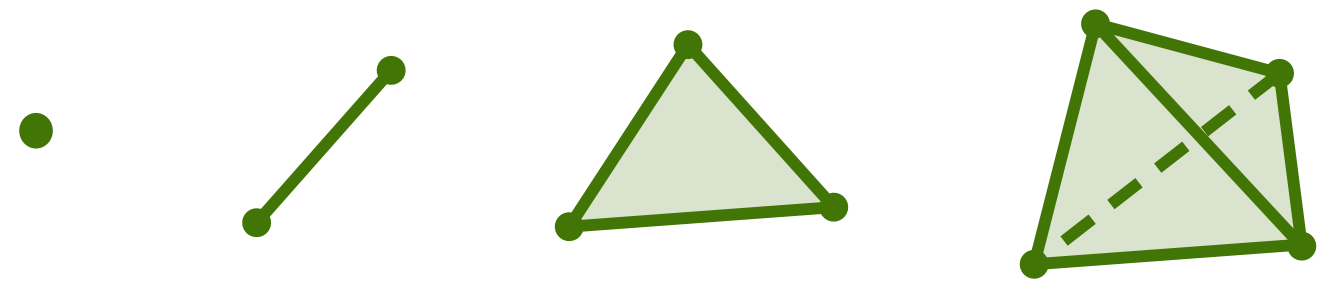
Hodge-Compositional Edge Gaussian Processes

- Edge flow; difference vs. node data; graph vs. simplicial complex
- Smoothness of edge flows: div and curl; Hodge decomposition
- GP modeling of edge functions: div-free, curl-free kernels …

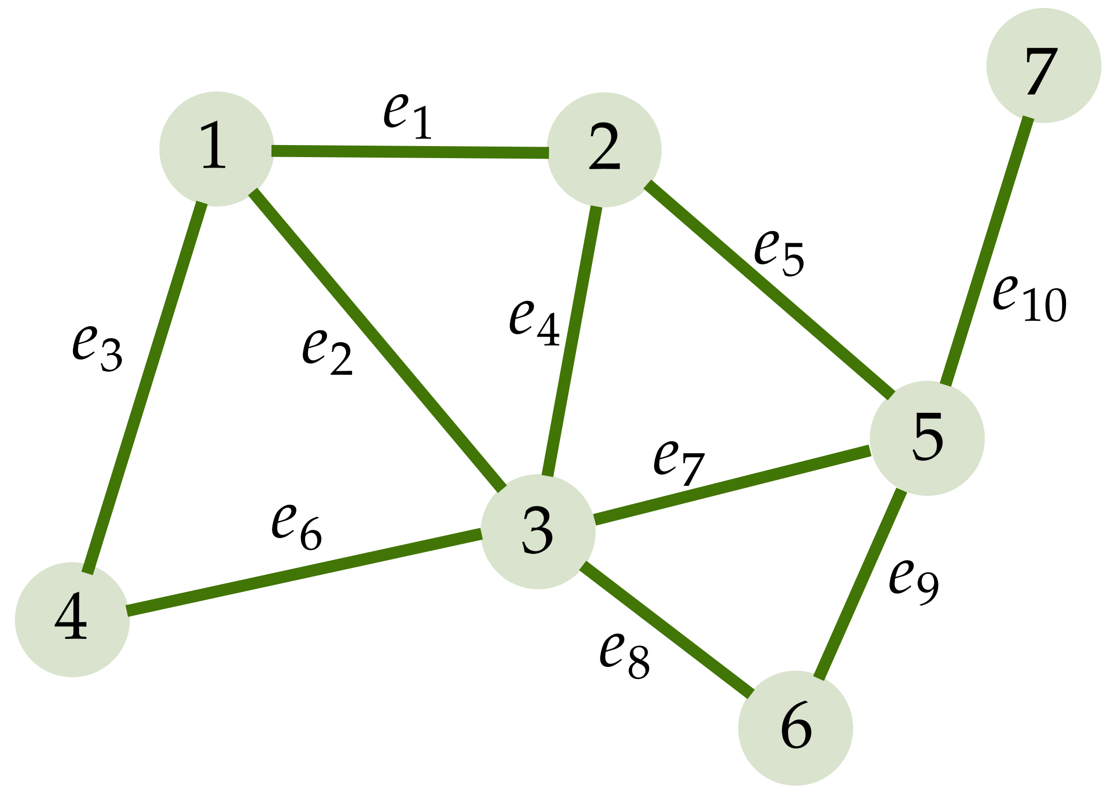
AISTATS 2024

Maosheng Yang (TU Delft),
Viacheslav Borovitskiy (ETH Zürich)
Elvin Isufi (TU Delft)

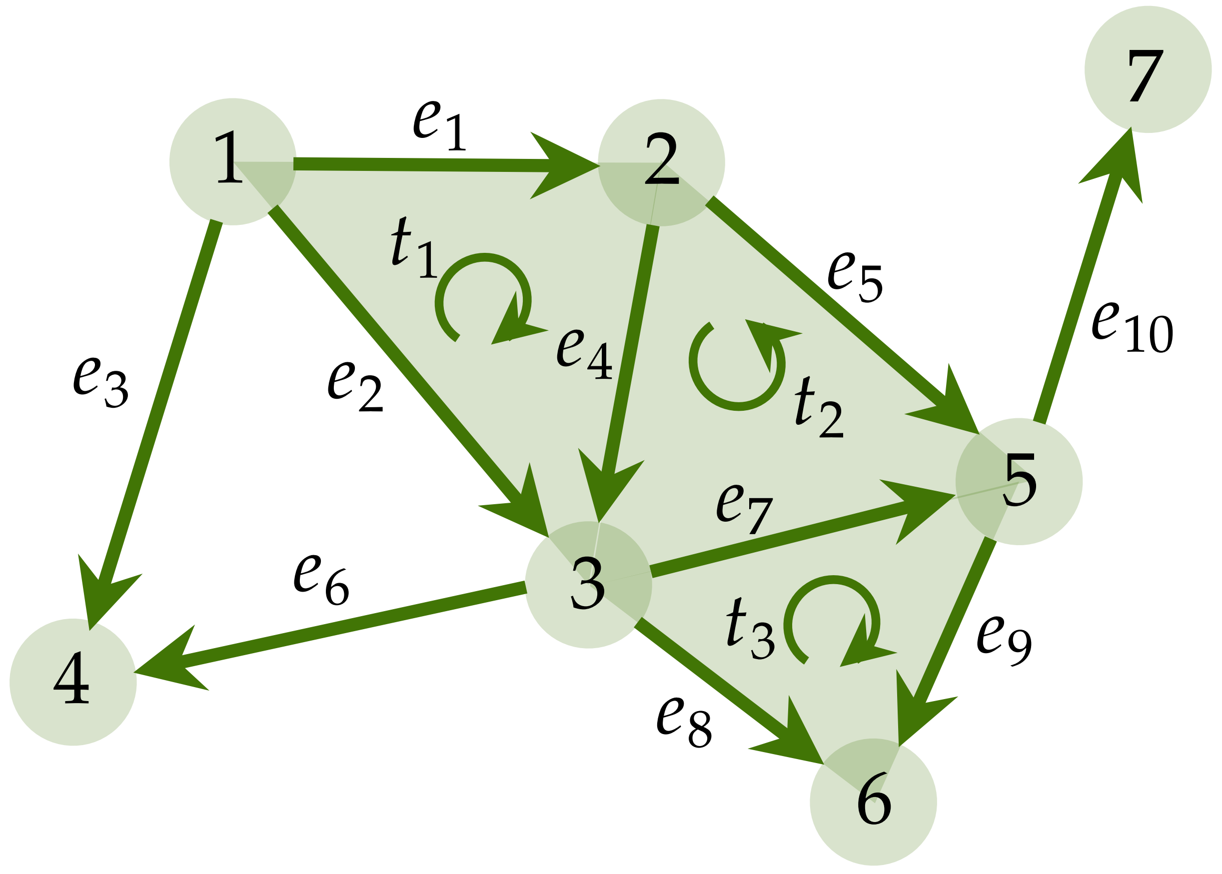
Graphs vs Simplicial 2-Complexes



0-, 1-, 2-, 3-simplices



Graph
Simplicial 1-complex
$G = (V, E)$



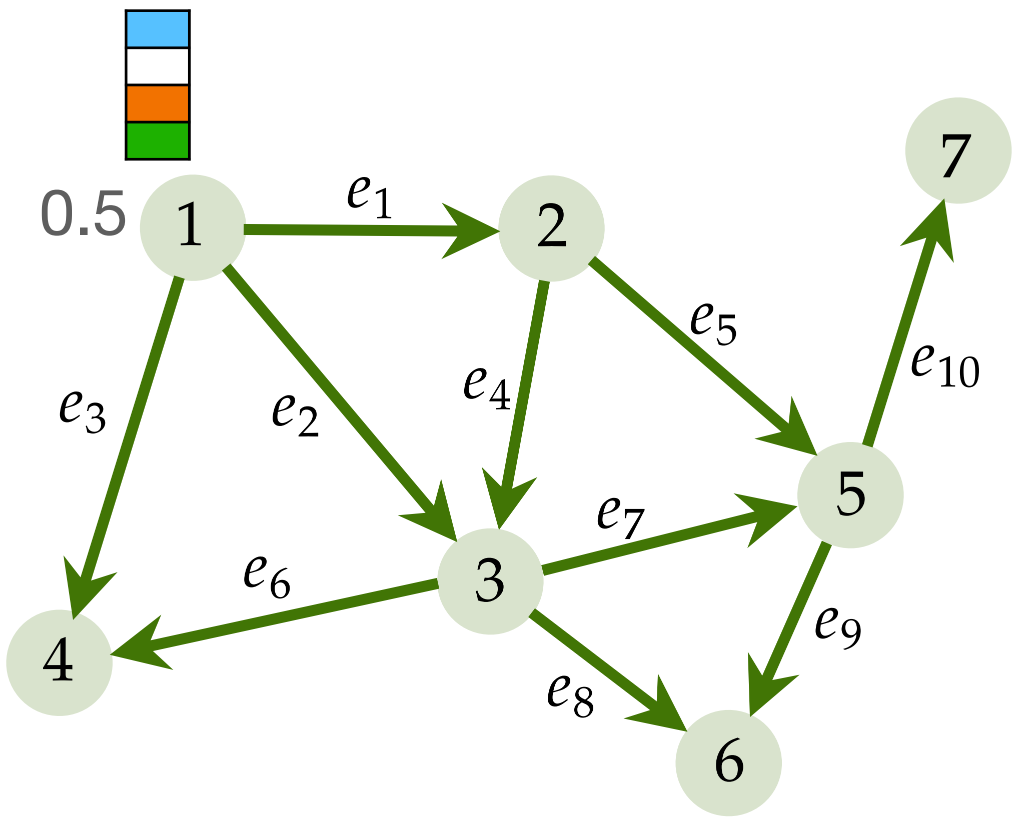
Simplicial 2-complex
$\mathrm{SC}_2 = (V, E, T)$

- Oriented simplices (equivalence class of permutations)

Where are SCs used?
- Network analysis
- Topological data analysis
- Topological signal processing
- Topological deep learning
- Numerical methods
- Computer graphics
- …

Functions on simplices

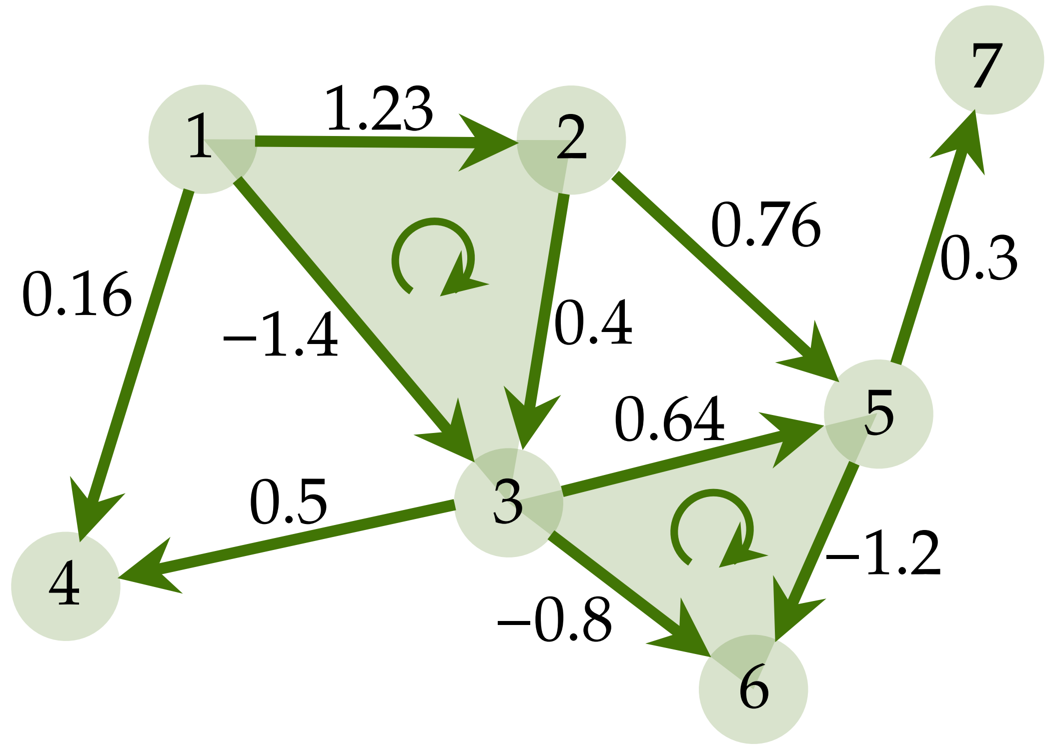
Signals on nodes, edges, triangles, …



Node function

$$f_0 : V \to \mathbb{R}$$

$$\mathbf{f}_0 = (f_0(1), \ldots, f_0(N_0))^\top$$



Edge function

$$f_1 : E \to \mathbb{R}$$

$$\mathbf{f}_1 = (f_1(e_1), \ldots, f_1(e_{N_1}))^\top$$

- Alternating property
- Magnitude and sign

- Flow-type data (natural)
  - Physical world: traffic flow, water flow, information flow…
  - Forex: exchange rates
  - Game theory (Candogan et al. 2011)
  - Ranking data (Jiang et al. 2011)
  - Edge-based vector field discretisation (computer graphics)
  - …

