Extracting cycles from spatiotemporal data and coherent sets across multiple dynamic regimes

Gary Froyland

School of Mathematics and Statistics
University of New South Wales, Sydney

Reconstructing Network Dynamics from Data: Applications to Neuroscience and Beyond

IPAM, UCLA, 29 August – 2 September 2022

joint work with

D. Giannakis (Dartmouth), B. Lintner (Rutgers), M. Pike (Rutgers), J. Slawinska (Dartmouth), P. Koltai (FU Berlin)

Outline

I will discuss two recent techniques for discovering dynamics from data.

1. **Cycle extraction from spatiotemporal data:**
   - Extracting the periods and phase patterns in state space of long-lived cycles embedded in high-dimensional data.
   - Specific example will be extracting a canonical El-Niño Southern Oscillation (ENSO) pattern from sea-surface temperature data.
2. **Coherent set extraction across multiple dynamic regimes:**
   - In turbulent fluid flow, coherent parcels of fluid emerge, remain coherent for a while, and dissipate.
   - I'll describe a framework to identify the presence, location, and lifetimes of coherent parcels.
   - In the setting of time-evolving complex networks, one can think of semi-persistent clusters that emerge, exist for a while, and then break up. How to find them?

Outline

I will discuss two recent techniques for discovering dynamics from data.

1. **Cycle extraction from spatiotemporal data:**
   - Extracting the periods and phase patterns in state space of long-lived cycles embedded in high-dimensional data.
   - Specific example will be extracting a canonical El-Niño Southern Oscillation (ENSO) pattern from sea-surface temperature data.
2. **Coherent set extraction across multiple dynamic regimes:**
   - In turbulent fluid flow, coherent parcels of fluid emerge, remain coherent for a while, and dissipate.
   - I'll describe a framework to identify the presence, location, and lifetimes of coherent parcels.
   - In the setting of time-evolving complex networks, one can think of semi-persistent clusters that emerge, exist for a while, and then break up. How to find them?

Outline

I will discuss two recent techniques for discovering dynamics from data.

1. **Cycle extraction from spatiotemporal data:**
   - Extracting the periods and phase patterns in state space of long-lived cycles embedded in high-dimensional data.
   - Specific example will be extracting a canonical El-Niño Southern Oscillation (ENSO) pattern from sea-surface temperature data.
2. **Coherent set extraction across multiple dynamic regimes:**
   - In turbulent fluid flow, coherent parcels of fluid emerge, remain coherent for a while, and dissipate.
   - I'll describe a framework to identify the presence, location, and lifetimes of coherent parcels.
   - In the setting of time-evolving complex networks, one can think of semi-persistent clusters that emerge, exist for a while, and then break up. How to find them?

Outline

I will discuss two recent techniques for discovering dynamics from data.

1. **Cycle extraction from spatiotemporal data:**
   - Extracting the periods and phase patterns in state space of long-lived cycles embedded in high-dimensional data.
   - Specific example will be extracting a canonical El-Niño Southern Oscillation (ENSO) pattern from sea-surface temperature data.
2. **Coherent set extraction across multiple dynamic regimes:**
   - In turbulent fluid flow, coherent parcels of fluid emerge, remain coherent for a while, and dissipate.
   - I'll describe a framework to identify the presence, location, and lifetimes of coherent parcels.
   - In the setting of time-evolving complex networks, one can think of semi-persistent clusters that emerge, exist for a while, and then break up. How to find them?

Outline

I will discuss two recent techniques for discovering dynamics from data.

1. **Cycle extraction from spatiotemporal data:**
   - Extracting the periods and phase patterns in state space of long-lived cycles embedded in high-dimensional data.
   - Specific example will be extracting a canonical El-Niño Southern Oscillation (ENSO) pattern from sea-surface temperature data.
2. **Coherent set extraction across multiple dynamic regimes:**
   - In turbulent fluid flow, coherent parcels of fluid emerge, remain coherent for a while, and dissipate.
   - I'll describe a framework to identify the presence, location, and lifetimes of coherent parcels.
   - In the setting of time-evolving complex networks, one can think of semi-persistent clusters that emerge, exist for a while, and then break up. How to find them?

Outline

I will discuss two recent techniques for discovering dynamics from data.

1. **Cycle extraction from spatiotemporal data:**
   - Extracting the periods and phase patterns in state space of long-lived cycles embedded in high-dimensional data.
   - Specific example will be extracting a canonical El-Niño Southern Oscillation (ENSO) pattern from sea-surface temperature data.
2. **Coherent set extraction across multiple dynamic regimes:**
   - In turbulent fluid flow, coherent parcels of fluid emerge, remain coherent for a while, and dissipate.
   - I'll describe a framework to identify the presence, location, and lifetimes of coherent parcels.
   - In the setting of time-evolving complex networks, one can think of semi-persistent clusters that emerge, exist for a while, and then break up. How to find them?

Outline

I will discuss two recent techniques for discovering dynamics from data.

