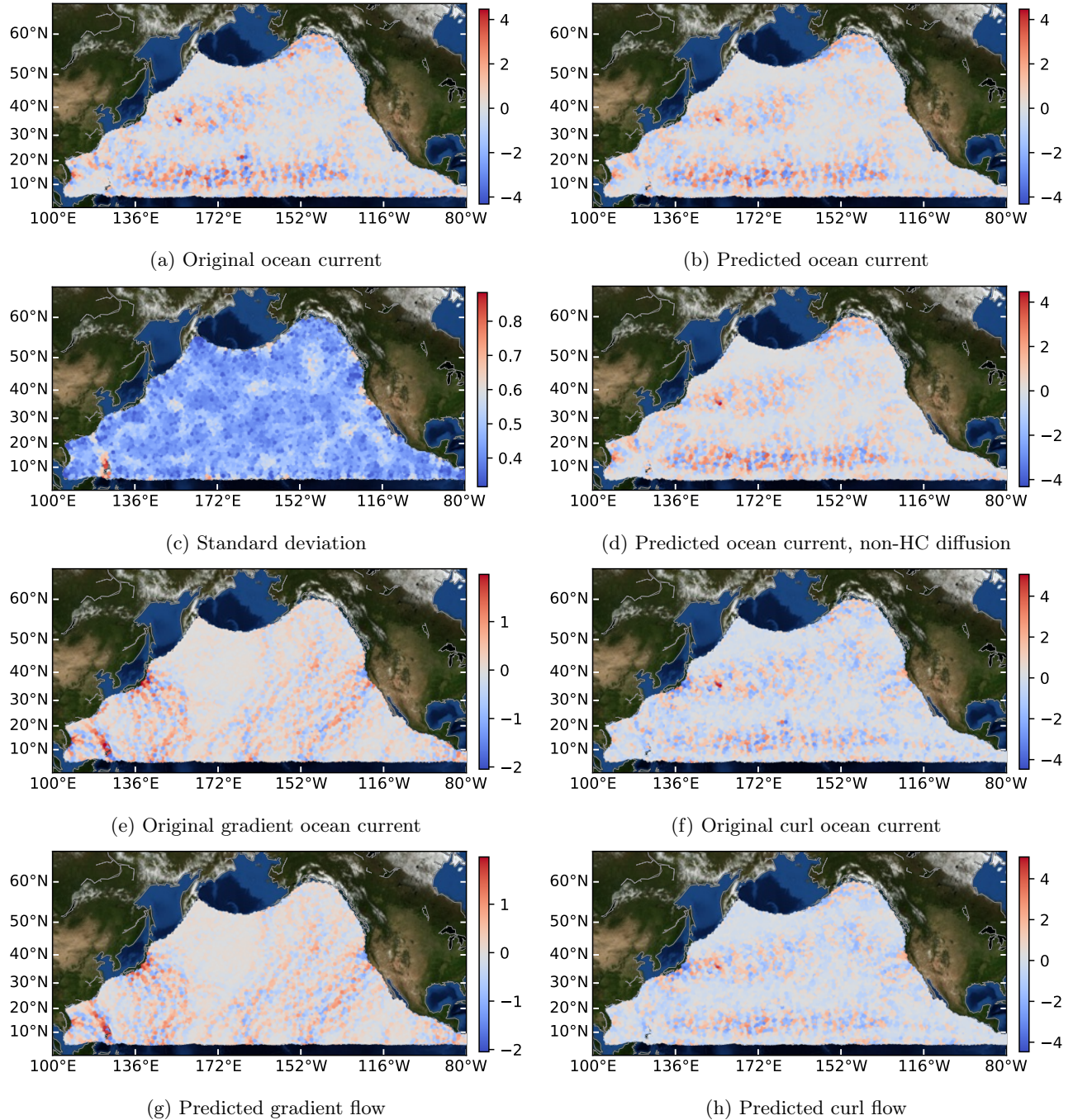(g) Predicted gradient flow

(h) Predicted curl flow

Figure C.3: (a-h) Results for ocean current prediction with 20% training ratio in the edge flow domain. Note that we highlight the edge flow values on the middle points of the edges.