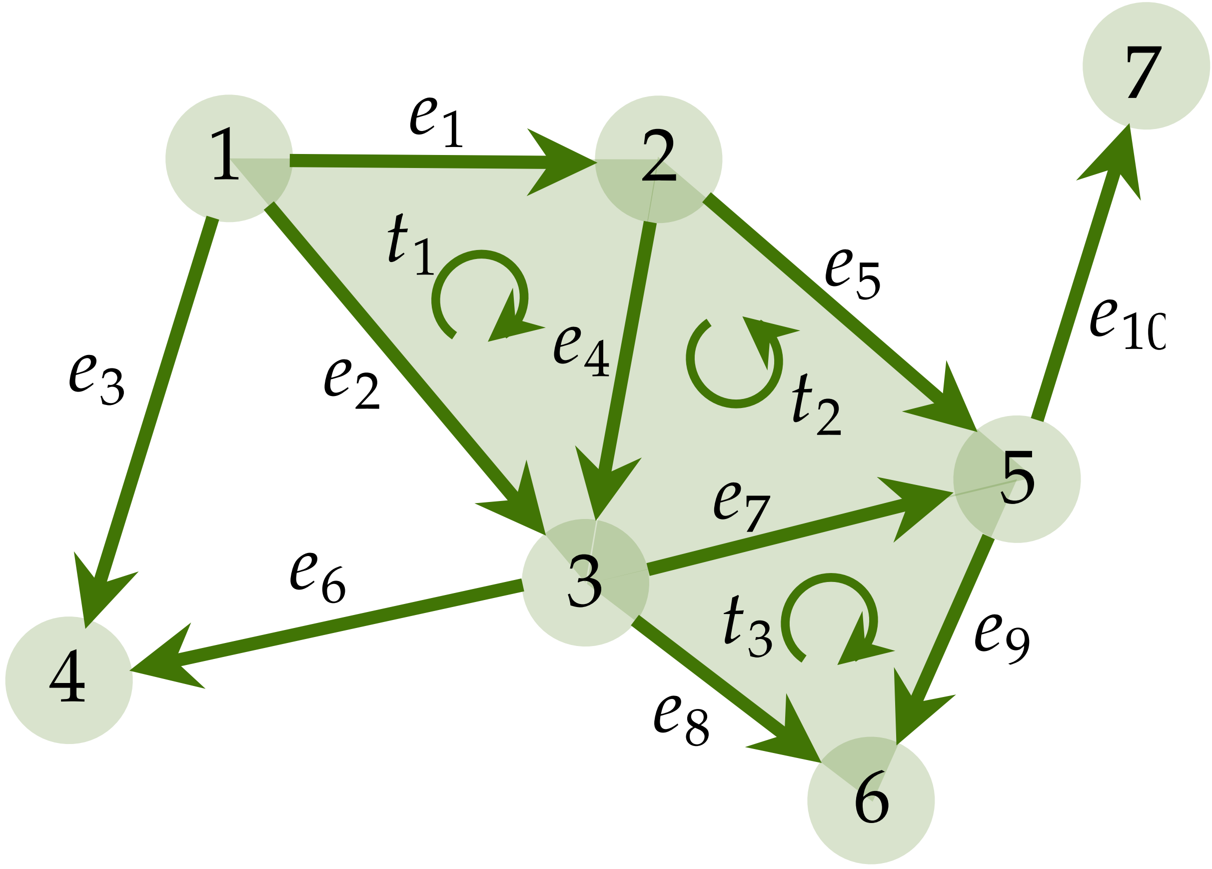
Triangle function

$$f_2 : T \to \mathbb{R}$$

0-, 1-, 2-cochains in topology

Algebraic reps. of simplicial 2-complex

Incidences & Laplacians



Node-to-Edge

Edge-to-Faces

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

Graph Laplacian: $\mathbf{L}_0 = \mathbf{B}_1 \mathbf{B}_1^\top$

1-Hodge Laplacian: $\mathbf{L}_1 = \mathbf{B}_1^\top \mathbf{B}_1 + \mathbf{B}_2 \mathbf{B}_2^\top := \mathbf{L}_{1,d} + \mathbf{L}_{1,u}$

Up

Down

GPs on graphs

Modeling node functions

- $\mathbf{f}_0 \sim \mathrm{GP}(\mathbf{0}, \mathbf{K}_0)$ (Borovitskiy et al. 2021)
- Matérn graph kernel

$$\Phi(\mathbf{L}_0)\mathbf{f}_0 = \mathbf{w}_0, \text{ with}$$

$$\Phi(\mathbf{L}_0) = \left(\frac{2\nu}{\kappa^2}\mathbf{I} + \mathbf{L}_0\right)^{\frac{\nu}{2}} \text{ and } \mathbf{w}_0 \sim N(\mathbf{0}, \sigma^2\mathbf{I})$$

- The solution has kernel

$$\mathbf{K}_0 = \sigma^2 \sum_{n=0}^{N_0-1} \psi(\lambda_n)\mathbf{u}_n\mathbf{u}_n^\top = \sigma^2\left(\frac{2\nu}{\kappa^2}\mathbf{I} + \mathbf{L}_0\right)^{-\nu}$$

$$\psi(\lambda) = \begin{cases} (\frac{2\nu}{\kappa^2} + \lambda)^{-\nu} & \nu < \infty, \text{Matern} \\ e^{-\frac{\kappa^2}{2}\lambda} & \nu = \infty, \text{Diffusion} \end{cases}$$

GPs from Euclidean to non-Euclidean

GP in Euclidean settings

Function on a set $f : X \to \mathbb{R}$

$$f \sim \mathrm{GP}(\mu, k)$$

- Predictive distribution $f_{|\mathbf{y}}$
- Matérn GP family, e.g., diffusion

$$k(x, x') = \sigma^2 \exp\left(-\frac{d(x, x')^2}{2\kappa^2}\right)$$

- Distance-based: geometry-aware, but not well-defined for manifolds, graphs …
- Instead, as solutions of SDEs (Whittle (1963); Lindgren et al. (2011))

$$\left(\frac{2\nu}{\kappa^2} - \Delta\right)^{\frac{\nu}{2}+\frac{d}{4}} f = w$$

- Δ: Laplacian, w: white noise
- implicit, generalizable, domain-aware
- explicit for some domains

Matérn Edge GPs

Derived from SDEs on the edge set

- $\mathbf{f}_1 \sim \mathrm{GP}(\mathbf{0}, \mathbf{K}_1)$

EVD: $\mathbf{L}_1 = \mathbf{U}_1 \mathbf{\Lambda}_1 \mathbf{U}_1^\top$

- Matérn graph kernel

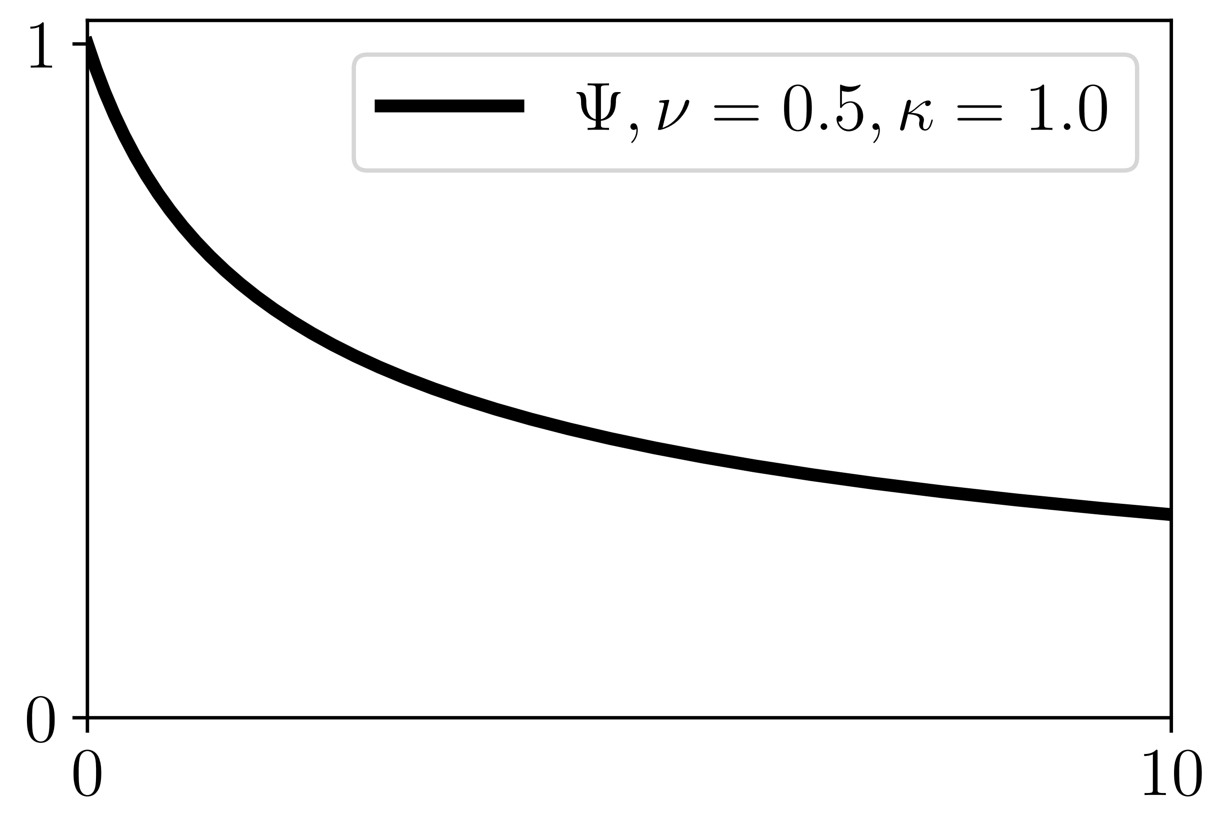
$$\Phi(\mathbf{L}_1)\mathbf{f}_1 = \mathbf{w}_1, \text{ with}$$

$$\Phi(\mathbf{L}_1) = \left(\frac{2\nu}{\kappa^2}\mathbf{I} + \mathbf{L}_1\right)^{\frac{\nu}{2}} \text{ and } \mathbf{w}_1 \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$$

- The solution gives edge GPs

$$\text{Matérn: } \mathbf{f}_1 \sim \mathrm{GP}\left(0, \left(\frac{2\nu}{\kappa^2}\mathbf{I} + \mathbf{L}_1\right)^{-\nu}\right)$$

$$\text{Diffusion: } \mathbf{f}_1 \sim \mathrm{GP}\left(0, e^{-\frac{\kappa^2}{2}\mathbf{L}_1}\right)$$



- Low-pass in the eigen-spectrum

Smoothness

Node function — 0-form (scalar field)

Edge function — 1-form (vector field)

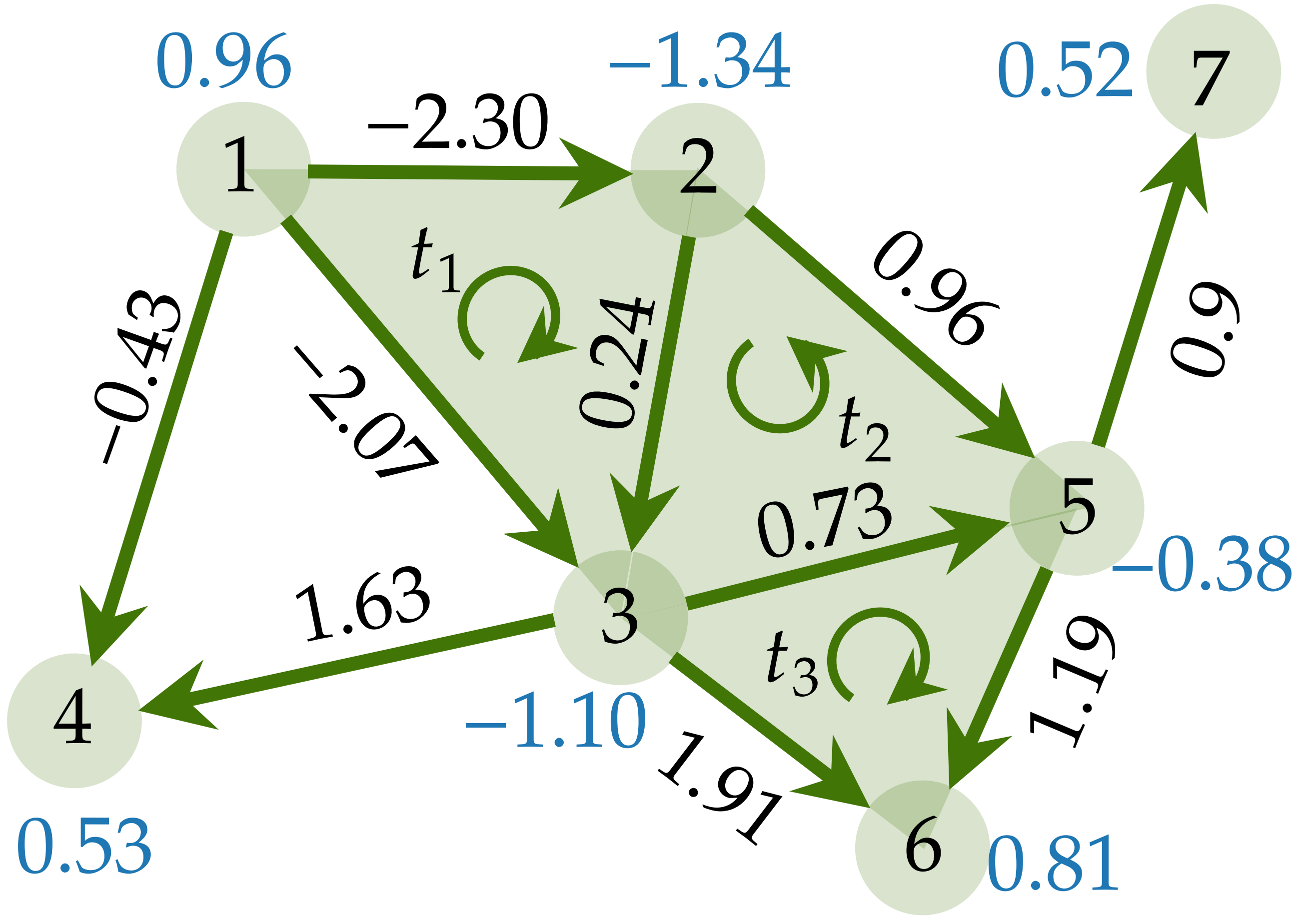
Divergence

Curl

Incidence & Laplacians

1st and 2nd order Discrete Derivatives

- Node signal $\mathbf{v}$
- Edge flows $\mathbf{f}$



Gradient of node signal: $[\mathbf{f}_G]_{[i,j]} = [\mathbf{B}_1^\top \mathbf{v}]_{[i,j]} = [\mathbf{v}]_j - [\mathbf{v}]_i$