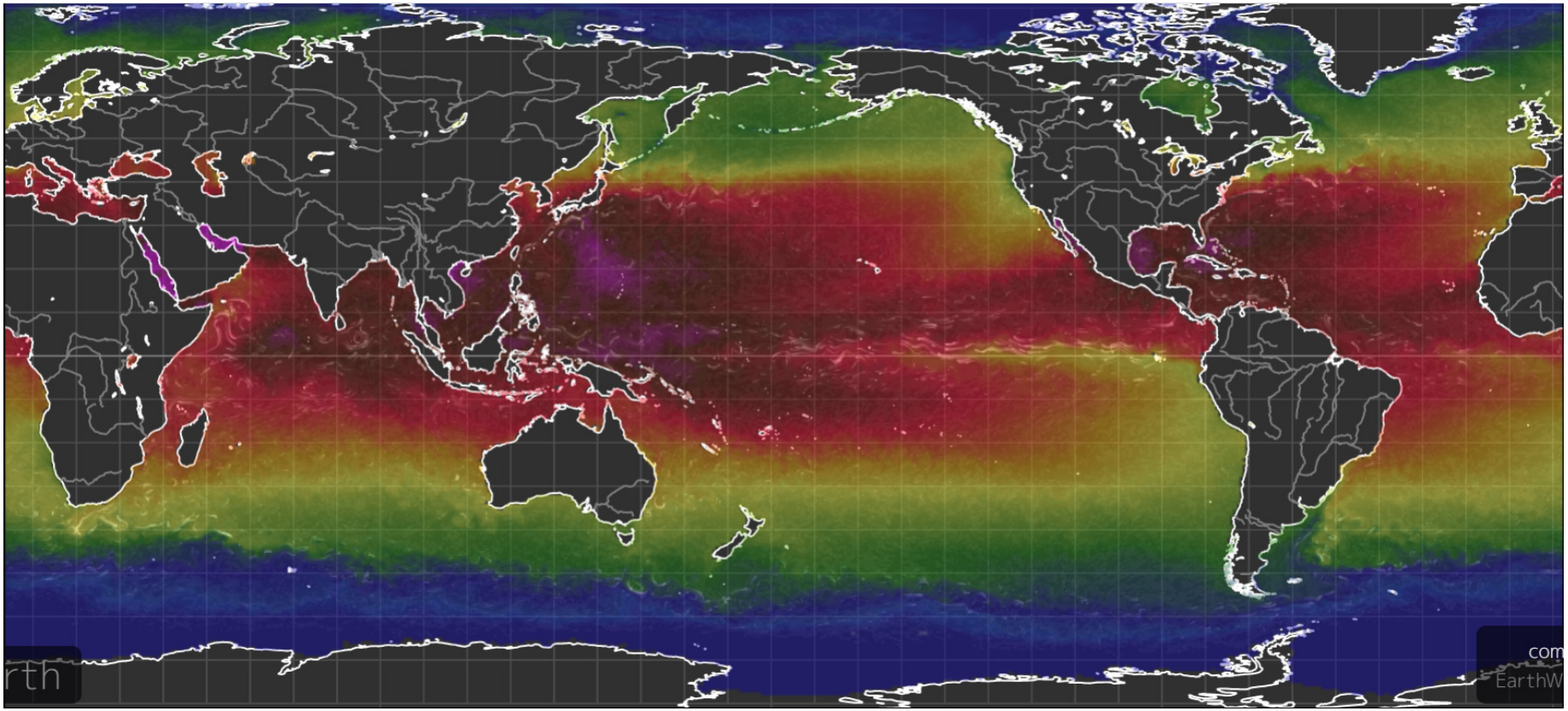
1. **Cycle extraction from spatiotemporal data:**
   - Extracting the periods and phase patterns in state space of long-lived cycles embedded in high-dimensional data.
   - Specific example will be extracting a canonical El-Niño Southern Oscillation (ENSO) pattern from sea-surface temperature data.
2. **Coherent set extraction across multiple dynamic regimes:**
   - In turbulent fluid flow, coherent parcels of fluid emerge, remain coherent for a while, and dissipate.
   - I'll describe a framework to identify the presence, location, and lifetimes of coherent parcels.
   - In the setting of time-evolving complex networks, one can think of semi-persistent clusters that emerge, exist for a while, and then break up. How to find them?