(a) Original ocean current

(b) Posterior mean

(c) Sample from posterior distribution

(d) Standard deviation approximated using 50 samples

(e) Original curl-free component

(f) Original div-free component

(g) Predicted curl-free component

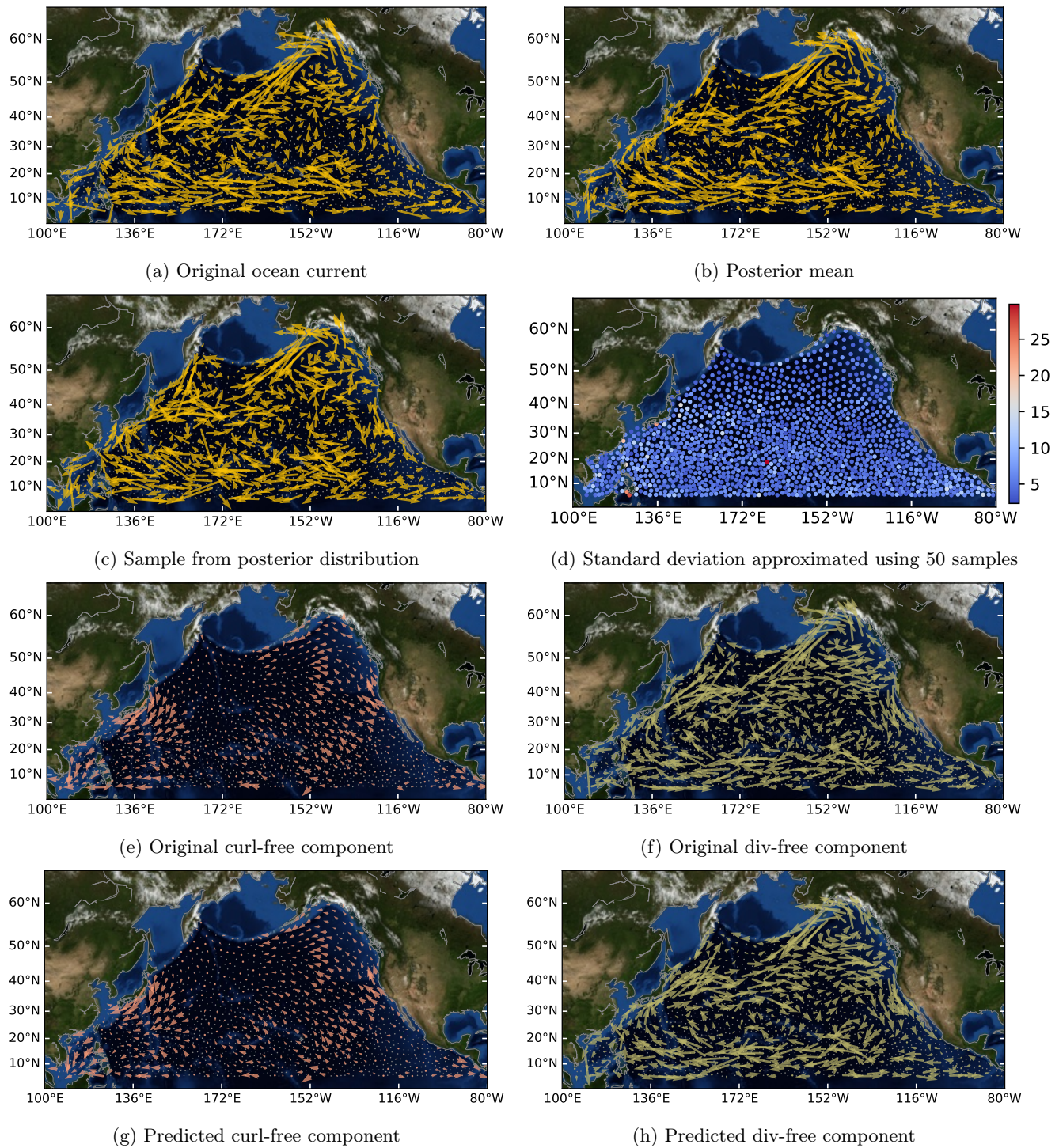(h) Predicted div-free component

Figure C.4: (a-h) Results for ocean current prediction with 20% training ratio in the vector field domain. Note that (a-b) and (g-h) are the same as in Fig. 4. We show them here for reader's convenience.