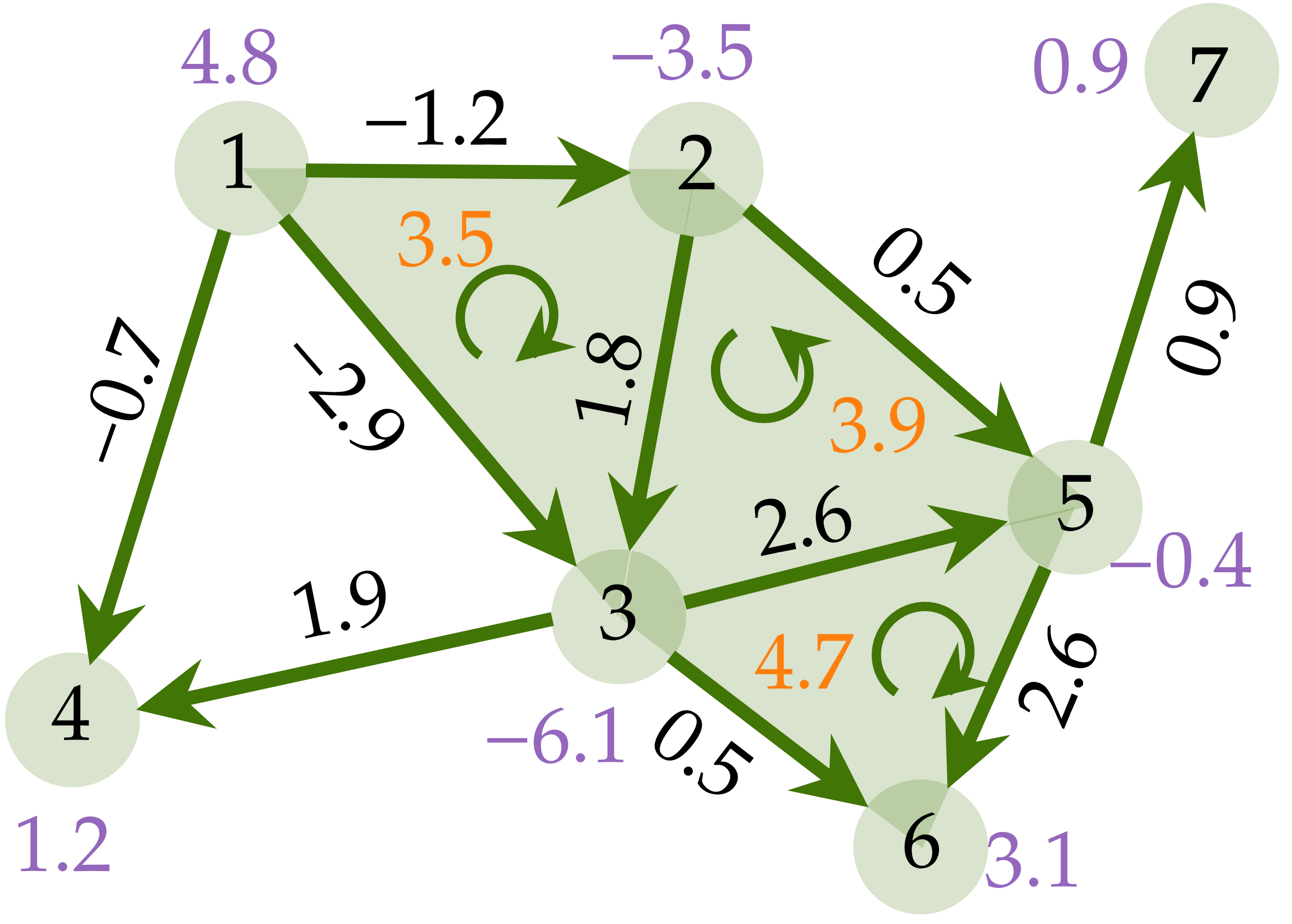
Divergence of edge flows: $[\mathbf{B}_1 \mathbf{f}]_{[i]} = \sum_{j<i} \mathbf{f}_{[j,i]} - \sum_{i<k} \mathbf{f}_{[i,k]}$

Curl of edge flows: $[\mathbf{B}_2^\top \mathbf{f}]_t = \mathbf{f}_{[i,j]} + \mathbf{f}_{[j,k]} - \mathbf{f}_{[i,k]}$, for $t = [i,j,k]$

$$[\mathbf{B}_1^\top \mathbf{v}]_{[1,2]} = -1.34 - 0.96 = -2.30$$

Incidence & Laplacians

1st and 2nd order Discrete Derivatives



Gradient of node signal: $\mathbf{B}_1^{\top}\mathbf{v}$ $\quad [\mathbf{f}_{\mathrm{G}}]_{[i,j]} = [\mathbf{v}]_j - [\mathbf{v}]_i$

Divergence of edge flows: $[\mathbf{B}_1\mathbf{f}]_{[i]} = \sum_{j<i} \mathbf{f}_{[j,i]} - \sum_{i<k} \mathbf{f}_{[i,k]}$

Net-flow = in_flow - out_flow

Curl of edge flows: $[\mathbf{B}_2^{\top}\mathbf{f}]_t = \mathbf{f}_{[i,j]} + \mathbf{f}_{[j,k]} - \mathbf{f}_{[i,k]}$, for $t = [i,j,k]$

Net-circulation in triangles

$$[\mathbf{B}_1\mathbf{f}]_5 = 0.5 + 2.6 - (0.9 + 2.6) = -0.4$$

Laplacians = Grad Div + Curl* Curl

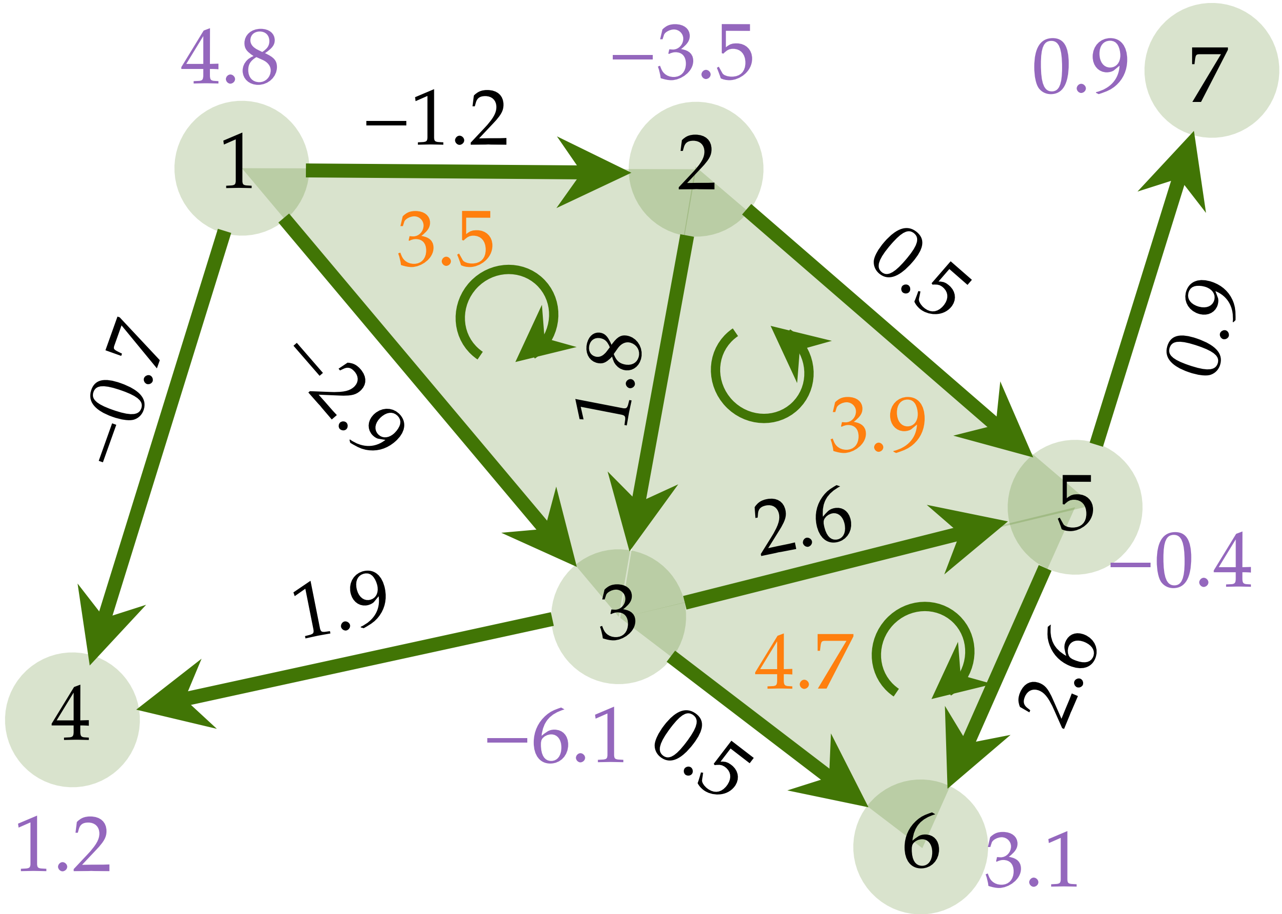
Hodge Laplacian: $\mathbf{L}_1 = \mathbf{B}_1^{\top}\mathbf{B}_1 + \mathbf{B}_2\mathbf{B}_2^{\top}$

$$[\mathbf{B}_2^{\top}\mathbf{f}]_{[1,2,3]} = -1.2 + 1.8 - (-2.9) = 3.5$$

$$\Delta_1 = \nabla(\nabla\cdot) + \nabla\times(\nabla\times)$$

Incidence & Laplacians

1st and 2nd order Discrete Derivatives



Gradient of node signal: $\mathbf{B}_1^\top \mathbf{v}$ $[\mathbf{f}_G]_{[i,j]} = [\mathbf{v}]_j - [\mathbf{v}]_i$

Divergence of edge flows: $[\mathbf{B}_1 \mathbf{f}]_{[i]} = \sum_{j<i} \mathbf{f}_{[j,i]} - \sum_{i<k} \mathbf{f}_{[i,k]}$

Net-flow = in_flow - out_flow

Curl of edge flows: $[\mathbf{B}_2^\top \mathbf{f}]_t = \mathbf{f}_{[i,j]} + \mathbf{f}_{[j,k]} - \mathbf{f}_{[i,k]}$, for $t = [i, j, k]$

Net-circulation in triangles

$$[\mathbf{B}_1 \mathbf{f}]_5 = 0.5 + 2.6 - (0.9 + 2.6) = -0.4$$

Laplacians = Grad Div + Curl* Curl

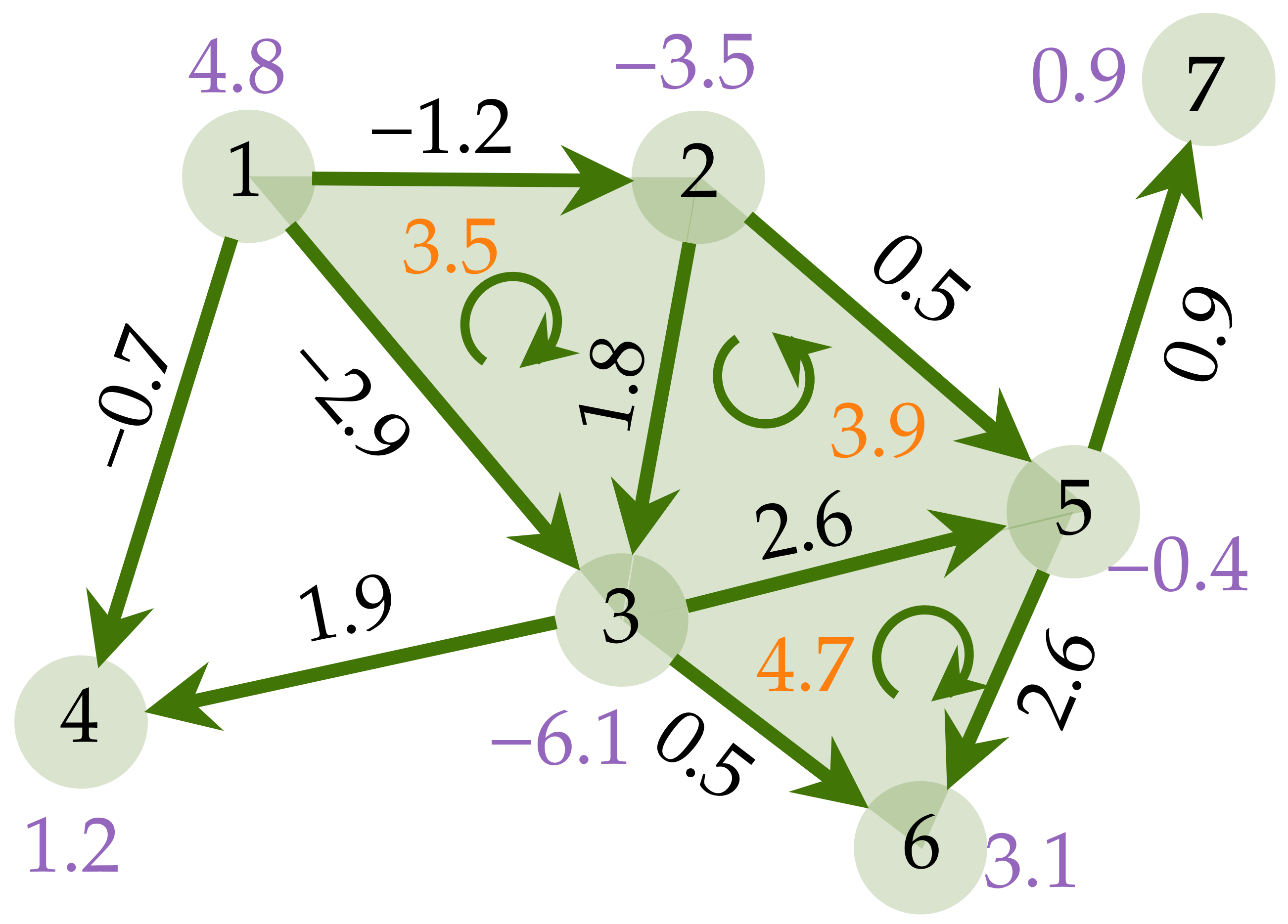
Hodge Laplacian: $\mathbf{L}_1 = \mathbf{B}_1^\top \mathbf{B}_1 + \mathbf{B}_2 \mathbf{B}_2^\top$

$$[\mathbf{B}_2^\top \mathbf{f}]_{[1,2,3]} = -1.2 + 1.8 - (-2.9) = 3.5$$

$$\Delta_1 = \nabla(\nabla \cdot\ ) + \nabla \times (\nabla \times\ )$$

Incidence & Laplacians

1st and 2nd order Discrete Derivatives



Gradient of node signal: $\mathbf{B}_1^\top \mathbf{v}$ $\quad [\mathbf{f}_G]_{[i,j]} = [\mathbf{v}]_j - [\mathbf{v}]_i$

Divergence of edge flows: $[\mathbf{B}_1 \mathbf{f}]_{[i]} = \sum_{j<i} \mathbf{f}_{[j,i]} - \sum_{i<k} \mathbf{f}_{[i,k]}$

Net-flow = in_flow - out_flow

Curl of edge flows: $[\mathbf{B}_2^\top \mathbf{f}]_t = \mathbf{f}_{[i,j]} + \mathbf{f}_{[j,k]} - \mathbf{f}_{[i,k]}$, for $t = [i, j, k]$