PART I

CYCLE EXTRACTION FROM HIGH-DIMENSIONAL SPATIOTEMPORAL DATA

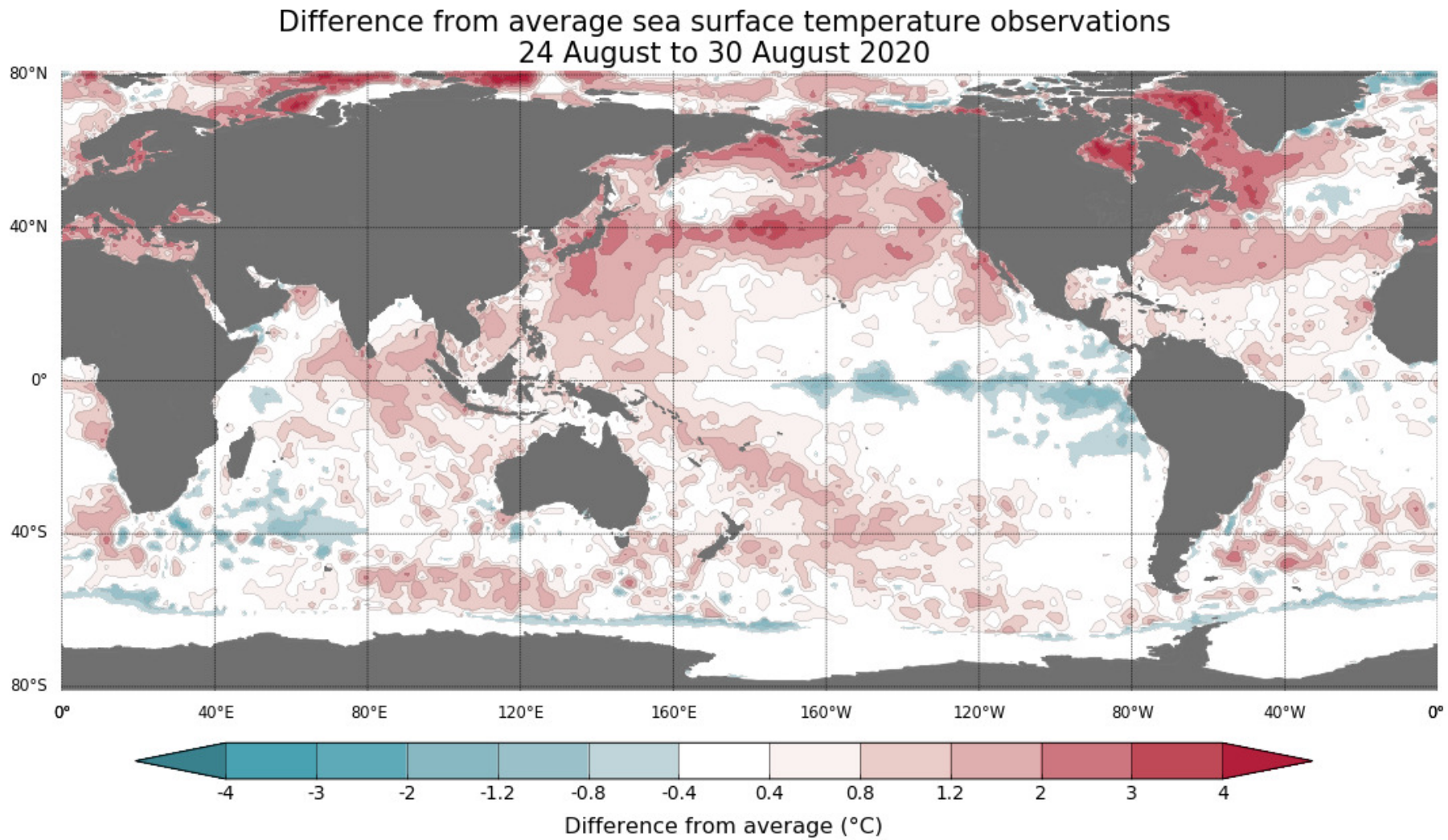
Global Sea Surface Temperature

SST on 1 September 2020.



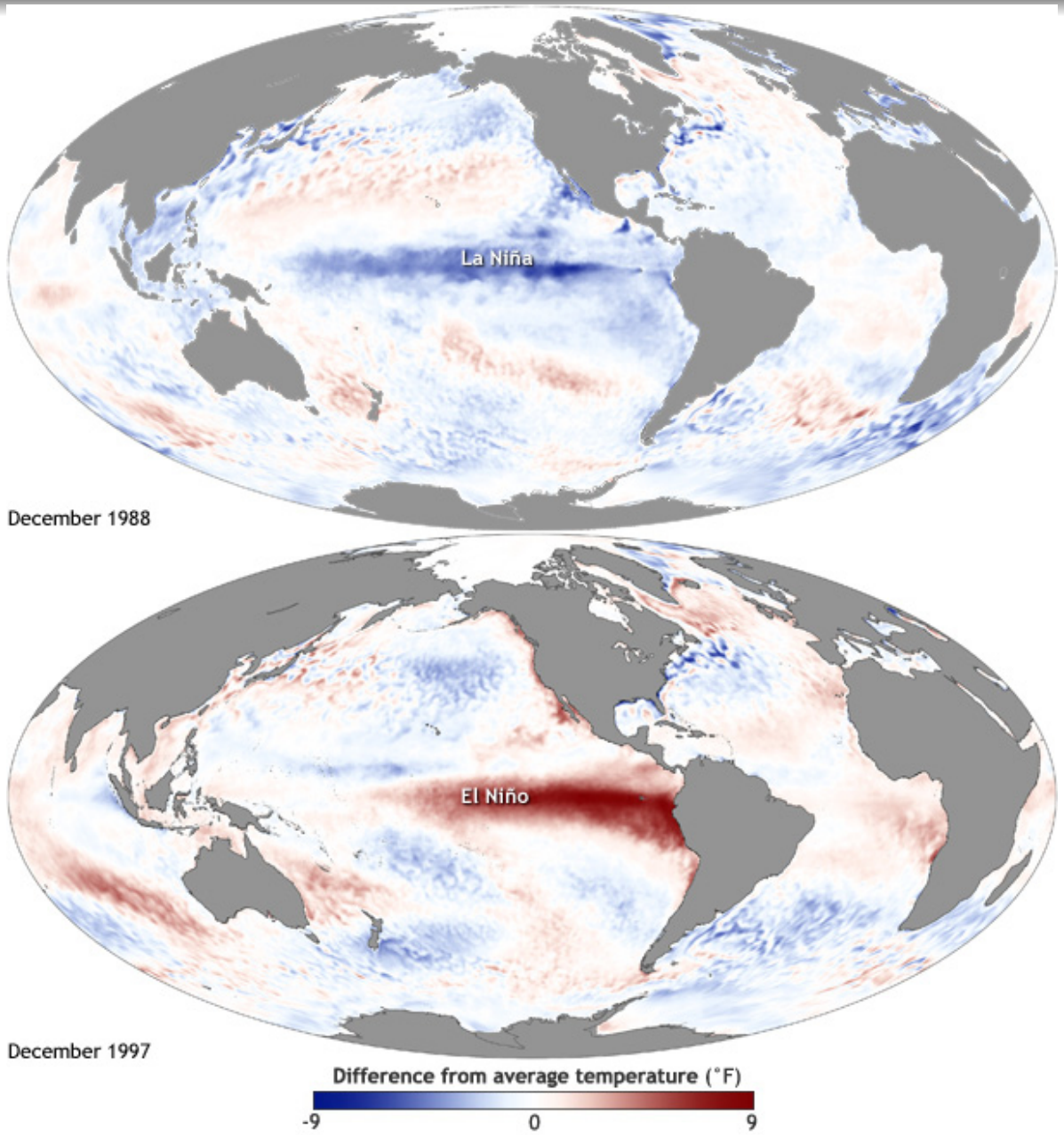
Source: earth.nullschool.net

Global Sea Surface Temperature Anomaly



Source: BoM

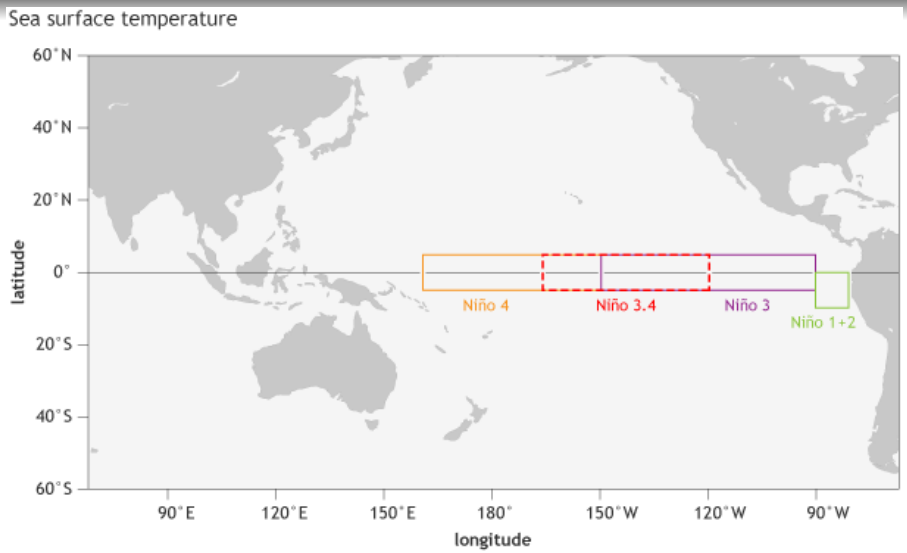
El Niño and La Niña Extremes

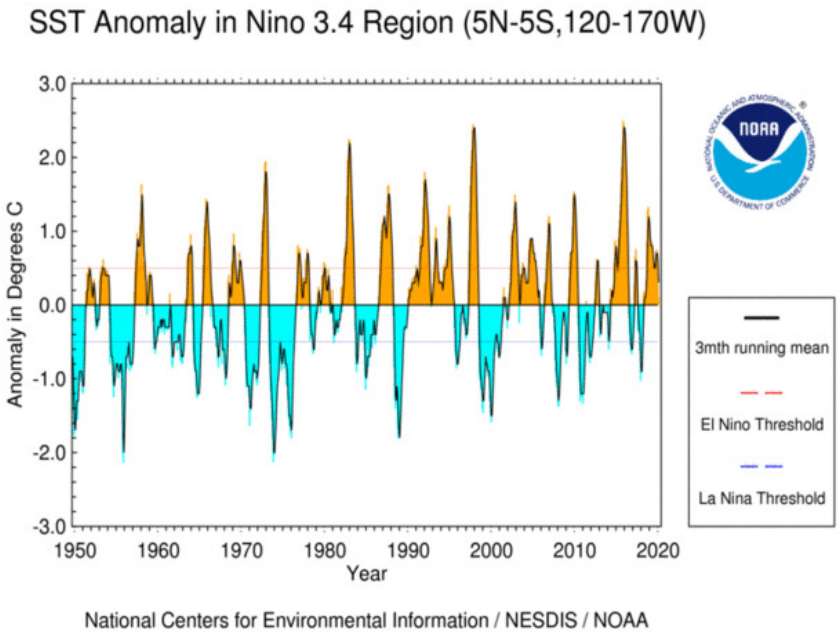


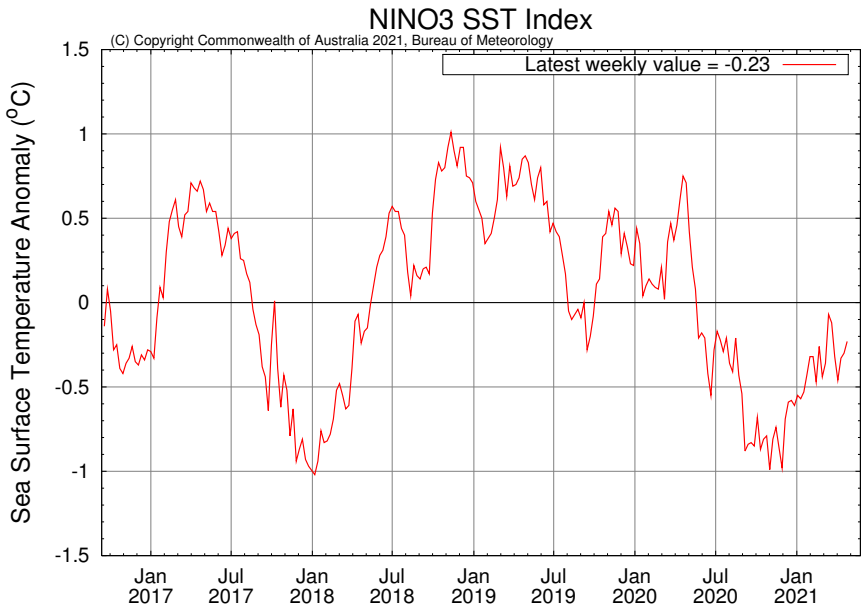
Source: NOAA

The Niño 3.4 Index

A standard way of assessing current El Niño strength is to average the sea-surface temperature over a well-chosen "Niño 3.4" box in the Indo-Pacific. This is called the Niño 3.4 Index and there is an **approximate cycle with period around 4 years**.







Our goals

1. **Extract a canonical strong ENSO cycle** from observational data (sea-surface temperature fields).
2. **"Rectify" the cycle** so that dynamics proceeds at a constant rate around the cycle; this will reveal more detail on El-Niño formation.
3. Demonstrate **better self-consistency** (equivariance/cyclicity in time) when compared with the Niño 3.4 index.
4. Because of items 2 and 3, obtain an **improved characterisation of ENSO with greater predictability**.

Our goals

1. **Extract a canonical strong ENSO cycle** from observational data (sea-surface temperature fields).
2. **"Rectify" the cycle** so that dynamics proceeds at a constant rate around the cycle; this will reveal more detail on El-Niño formation.
3. Demonstrate **better self-consistency** (equivariance/cyclicity in time) when compared with the Niño 3.4 index.
4. Because of items 2 and 3, obtain an **improved characterisation of ENSO with greater predictability**.