Net-circulation in triangles

$$[\mathbf{B}_1 \mathbf{f}]_5 = 0.5 + 2.6 - (0.9 + 2.6) = -0.4$$

$$[\mathbf{B}_2^\top \mathbf{f}]_{[1,2,3]} = -1.2 + 1.8 - (-2.9) = 3.5$$

Hodge Laplacians = Grad Div + Curl* Curl

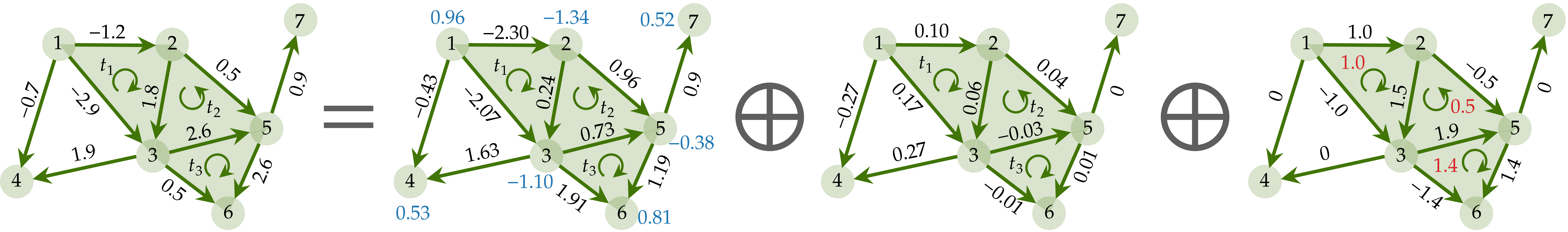
Hodge Laplacian: $\mathbf{L}_1 = \mathbf{B}_1^\top \mathbf{B}_1 + \mathbf{B}_2 \mathbf{B}_2^\top$

Hodge decomposition

Lovász et al. 2004; Lim et al. 2020

$$\mathbb{R}^{N_1} = \mathrm{im}(\mathbf{B}_1^\top) \oplus \ker(\mathbf{L}_1) \oplus \mathrm{im}(\mathbf{B}_2)$$

$$\mathbf{f}_1 = \mathbf{f}_G + \mathbf{f}_H + \mathbf{f}_C$$

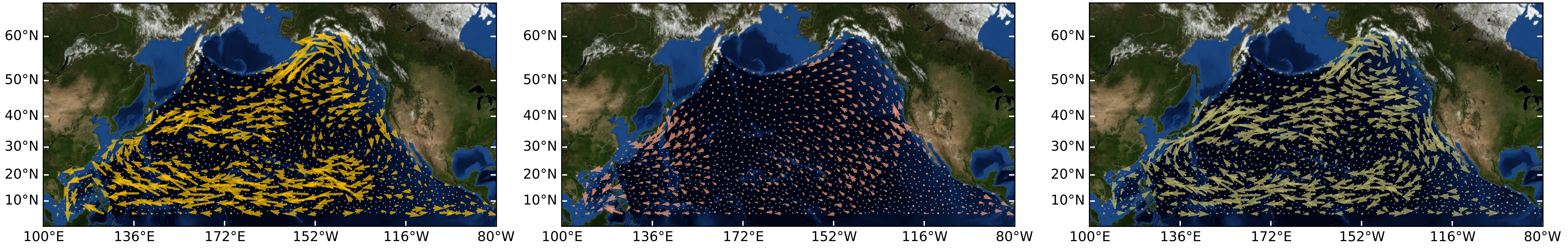


Edge flow = Gradient flow (Curl-free, irrotational) + Harmonic flow (Div- and curl-free) + Curl flow (Div-free, solenoidal)

Hodge-compositional Edge GP

$$\mathbf{f}_G \sim \mathrm{GP}(\mathbf{0}, \mathbf{K}_G)$$
$$\mathbf{f}_H \sim \mathrm{GP}(\mathbf{0}, \mathbf{K}_H)$$
$$\mathbf{f}_C \sim \mathrm{GP}(\mathbf{0}, \mathbf{K}_C)$$

Applications of Hodge decomposition



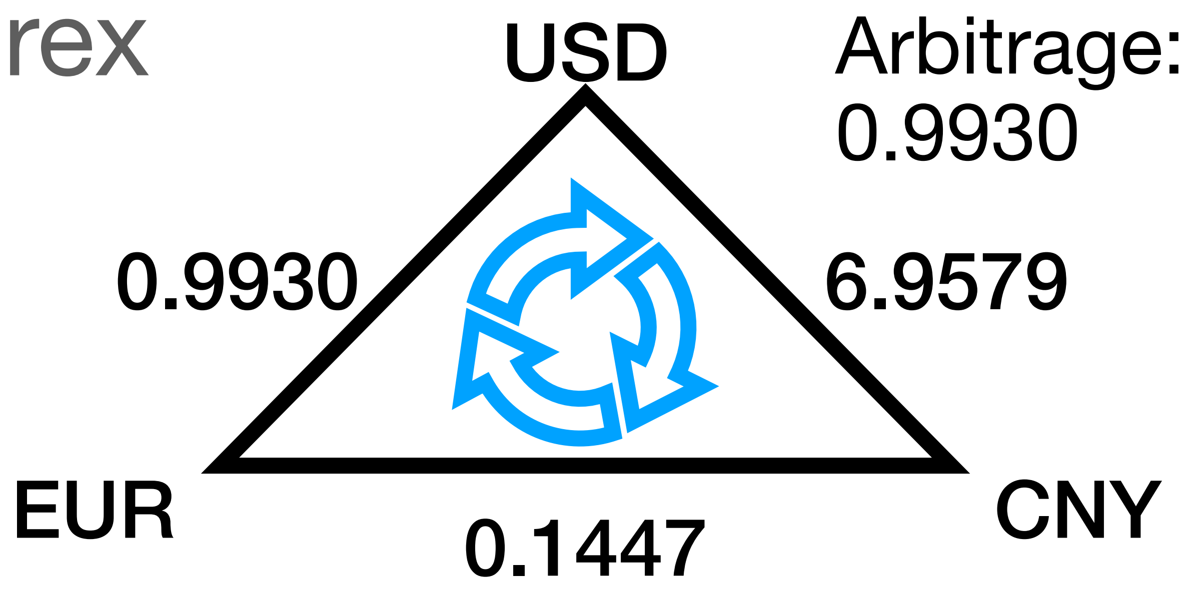
Ocean currents

Gradient flow
Curl-free, irrotational

Curl flow
Div-free, solenoidal

Chen, Yu-Chia et al. (2021) "Helmholtzian Eigenmap."

Forex



$r^{a/b} r^{b/c} = r^{a/c}$ Arbitrage-free

$f_{[a,b]} + f_{[b,c]} - f_{[a,c]} = 0$ Curl-free

- Water flows (div-free)
- Electrical currents (KCL), voltages (KVL)

- Brain networks (Anand et al. 2022)
- Game theory (Candogan et al. 2011)
- Ranking problems (Jiang et al. 2011)
- Random walks (Strang et al. 2020)
- …

Eigenspace of $\mathbf{L}_1$ spans Hodge subspaces

- **Nonzero** Eigenspace of down Laplacian spans the gradient space
- **Nonzero** Eigenspace of up Laplacian spans the curl space
- **Zero** Eigenspace of Laplacian spans the harmonic space

Simplicial Fourier transform

Frequency — eigenvalues

Fourier basis — eigenvectors

$\lambda_G = \|\mathbf{B}_1 \mathbf{u}_G\|_2^2$

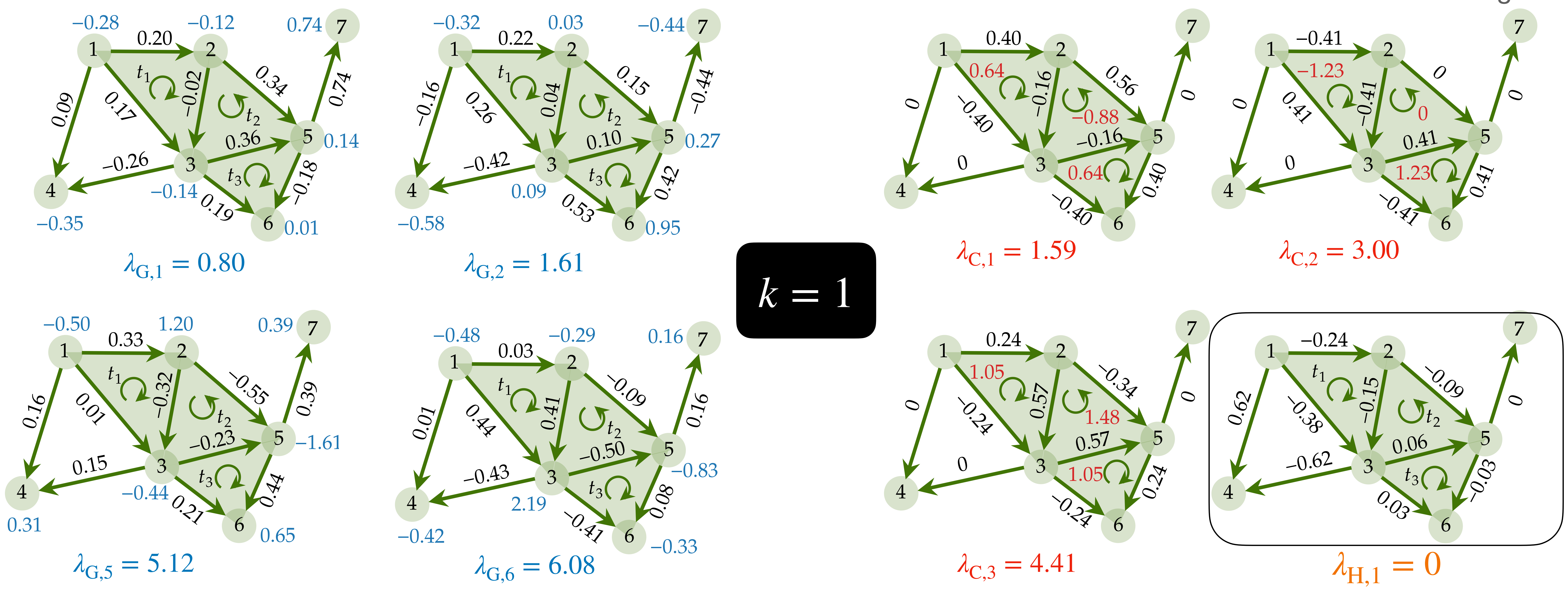
Gradient eigenvector

Fourier basis reflecting divergent properties

$\lambda_C = \|\mathbf{B}_2^\top \mathbf{u}_C\|_2^2$

Curl eigenvector

Fourier basis reflecting rotational properties



EVD: $\mathbf{L}_1 = \mathbf{U}_1 \mathbf{\Lambda}_1 \mathbf{U}_1^\top$

$\mathbf{U}_1 = [\mathbf{U}_H \ \mathbf{U}_G \ \mathbf{U}_C]$

$\text{span}(\mathbf{U}_H) = \ker(\mathbf{L}_1)$

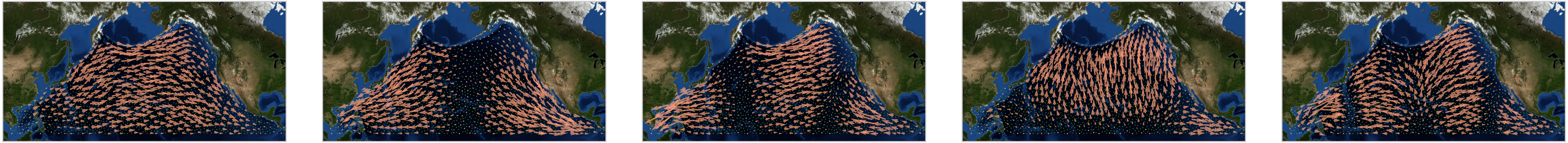
$\text{span}(\mathbf{U}_G) = \text{im}(\mathbf{B}_1^\top)$

$\text{span}(\mathbf{U}_C) = \text{im}(\mathbf{B}_2)$

Yang et al. Simplicial Convolutional Filters, 2022

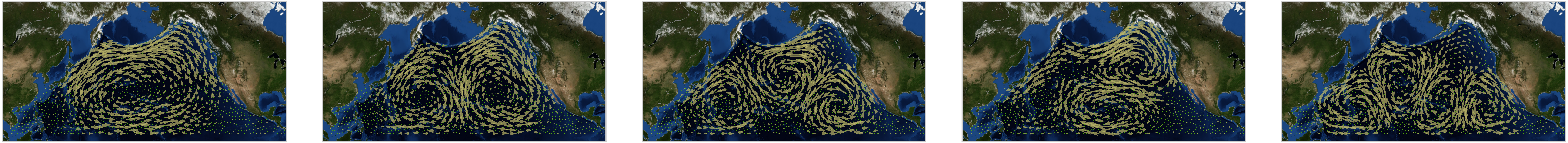
Eigenspace spans Hodge subspaces

- Down Laplacian, its nonzero eigenspace spans the gradient space



λ_G, more divergent

- Up laplacian, its nonzero eigenspace spans the curl space



λ_C, more rotational

Hodge-compositional Edge GPs

Curl-free, div-free GPs

$\mathbf{f}_G \sim \mathrm{GP}(\mathbf{0}, \mathbf{K}_G)$
$\mathbf{f}_H \sim \mathrm{GP}(\mathbf{0}, \mathbf{K}_H)$
$\mathbf{f}_C \sim \mathrm{GP}(\mathbf{0}, \mathbf{K}_C)$

- Gradient kernel $\mathbf{K}_G = \mathbf{U}_G \Psi_G(\mathbf{\Lambda}_G) \mathbf{U}_G^\top$; Curl kernel $\mathbf{K}_C = \mathbf{U}_C \Psi_C(\mathbf{\Lambda}_C) \mathbf{U}_C^\top$
- Matérn family: $\Psi_\square(\mathbf{\Lambda}_\square) = \sigma_\square^2 \left(\frac{2\nu_\square}{\kappa_\square^2} \mathbf{I} + \mathbf{\Lambda}_\square \right)^{-\nu_\square}, \quad \square = \mathrm{H}, \mathrm{G}, \mathrm{C}$
- Also as solutions of SDEs, e.g.,

$\Phi_C(\mathbf{L}_{1,\mathrm{u}})\, \mathbf{f}_1 = \mathbf{w}_C$, with curl noise $\mathbf{w}_C \sim N(0, \sigma_C^2 \mathbf{U}_C \mathbf{U}_C^\top)$ and