Our goals

1. **Extract a canonical strong ENSO cycle** from observational data (sea-surface temperature fields).
2. **"Rectify" the cycle** so that dynamics proceeds at a constant rate around the cycle; this will reveal more detail on El-Niño formation.
3. Demonstrate **better self-consistency** (equivariance/cyclicity in time) when compared with the Niño 3.4 index.
4. Because of items 2 and 3, obtain an **improved characterisation of ENSO with greater predictability**.

Our goals

1. **Extract a canonical strong ENSO cycle** from observational data (sea-surface temperature fields).
2. **"Rectify" the cycle** so that dynamics proceeds at a constant rate around the cycle; this will reveal more detail on El-Niño formation.
3. Demonstrate **better self-consistency** (equivariance/cyclicity in time) when compared with the Niño 3.4 index.
4. Because of items 2 and 3, obtain an **improved characterisation of ENSO with greater predictability**.

Transfer operator spectrum reveals angular rotation

- Let (Ω, μ) be a probability space (our phase space) and $\Phi : \Omega \circlearrowleft$ be our discrete-time dynamics (assumed invertible and ergodic).
- The **transfer operator** $\mathcal{P} : L^2(\Omega) \to L^2(\Omega)$ is given by $\mathcal{P}f = f \circ \Phi^{-1}$, where $f : \Omega \to \mathbb{C}$; note $\mathcal{P}$ is a linear composition operator.
- The simplest possible dynamical system for our purposes is where $\Omega = S^1$ is a circle of radius 1 and Φ is rotation by angle α.
- For the circle rotation, $\mathcal{P}$ has eigenvalues $\Lambda_k = e^{ik\alpha}$ with corresponding eigenfunctions $g_k(\theta) = e^{ik\theta}$ for $k \in \mathbb{Z}$ and $\theta \in S^1$.
- The leading nontrivial eigenvalue $\Lambda_1 = e^{i\alpha}$ gives the rotation angle α.
- Further, the eigenfunction $g_1 : S^1 \to \mathbb{C}$ given by $g_1(\theta) = e^{i\theta}$ is an isometric conjugacy between the dynamical system (S^1, Φ) and **rotation of the unit circle in $\mathbb{C}$ by α.**

$$\begin{array}{ccc} S^1 & \xrightarrow{\text{rotation by } \alpha} & S^1 \\ \downarrow{\scriptstyle g_1} & & \downarrow{\scriptstyle g_1} \\ \mathbb{C} & \xrightarrow{\text{rotation by } \alpha} & \mathbb{C} \end{array}$$

Transfer operator spectrum reveals angular rotation

- Let (Ω, μ) be a probability space (our phase space) and $\Phi : \Omega \circlearrowleft$ be our discrete-time dynamics (assumed invertible and ergodic).
- The **transfer operator** $\mathcal{P} : L^2(\Omega) \to L^2(\Omega)$ is given by $\mathcal{P}f = f \circ \Phi^{-1}$, where $f : \Omega \to \mathbb{C}$; note $\mathcal{P}$ is a linear composition operator.
- The simplest possible dynamical system for our purposes is where $\Omega = S^1$ is a circle of radius 1 and Φ is rotation by angle α.
- For the circle rotation, $\mathcal{P}$ has eigenvalues $\Lambda_k = e^{ik\alpha}$ with corresponding eigenfunctions $g_k(\theta) = e^{ik\theta}$ for $k \in \mathbb{Z}$ and $\theta \in S^1$.
- The leading nontrivial eigenvalue $\Lambda_1 = e^{i\alpha}$ gives the rotation angle α.
- Further, the eigenfunction $g_1 : S^1 \to \mathbb{C}$ given by $g_1(\theta) = e^{i\theta}$ is an isometric conjugacy between the dynamical system (S^1, Φ) and **rotation of the unit circle in $\mathbb{C}$ by α.**

$$\begin{array}{ccc} S^1 & \xrightarrow{\text{rotation by } \alpha} & S^1 \\ \downarrow{\scriptstyle g_1} & & \downarrow{\scriptstyle g_1} \\ \mathbb{C} & \xrightarrow{\text{rotation by } \alpha} & \mathbb{C} \end{array}$$

Transfer operator spectrum reveals angular rotation

- Let (Ω, μ) be a probability space (our phase space) and $\Phi : \Omega \circlearrowleft$ be our discrete-time dynamics (assumed invertible and ergodic).
- The **transfer operator** $\mathcal{P} : L^2(\Omega) \to L^2(\Omega)$ is given by $\mathcal{P}f = f \circ \Phi^{-1}$, where $f : \Omega \to \mathbb{C}$; note $\mathcal{P}$ is a linear composition operator.
- The simplest possible dynamical system for our purposes is where $\Omega = S^1$ is a circle of radius 1 and Φ is rotation by angle α.
- For the circle rotation, $\mathcal{P}$ has eigenvalues $\Lambda_k = e^{ik\alpha}$ with corresponding eigenfunctions $g_k(\theta) = e^{ik\theta}$ for $k \in \mathbb{Z}$ and $\theta \in S^1$.
- The leading nontrivial eigenvalue $\Lambda_1 = e^{i\alpha}$ gives the rotation angle α.
- Further, the eigenfunction $g_1 : S^1 \to \mathbb{C}$ given by $g_1(\theta) = e^{i\theta}$ is an isometric conjugacy between the dynamical system (S^1, Φ) and **rotation of the unit circle in $\mathbb{C}$ by α.**

$$\begin{array}{ccc} S^1 & \xrightarrow{\text{rotation by } \alpha} & S^1 \\ \downarrow g_1 & & \downarrow g_1 \\ \mathbb{C} & \xrightarrow{\text{rotation by } \alpha} & \mathbb{C} \end{array}$$

Transfer operator spectrum reveals angular rotation

- Let (Ω, μ) be a probability space (our phase space) and $\Phi : \Omega \circlearrowleft$ be our discrete-time dynamics (assumed invertible and ergodic).
- The **transfer operator** $\mathcal{P} : L^2(\Omega) \to L^2(\Omega)$ is given by $\mathcal{P}f = f \circ \Phi^{-1}$, where $f : \Omega \to \mathbb{C}$; note $\mathcal{P}$ is a linear composition operator.
- The simplest possible dynamical system for our purposes is where $\Omega = S^1$ is a circle of radius 1 and Φ is rotation by angle α.
- For the circle rotation, $\mathcal{P}$ has eigenvalues $\Lambda_k = e^{ik\alpha}$ with corresponding eigenfunctions $g_k(\theta) = e^{ik\theta}$ for $k \in \mathbb{Z}$ and $\theta \in S^1$.
- The leading nontrivial eigenvalue $\Lambda_1 = e^{i\alpha}$ gives the rotation angle α.
- Further, the eigenfunction $g_1 : S^1 \to \mathbb{C}$ given by $g_1(\theta) = e^{i\theta}$ is an isometric conjugacy between the dynamical system (S^1, Φ) and **rotation of the unit circle in $\mathbb{C}$ by α.**

$$\begin{array}{ccc} S^1 & \xrightarrow{\text{rotation by } \alpha} & S^1 \\ \downarrow g_1 & & \downarrow g_1 \\ \mathbb{C} & \xrightarrow{\text{rotation by } \alpha} & \mathbb{C} \end{array}$$

Transfer operator spectrum reveals angular rotation

- Let (Ω, μ) be a probability space (our phase space) and $\Phi : \Omega \circlearrowleft$ be our discrete-time dynamics (assumed invertible and ergodic).
- The **transfer operator** $\mathcal{P} : L^2(\Omega) \to L^2(\Omega)$ is given by $\mathcal{P}f = f \circ \Phi^{-1}$, where $f : \Omega \to \mathbb{C}$; note $\mathcal{P}$ is a linear composition operator.
- The simplest possible dynamical system for our purposes is where $\Omega = S^1$ is a circle of radius 1 and Φ is rotation by angle α.
- For the circle rotation, $\mathcal{P}$ has eigenvalues $\Lambda_k = e^{ik\alpha}$ with corresponding eigenfunctions $g_k(\theta) = e^{ik\theta}$ for $k \in \mathbb{Z}$ and $\theta \in S^1$.
- The leading nontrivial eigenvalue $\Lambda_1 = e^{i\alpha}$ gives the rotation angle α.
- Further, the eigenfunction $g_1 : S^1 \to \mathbb{C}$ given by $g_1(\theta) = e^{i\theta}$ is an isometric conjugacy between the dynamical system (S^1, Φ) and **rotation of the unit circle in $\mathbb{C}$ by α.**

$$\begin{array}{ccc} S^1 & \xrightarrow{\text{rotation by } \alpha} & S^1 \\ \downarrow g_1 & & \downarrow g_1 \\ \mathbb{C} & \xrightarrow{\text{rotation by } \alpha} & \mathbb{C} \end{array}$$

Transfer operator spectrum reveals angular rotation

- Let (Ω, μ) be a probability space (our phase space) and $\Phi : \Omega \circlearrowleft$ be our discrete-time dynamics (assumed invertible and ergodic).
- The **transfer operator** $\mathcal{P} : L^2(\Omega) \to L^2(\Omega)$ is given by $\mathcal{P}f = f \circ \Phi^{-1}$, where $f : \Omega \to \mathbb{C}$; note $\mathcal{P}$ is a linear composition operator.
- The simplest possible dynamical system for our purposes is where $\Omega = S^1$ is a circle of radius 1 and Φ is rotation by angle α.
- For the circle rotation, $\mathcal{P}$ has eigenvalues $\Lambda_k = e^{ik\alpha}$ with corresponding eigenfunctions $g_k(\theta) = e^{ik\theta}$ for $k \in \mathbb{Z}$ and $\theta \in S^1$.
- The leading nontrivial eigenvalue $\Lambda_1 = e^{i\alpha}$ gives the rotation angle α.
- Further, the eigenfunction $g_1 : S^1 \to \mathbb{C}$ given by $g_1(\theta) = e^{i\theta}$ is an isometric conjugacy between the dynamical system (S^1, Φ) and **rotation of the unit circle in $\mathbb{C}$ by α**.

$$\begin{array}{ccc} S^1 & \xrightarrow{\text{rotation by } \alpha} & S^1 \\ \downarrow g_1 & & \downarrow g_1 \\ \mathbb{C} & \xrightarrow{\text{rotation by } \alpha} & \mathbb{C} \end{array}$$

Transfer operator spectrum reveals angular rotation

- In the previous example, g_1 simply maps a geometric unit circle to the unit circle in the complex plane, but this idea generalises...
- I have two nontrivial examples: (i) Ω is the Lorenz attractor and Φ is a time-t map of the Lorenz flow; (ii) Ω is the collection of climate configurations and Φ is a monthly evolution of the climate.
- For dynamics (Ω, Φ), an eigenfunction $g : \Omega \to \mathbb{C}$ of $\mathcal{P}$ with eigenvalue Λ ($|\Lambda| = 1$) will **project (semi-conjugate)** [Halmos–von Neumann] the dynamics (Ω, Φ) with a rigid rotation on the unit circle in $\mathbb{C}$ by an amount $\arg(\Lambda)$; the latter represents **a particular cycle** in the Φ-system.
- For $|\Lambda| = 1$ let $M_\Lambda : \mathbb{C} \to \mathbb{C}$ be the multiplication-by-Λ dynamics in the complex plane: $M_\Lambda z = \Lambda z = e^{i \arg(\Lambda)} z$; a rotation by $\arg(\Lambda)$.
- The semi-conjugacy diagram is:

$$\begin{array}{ccc} \Omega & \xrightarrow{\Phi} & \Omega \\ \downarrow g & & \downarrow g \\ \mathbb{C} & \xrightarrow{M_\Lambda} & \mathbb{C} \end{array}$$