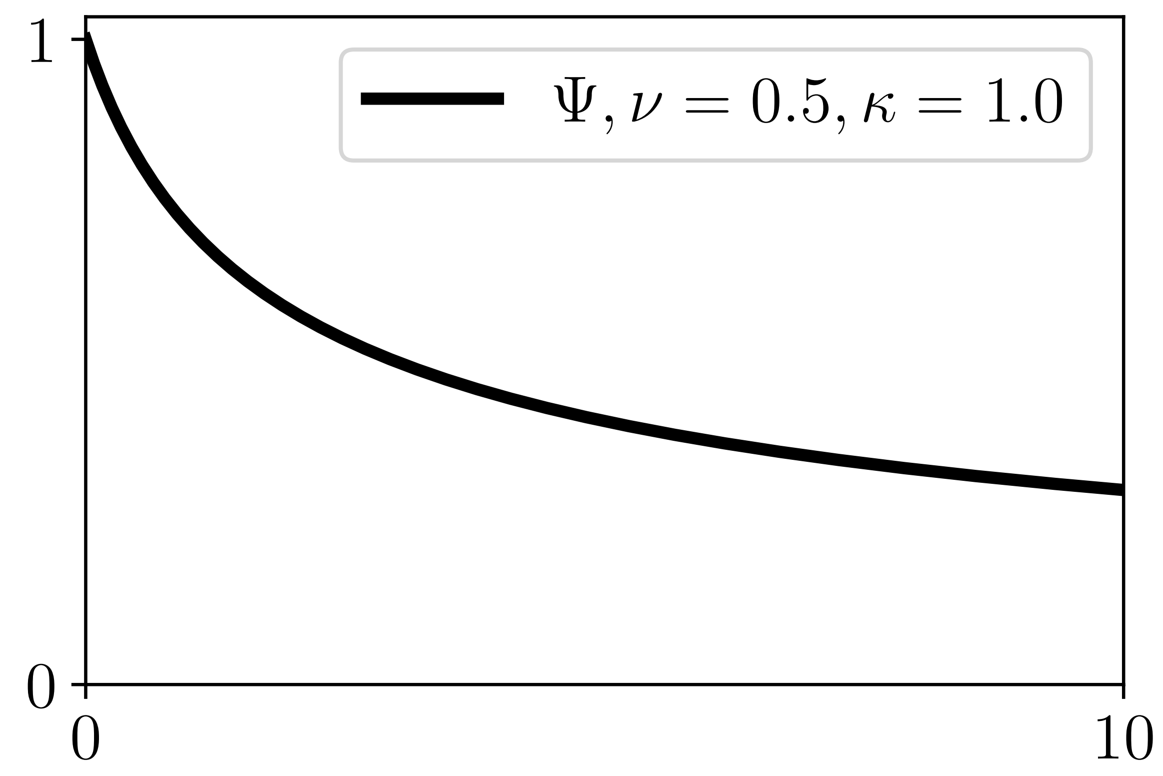
$$\Phi(\mathbf{L}_{1,u}) = \left(\frac{2\nu_C}{\kappa_C^2} \mathbf{I} + \mathbf{L}_{1,u} \right)^{\frac{\nu_C}{2}} \quad \text{or} \quad \Phi(\mathbf{L}_{1,u}) = e^{\frac{\kappa_C^2}{4} \mathbf{L}_{1,u}}$$

Hodge-compositional Edge GPs

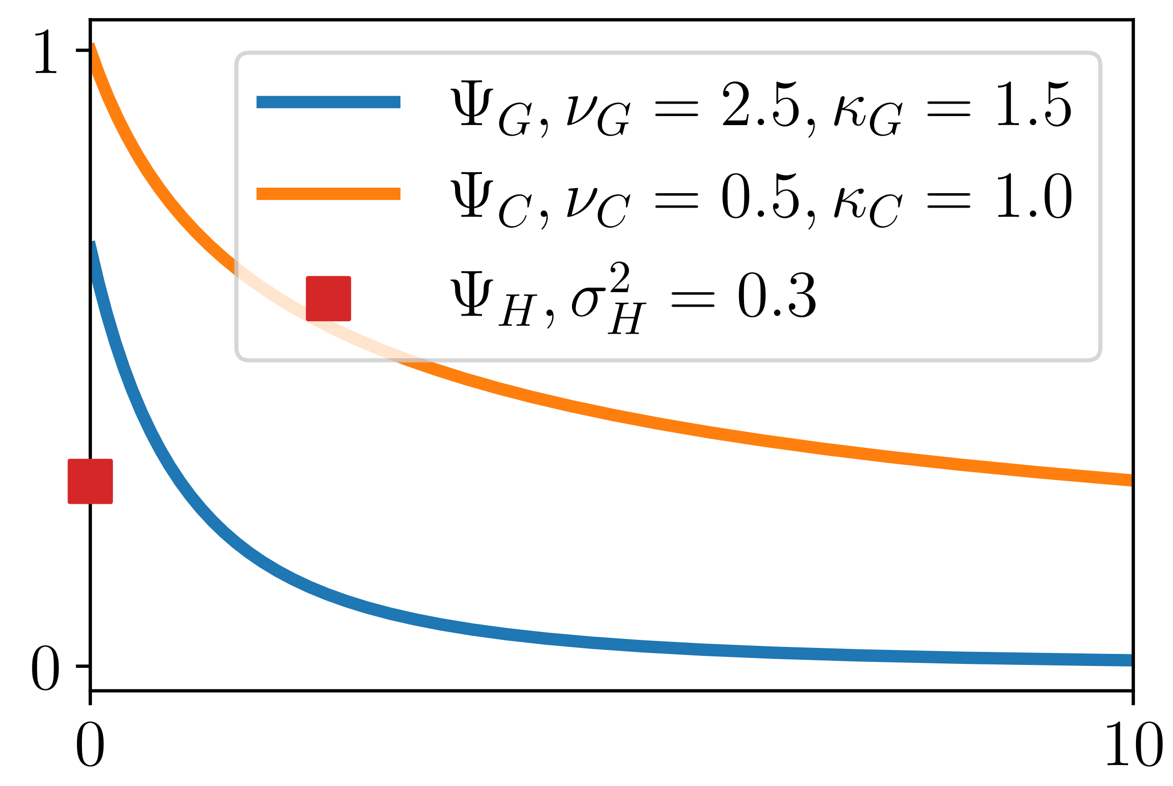
Composition of three GPs on the Hodge subspaces

- Kernel: $K_1 = K_G + K_H + K_C$
- Mutual independence hypothesis
- Separate learning of different components
- Automatic determination of Hodge components, instead of solving Hodge decomp.
- Edge Fourier Feature perspective

non-HC GP



HC GP



Alternative formulation

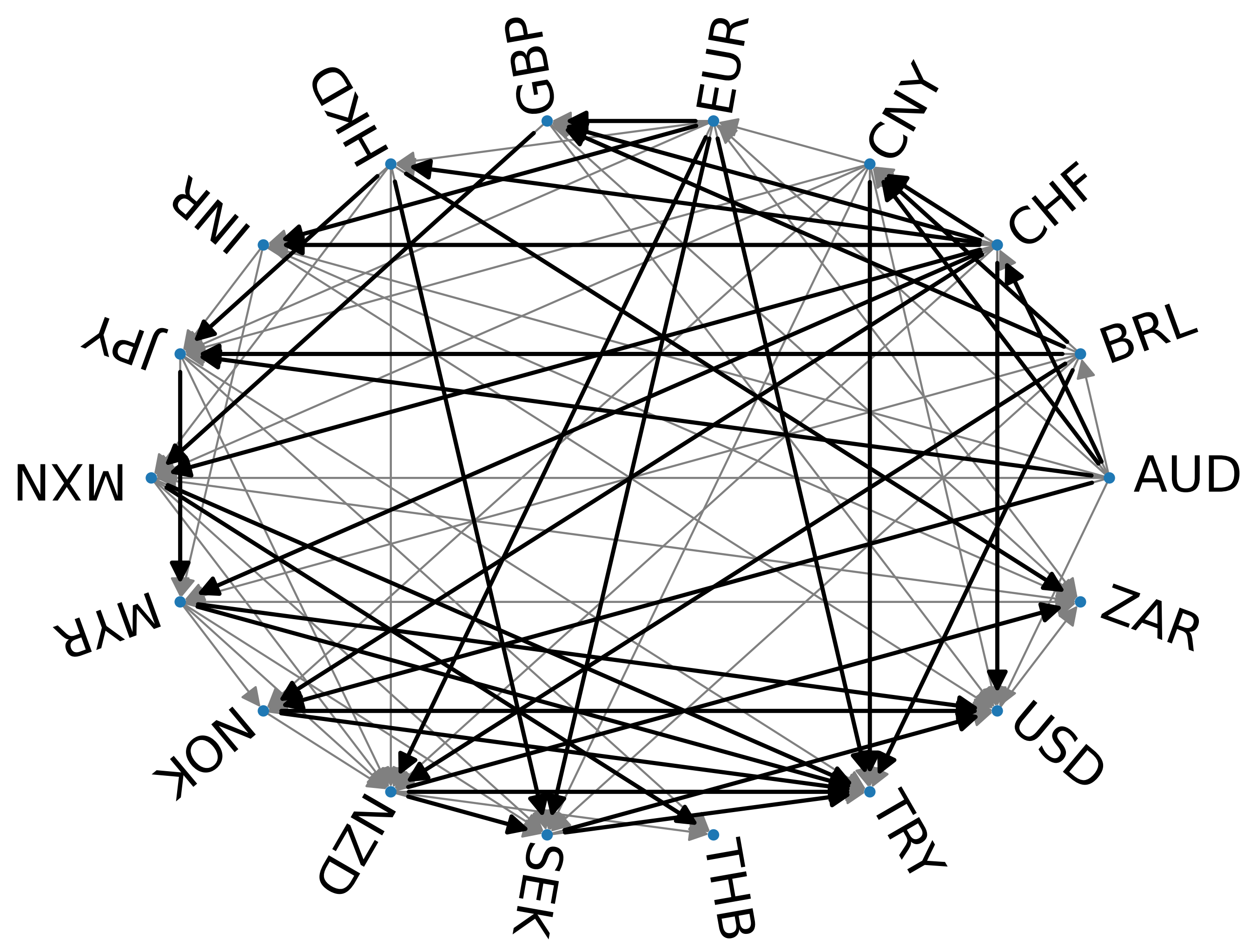
via node-edge-triangle interactions

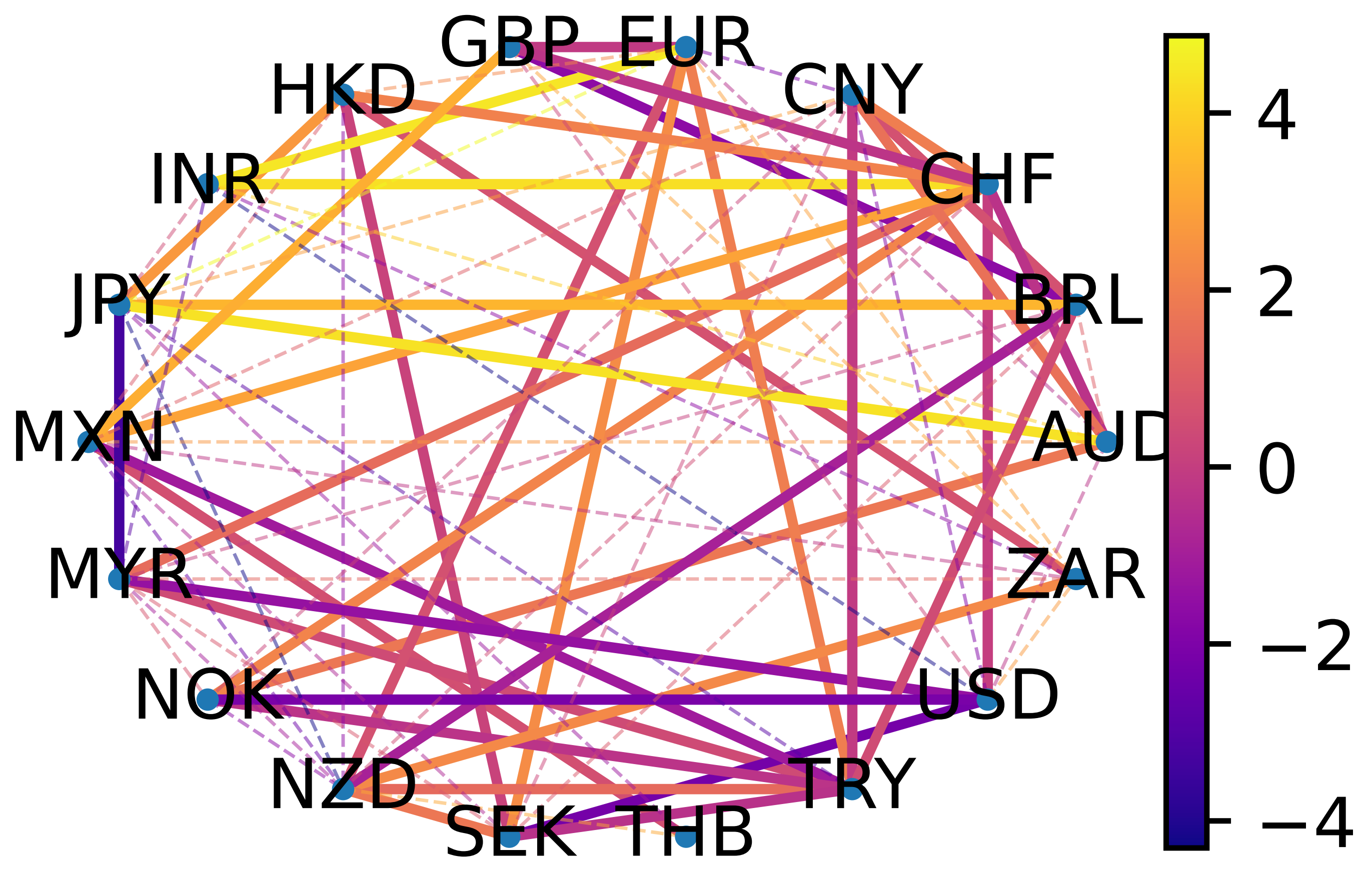
- Derivatives of GPs are also GPs
- Induce edge GPs from node and triangle GPs

$$K_1 = K_H + B_1^\top K_0 B_1 + B_2 K_2 B_2^\top$$

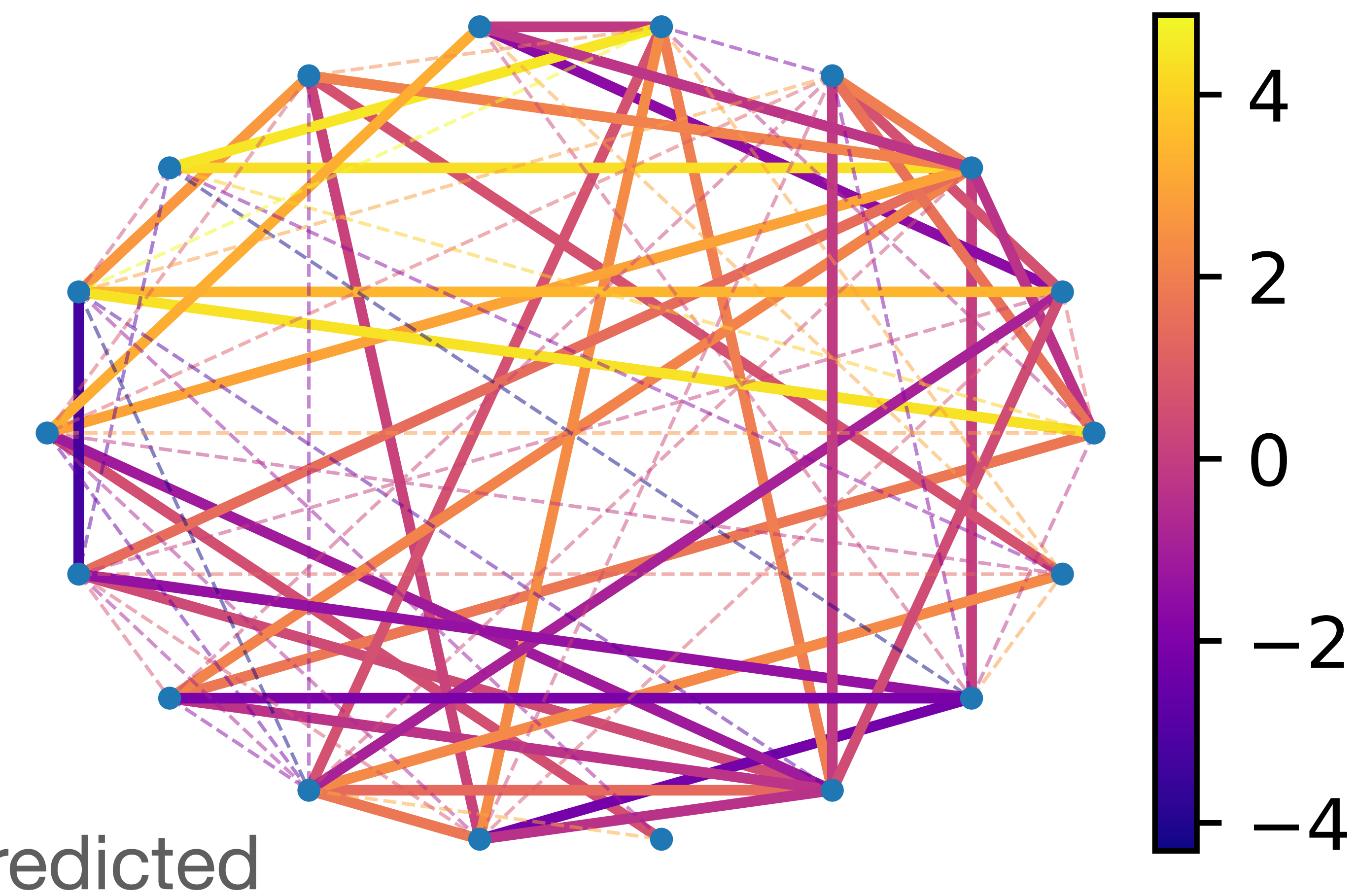
- Induce node GPs from edge GPs

GP based Forex prediction

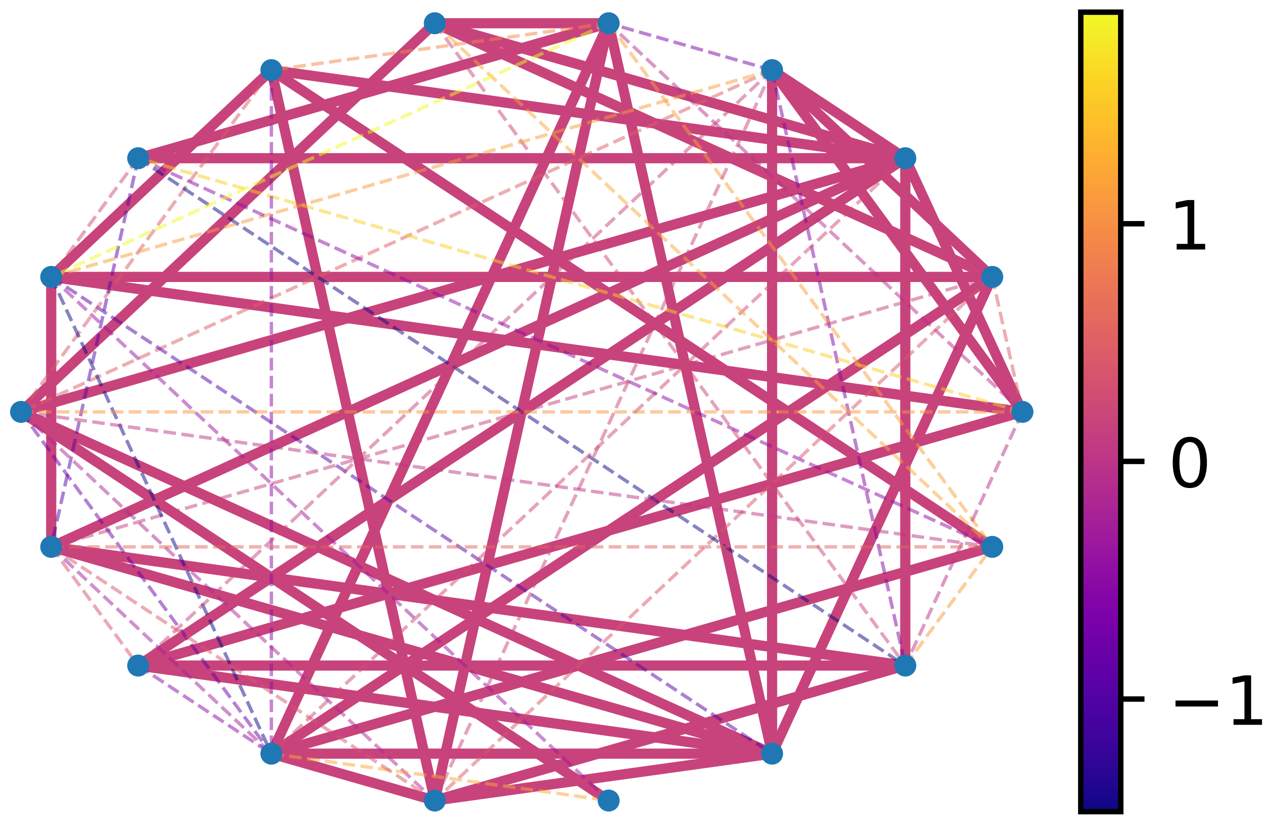




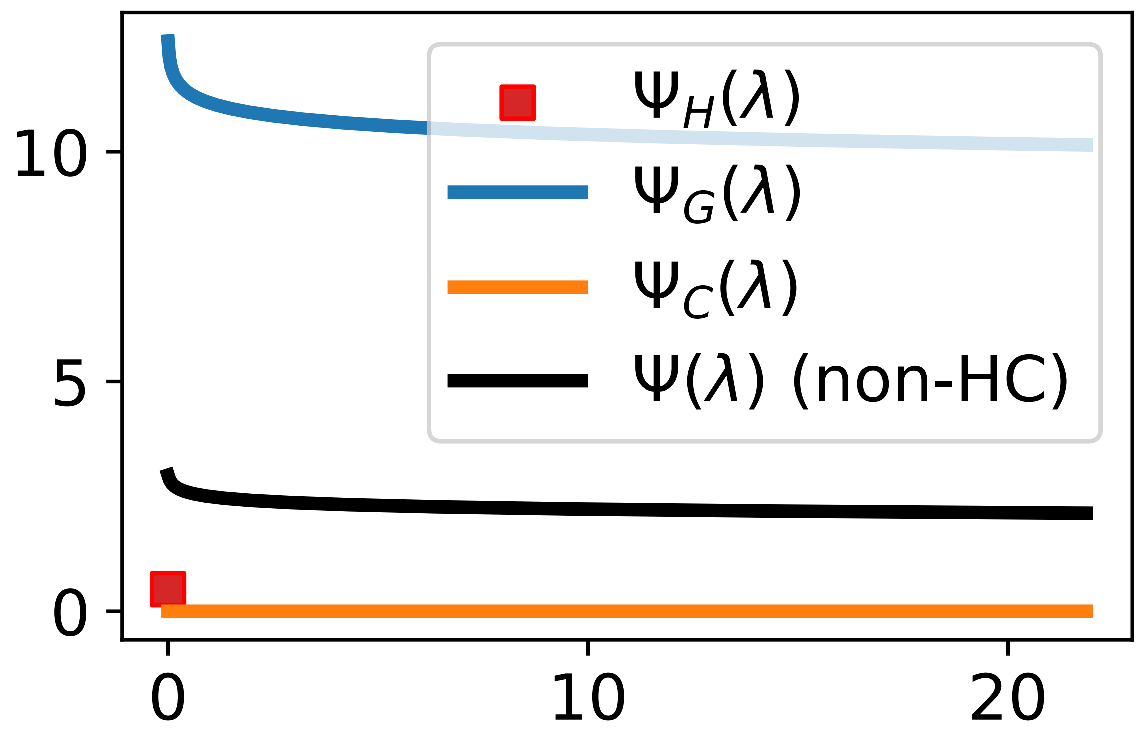
True



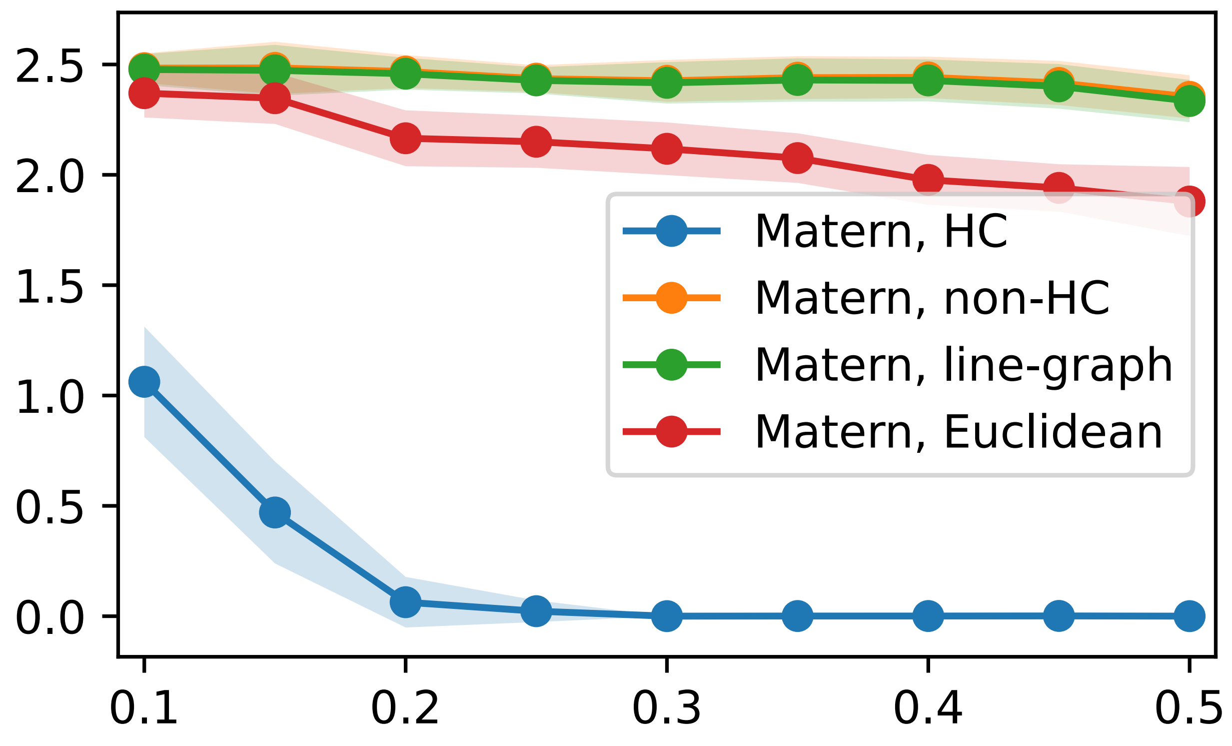
Predicted



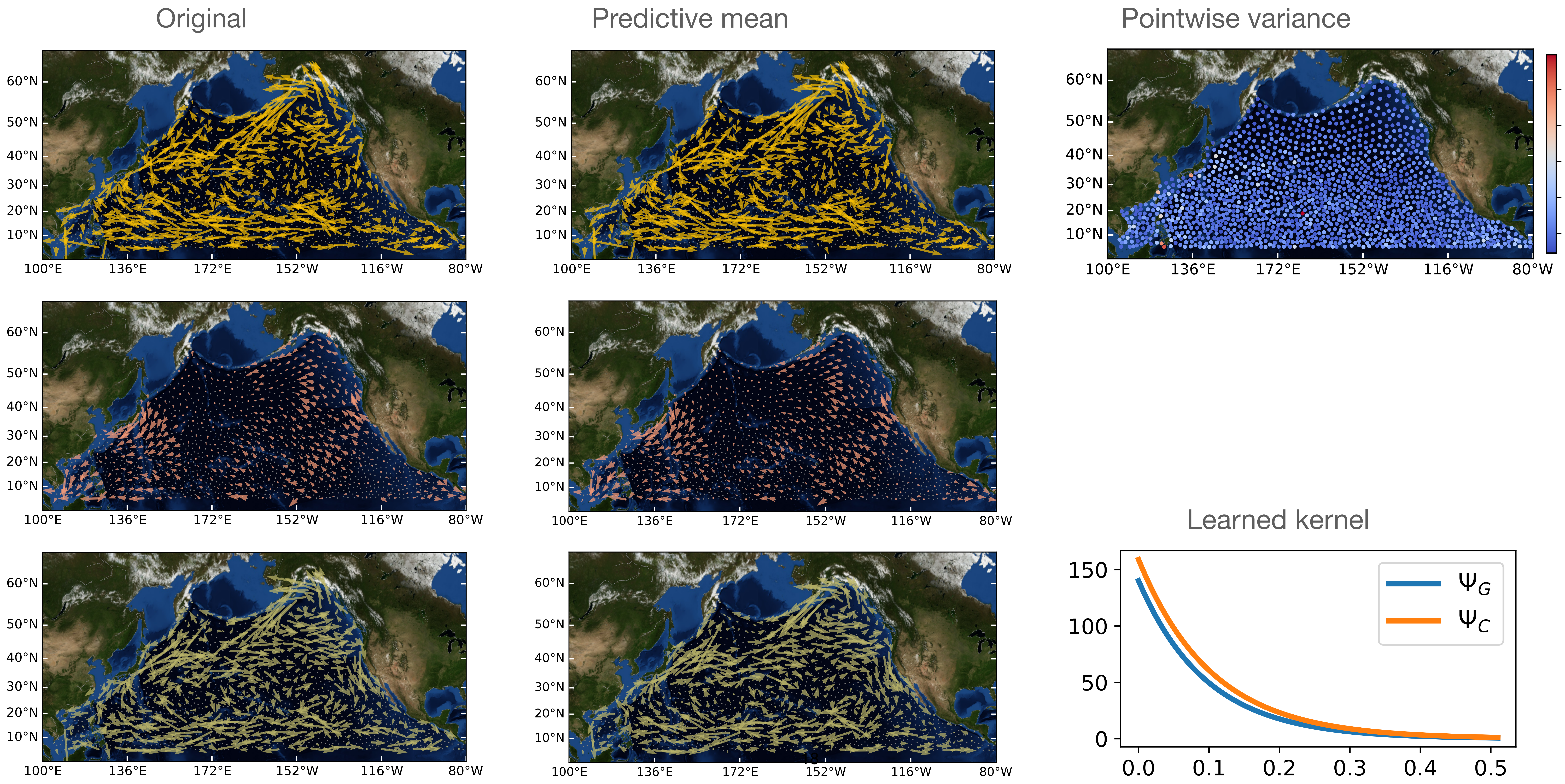
non-Hodge



Learned kernels



GP based Ocean current analysis

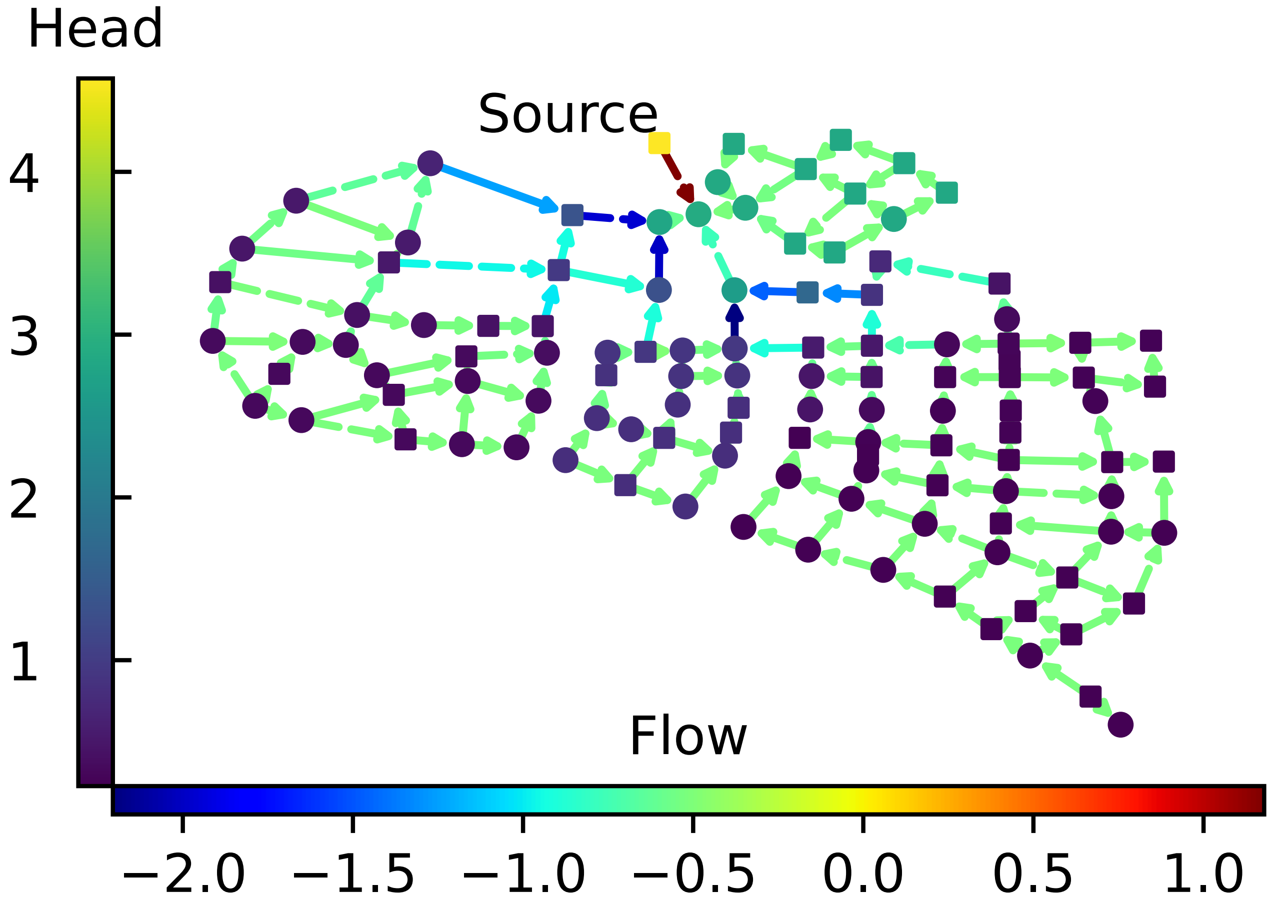