Transfer operator spectrum reveals angular rotation

- In the previous example, g_1 simply maps a geometric unit circle to the unit circle in the complex plane, but this idea generalises...
- I have two nontrivial examples: (i) Ω is the Lorenz attractor and Φ is a time-t map of the Lorenz flow; (ii) Ω is the collection of climate configurations and Φ is a monthly evolution of the climate.
- For dynamics (Ω, Φ), an eigenfunction $g : \Omega \to \mathbb{C}$ of $\mathcal{P}$ with eigenvalue Λ ($|\Lambda| = 1$) will **project (semi-conjugate)** [Halmos–von Neumann] the dynamics (Ω, Φ) with a rigid rotation on the unit circle in $\mathbb{C}$ by an amount $\arg(\Lambda)$; the latter represents **a particular cycle** in the Φ-system.
- For $|\Lambda| = 1$ let $M_\Lambda : \mathbb{C} \to \mathbb{C}$ be the multiplication-by-Λ dynamics in the complex plane: $M_\Lambda z = \Lambda z = e^{i \arg(\Lambda)} z$; a rotation by $\arg(\Lambda)$.
- The semi-conjugacy diagram is:

$$\begin{array}{ccc} \Omega & \xrightarrow{\Phi} & \Omega \\ \downarrow g & & \downarrow g \\ \mathbb{C} & \xrightarrow{M_\Lambda} & \mathbb{C} \end{array}$$

Transfer operator spectrum reveals angular rotation

- In the previous example, g_1 simply maps a geometric unit circle to the unit circle in the complex plane, but this idea generalises...
- I have two nontrivial examples: (i) Ω is the Lorenz attractor and Φ is a time-t map of the Lorenz flow; (ii) Ω is the collection of climate configurations and Φ is a monthly evolution of the climate.
- For dynamics (Ω, Φ), an eigenfunction $g : \Omega \to \mathbb{C}$ of $\mathcal{P}$ with eigenvalue Λ ($|\Lambda| = 1$) will **project (semi-conjugate)** [Halmos–von Neumann] the dynamics (Ω, Φ) with a rigid rotation on the unit circle in $\mathbb{C}$ by an amount $\arg(\Lambda)$; the latter represents **a particular cycle** in the Φ-system.
- For $|\Lambda| = 1$ let $M_\Lambda : \mathbb{C} \to \mathbb{C}$ be the multiplication-by-Λ dynamics in the complex plane: $M_\Lambda z = \Lambda z = e^{i \arg(\Lambda)} z$; a rotation by $\arg(\Lambda)$.
- The semi-conjugacy diagram is:

$$\begin{array}{ccc} \Omega & \xrightarrow{\Phi} & \Omega \\ \downarrow g & & \downarrow g \\ \mathbb{C} & \xrightarrow{M_\Lambda} & \mathbb{C} \end{array}$$

Transfer operator spectrum reveals angular rotation

- In the previous example, g_1 simply maps a geometric unit circle to the unit circle in the complex plane, but this idea generalises...
- I have two nontrivial examples: (i) Ω is the Lorenz attractor and Φ is a time-t map of the Lorenz flow; (ii) Ω is the collection of climate configurations and Φ is a monthly evolution of the climate.
- For dynamics (Ω, Φ), an eigenfunction $g : \Omega \to \mathbb{C}$ of $\mathcal{P}$ with eigenvalue Λ ($|\Lambda| = 1$) will **project (semi-conjugate)** [Halmos–von Neumann] the dynamics (Ω, Φ) with a rigid rotation on the unit circle in $\mathbb{C}$ by an amount $\arg(\Lambda)$; the latter represents **a particular cycle** in the Φ-system.
- For $|\Lambda| = 1$ let $M_\Lambda : \mathbb{C} \to \mathbb{C}$ be the multiplication-by-Λ dynamics in the complex plane: $M_\Lambda z = \Lambda z = e^{i \arg(\Lambda)} z$; a rotation by $\arg(\Lambda)$.
- The semi-conjugacy diagram is:

$$\begin{array}{ccc} \Omega & \xrightarrow{\Phi} & \Omega \\ \downarrow{\scriptstyle g} & & \downarrow{\scriptstyle g} \\ \mathbb{C} & \xrightarrow{M_\Lambda} & \mathbb{C} \end{array}$$

Transfer operator spectrum reveals angular rotation

- In the previous example, g_1 simply maps a geometric unit circle to the unit circle in the complex plane, but this idea generalises...
- I have two nontrivial examples: (i) Ω is the Lorenz attractor and Φ is a time-t map of the Lorenz flow; (ii) Ω is the collection of climate configurations and Φ is a monthly evolution of the climate.
- For dynamics (Ω, Φ), an eigenfunction $g : \Omega \to \mathbb{C}$ of $\mathcal{P}$ with eigenvalue Λ ($|\Lambda| = 1$) will **project (semi-conjugate)** [Halmos–von Neumann] the dynamics (Ω, Φ) with a rigid rotation on the unit circle in $\mathbb{C}$ by an amount $\arg(\Lambda)$; the latter represents **a particular cycle** in the Φ-system.
- For $|\Lambda| = 1$ let $M_\Lambda : \mathbb{C} \to \mathbb{C}$ be the multiplication-by-Λ dynamics in the complex plane: $M_\Lambda z = \Lambda z = e^{i \arg(\Lambda)} z$; a rotation by $\arg(\Lambda)$.
- The semi-conjugacy diagram is:

$$\begin{array}{ccc} \Omega & \xrightarrow{\Phi} & \Omega \\ \Big\downarrow g & & \Big\downarrow g \\ \mathbb{C} & \xrightarrow{M_\Lambda} & \mathbb{C} \end{array}$$

Numerics in brief

- We use a simple Gaussian collocation approach to estimate $\mathcal{P}$.
- To sketch this, suppose we have a data trajectory $x_i \in \Omega, i = 1, \ldots, N$.
- We form Gaussian kernels $k_\epsilon(x_i, y) = \exp(-\|x_i - y\|^2/\epsilon^2)$, centered on each data point x_i, for bandwidth parameter ϵ.
- We estimate the transfer operator $\mathcal{P}$ using the $N \times N$ stochastic matrix

$$P_{ij} = \frac{k_\epsilon(x_i, x_{j+1})}{\sum_{i'} k_\epsilon(x_{i'}, x_{j+1})}.$$

- Because the matrix P is nonnegative with all columns summing to 1, the eigenvalues of P are constrained to lie in the unit circle in $\mathbb{C}$ and for small ϵ, we expect many eigenvalues very near to the unit circle.
- The nontrivial eigenvalues of P of **greatest magnitude** will **automatically identify the rotation angle** α.
- Related ideas in [Dellnitz/Junge'99] (Λ a root of unity), and [F/González-Tokman/Quas'14, Eisner *et al.*'15, F/Koltai'17].