State estimation in water supply networks

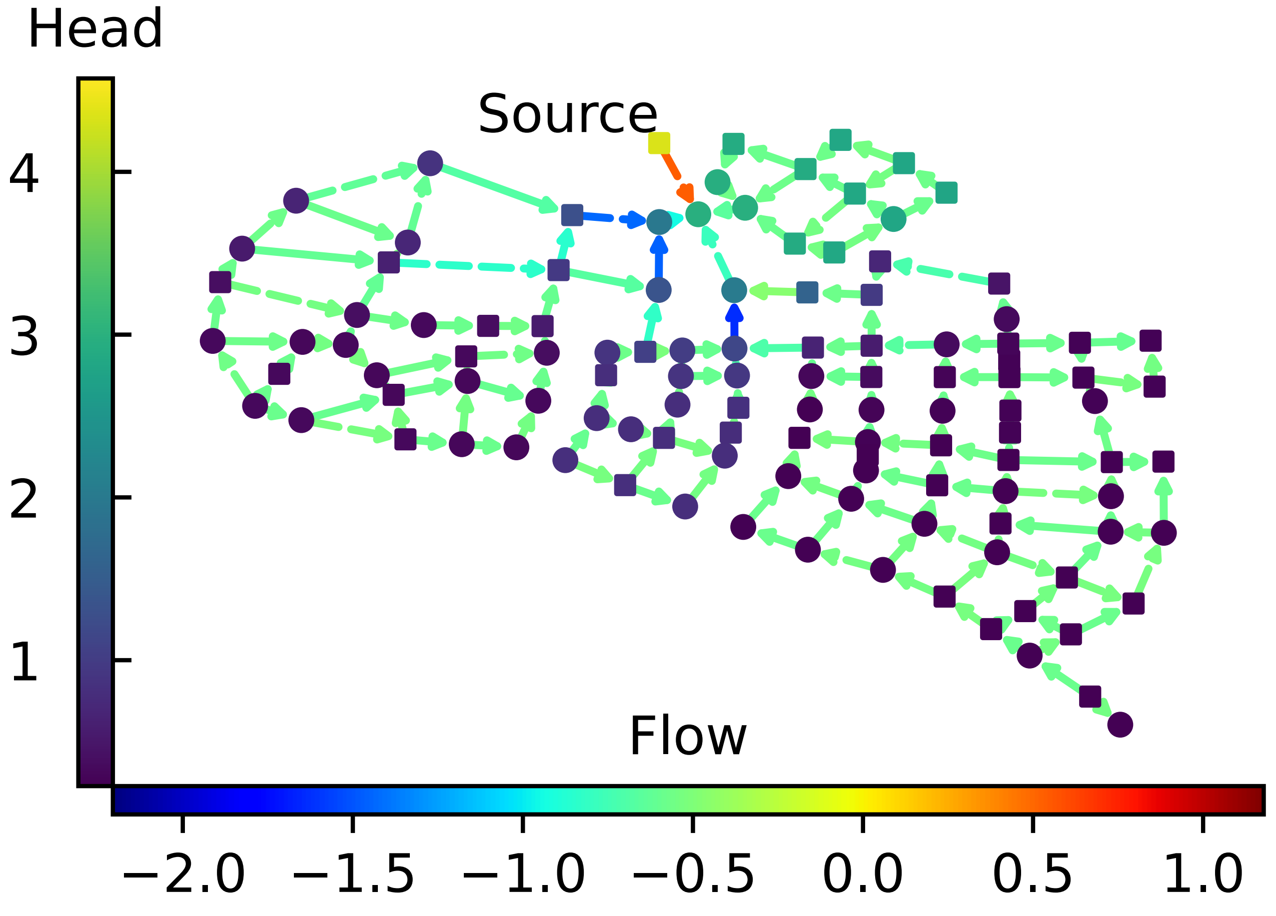
Based on the node-edge joint GPs

$$\mathbf{B}_1^\top \mathbf{f}_0 = \bar{\mathbf{f}}_1 := \mathrm{diag}(\mathbf{r})\mathbf{f}_1^{1.852}$$

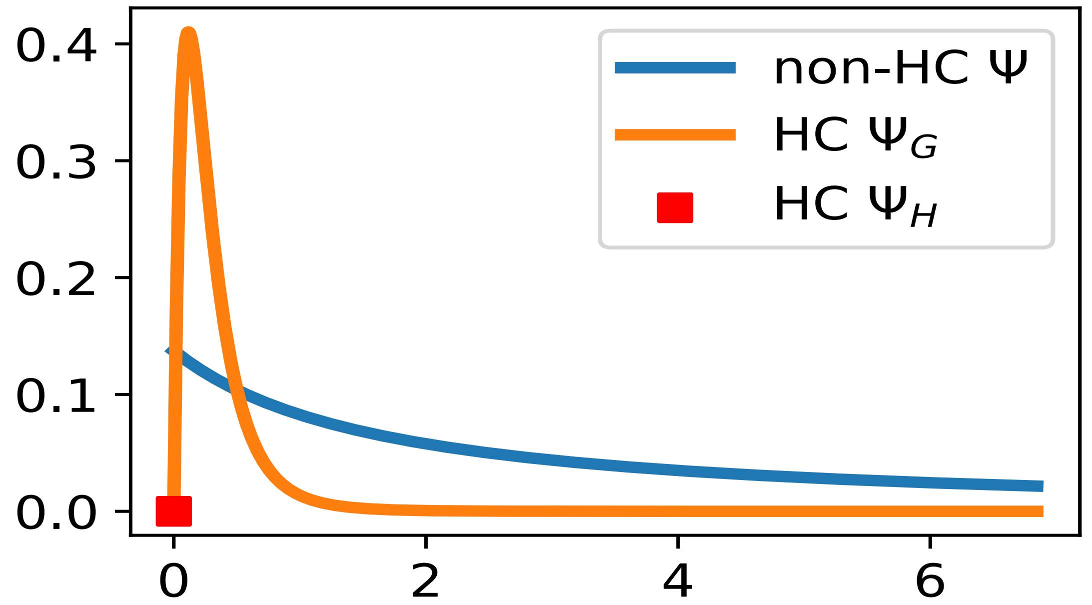
$$\begin{pmatrix} \mathbf{f}_0 \\ \bar{\mathbf{f}}_1 \end{pmatrix} \sim \mathrm{GP}\left(\mathbf{0}, \begin{pmatrix} \mathbf{K}_0 & \\ & \mathbf{K}_1 = \mathbf{B}_1^\top \mathbf{K}_0 \mathbf{B}_1 \end{pmatrix}\right)$$



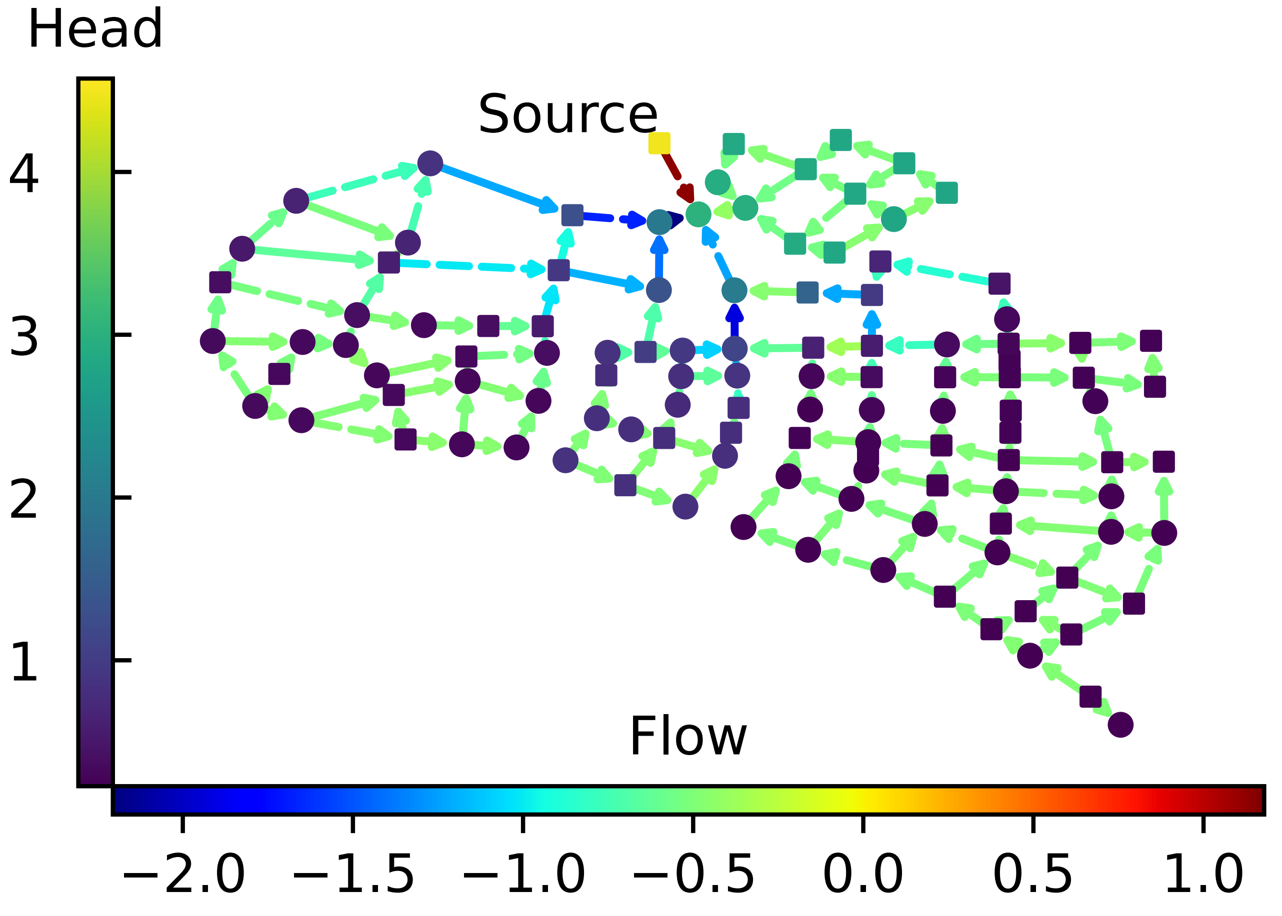
Original



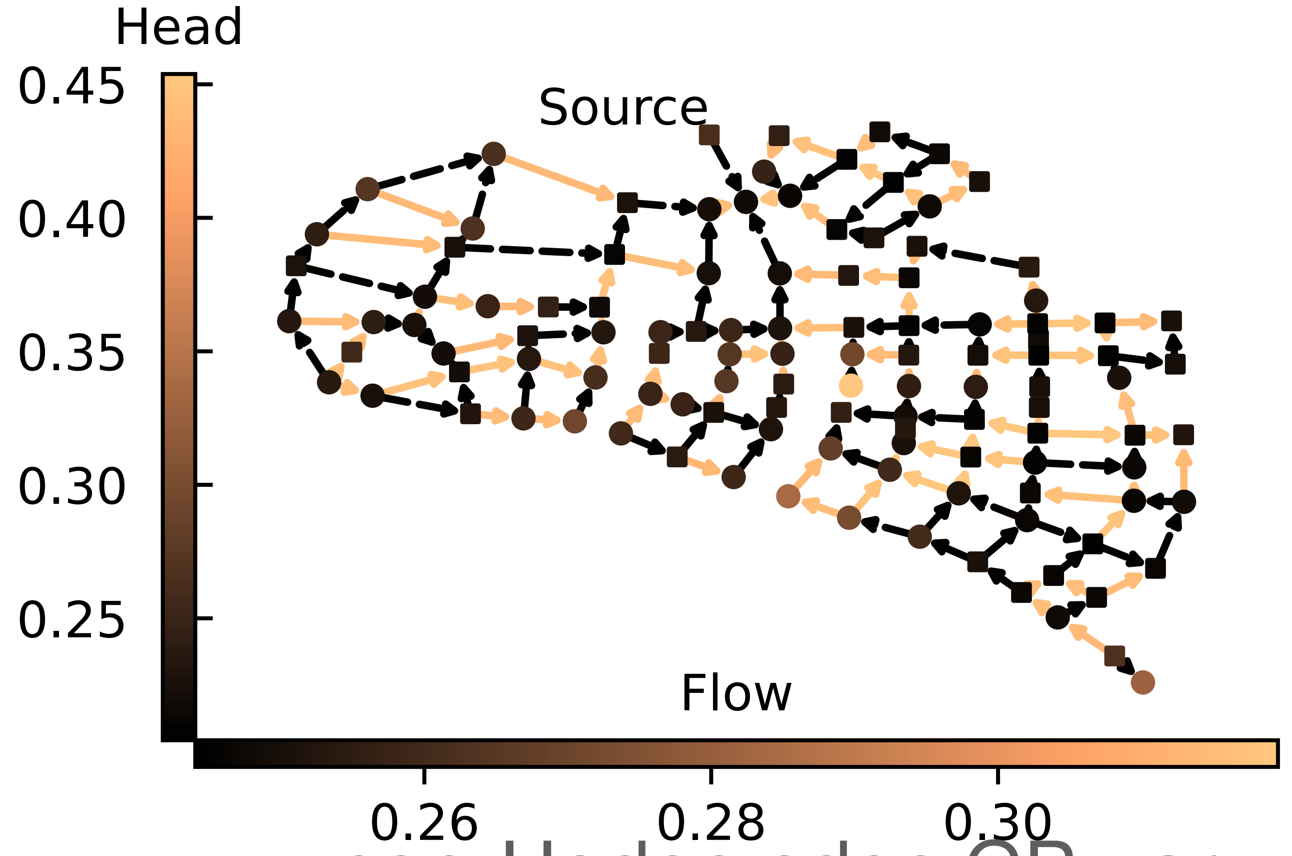
non-Hodge edge GP



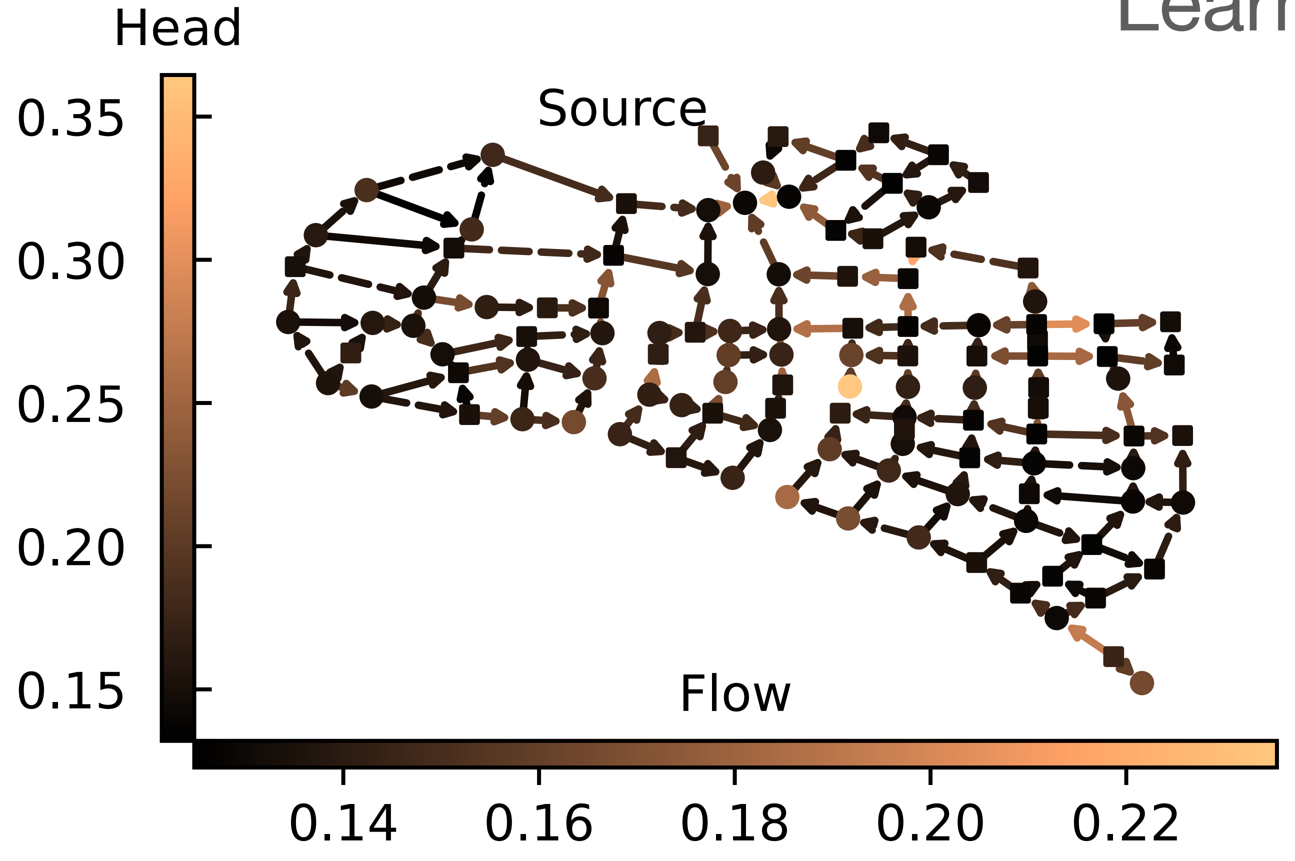
Learned kernel



Hodge edge GP



non-Hodge edge GP, var



Hodge edge GP, var

Conclusion

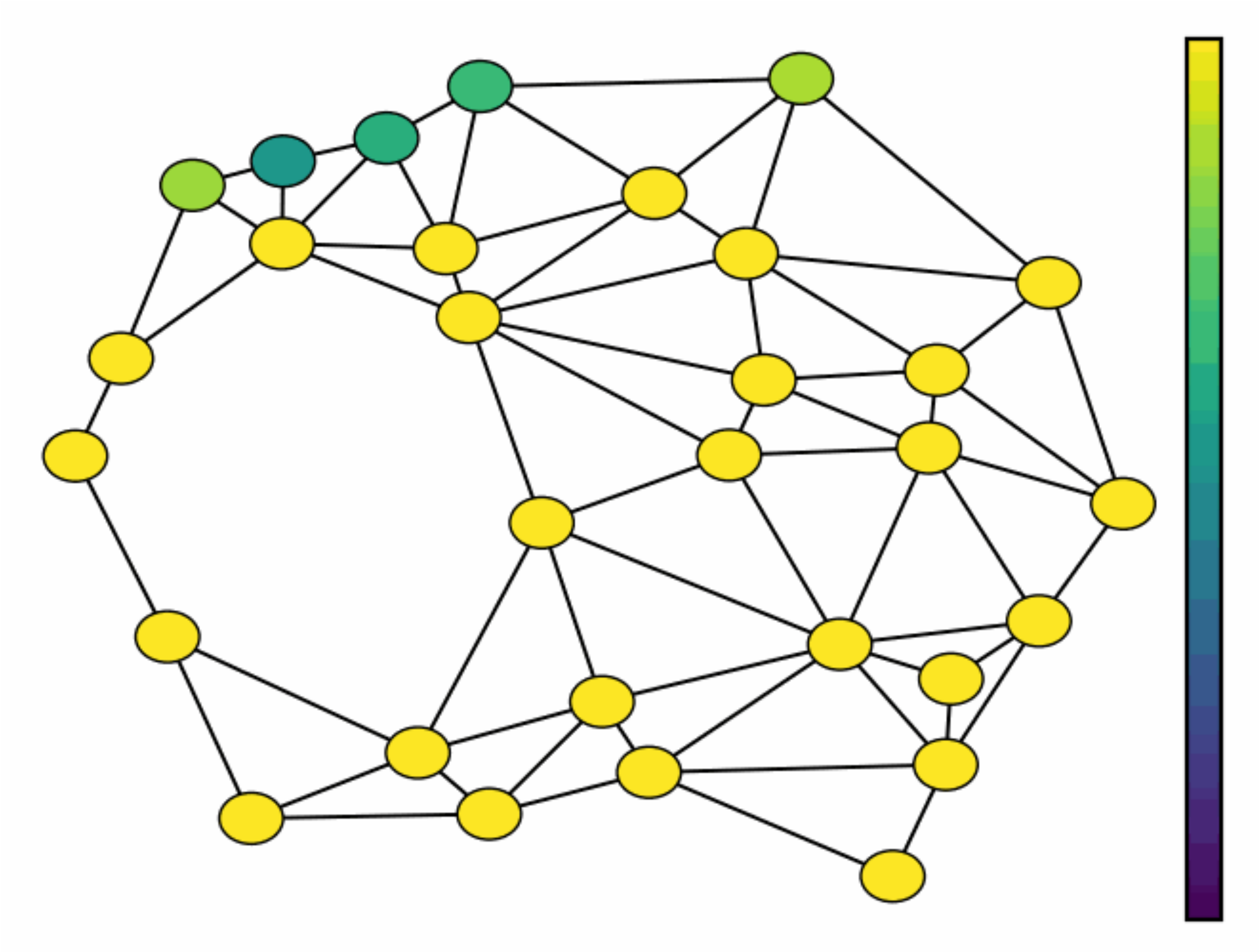
- How to generalize GPs to non-Euclidean domains? SDE framework
- How to measure edge functions? Div and curl, like VFs
- What is a good edge GP? Edge dependency + Hodge decomposition
- Node-edge-triangle joint GPs Alain et al. 2023
- Continuous version: Euclidean VF Berlinghieri et al. 2023; Manifold VF Robert-Nicoud et al. 2024

Paper
Code

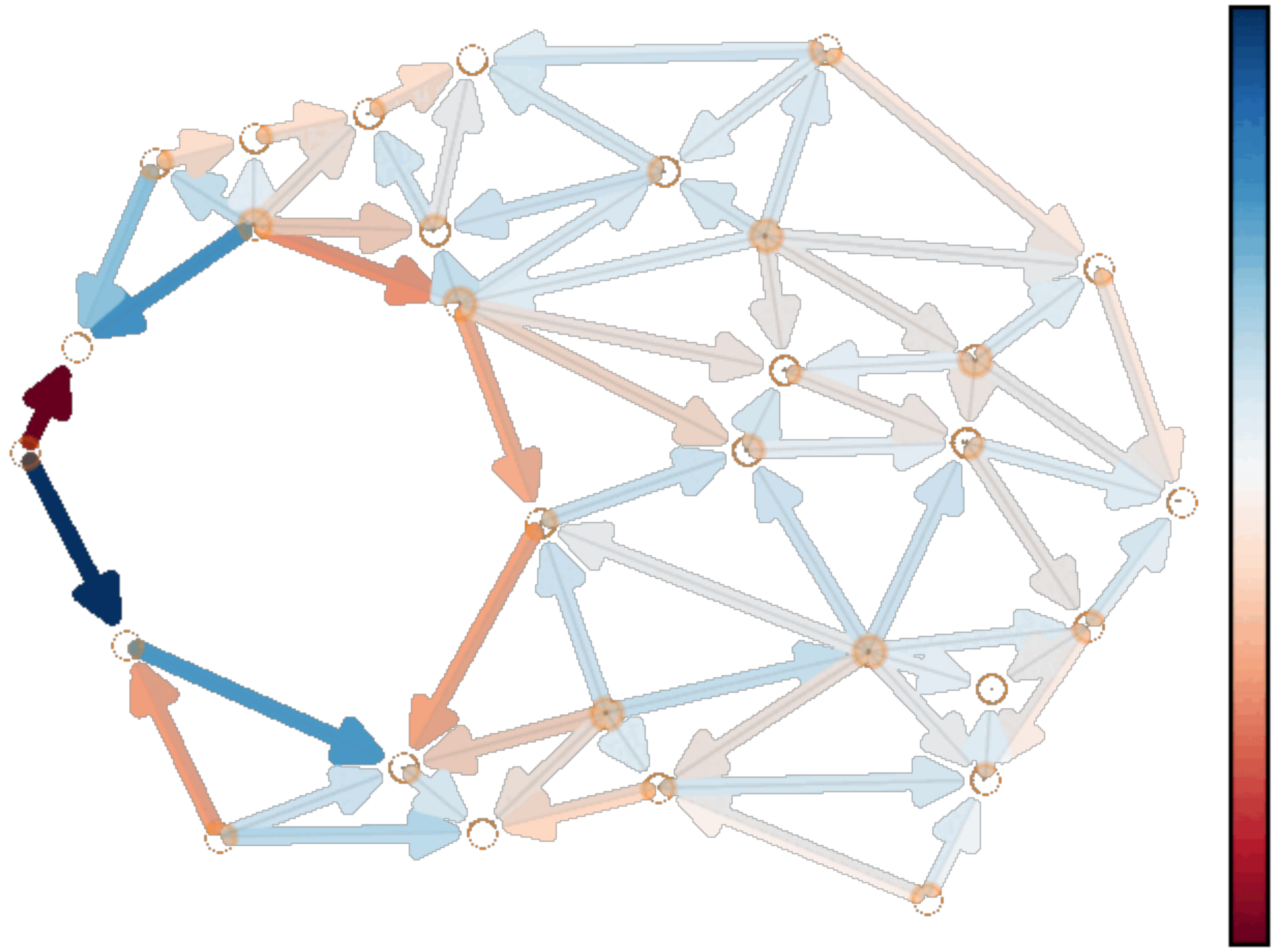
Thank you!

Appendix

Diffusion kernels from diffusion processes



Diffusion on nodes



Diffusion on edges

Tabular results

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

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

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

Sampling gradient and curl edge GPs

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.

Posterior distribution of Hodge components

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

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.