Numerics in brief

- We use a simple Gaussian collocation approach to estimate $\mathcal{P}$.
- To sketch this, suppose we have a data trajectory $x_i \in \Omega, i = 1, \ldots, N$.
- We form Gaussian kernels $k_\epsilon(x_i, y) = \exp(-\|x_i - y\|^2/\epsilon^2)$, centered on each data point x_i, for bandwidth parameter ϵ.
- We estimate the transfer operator $\mathcal{P}$ using the $N \times N$ stochastic matrix

$$P_{ij} = \frac{k_\epsilon(x_i, x_{j+1})}{\sum_{i'} k_\epsilon(x_{i'}, x_{j+1})}.$$

- Because the matrix P is nonnegative with all columns summing to 1, the eigenvalues of P are constrained to lie in the unit circle in $\mathbb{C}$ and for small ϵ, we expect many eigenvalues very near to the unit circle.
- The nontrivial eigenvalues of P of **greatest magnitude** will **automatically identify the rotation angle** α.
- Related ideas in [Dellnitz/Junge'99] (Λ a root of unity), and [F/González-Tokman/Quas'14, Eisner *et al.*'15, F/Koltai'17].

Numerics in brief

- We use a simple Gaussian collocation approach to estimate $\mathcal{P}$.
- To sketch this, suppose we have a data trajectory $x_i \in \Omega, i = 1, \ldots, N$.
- We form Gaussian kernels $k_\epsilon(x_i, y) = \exp(-\|x_i - y\|^2/\epsilon^2)$, centered on each data point x_i, for bandwidth parameter ϵ.
- We estimate the transfer operator $\mathcal{P}$ using the $N \times N$ stochastic matrix

$$P_{ij} = \frac{k_\epsilon(x_i, x_{j+1})}{\sum_{i'} k_\epsilon(x_{i'}, x_{j+1})}.$$

- Because the matrix P is nonnegative with all columns summing to 1, the eigenvalues of P are constrained to lie in the unit circle in $\mathbb{C}$ and for small ϵ, we expect many eigenvalues very near to the unit circle.
- The nontrivial eigenvalues of P of **greatest magnitude** will **automatically identify the rotation angle** α.
- Related ideas in [Dellnitz/Junge'99] (Λ a root of unity), and [F/González-Tokman/Quas'14, Eisner *et al.*'15, F/Koltai'17].

Numerics in brief

- We use a simple Gaussian collocation approach to estimate $\mathcal{P}$.
- To sketch this, suppose we have a data trajectory $x_i \in \Omega, i = 1, \ldots, N$.
- We form Gaussian kernels $k_\epsilon(x_i, y) = \exp(-\|x_i - y\|^2/\epsilon^2)$, centered on each data point x_i, for bandwidth parameter ϵ.
- We estimate the transfer operator $\mathcal{P}$ using the $N \times N$ stochastic matrix

$$P_{ij} = \frac{k_\epsilon(x_i, x_{j+1})}{\sum_{i'} k_\epsilon(x_{i'}, x_{j+1})}.$$

- Because the matrix P is nonnegative with all columns summing to 1, the eigenvalues of P are constrained to lie in the unit circle in $\mathbb{C}$ and for small ϵ, we expect many eigenvalues very near to the unit circle.
- The nontrivial eigenvalues of P of **greatest magnitude** will **automatically identify the rotation angle** α.
- Related ideas in [Dellnitz/Junge'99] (Λ a root of unity), and [F/González-Tokman/Quas'14, Eisner *et al.*'15, F/Koltai'17].

Numerics in brief

- We use a simple Gaussian collocation approach to estimate $\mathcal{P}$.
- To sketch this, suppose we have a data trajectory $x_i \in \Omega, i = 1, \ldots, N$.
- We form Gaussian kernels $k_\epsilon(x_i, y) = \exp(-\|x_i - y\|^2/\epsilon^2)$, centered on each data point x_i, for bandwidth parameter ϵ.
- We estimate the transfer operator $\mathcal{P}$ using the $N \times N$ stochastic matrix

$$P_{ij} = \frac{k_\epsilon(x_i, x_{j+1})}{\sum_{i'} k_\epsilon(x_{i'}, x_{j+1})}.$$

- Because the matrix P is nonnegative with all columns summing to 1, the eigenvalues of P are constrained to lie in the unit circle in $\mathbb{C}$ and for small ϵ, we expect many eigenvalues very near to the unit circle.
- The nontrivial eigenvalues of P of **greatest magnitude** will **automatically identify the rotation angle** α.
- Related ideas in [Dellnitz/Junge'99] (Λ a root of unity), and [F/González-Tokman/Quas'14, Eisner *et al.*'15, F/Koltai'17].

Numerics in brief

- We use a simple Gaussian collocation approach to estimate $\mathcal{P}$.
- To sketch this, suppose we have a data trajectory $x_i \in \Omega, i = 1, \ldots, N$.
- We form Gaussian kernels $k_\epsilon(x_i, y) = \exp(-\|x_i - y\|^2/\epsilon^2)$, centered on each data point x_i, for bandwidth parameter ϵ.
- We estimate the transfer operator $\mathcal{P}$ using the $N \times N$ stochastic matrix

$$P_{ij} = \frac{k_\epsilon(x_i, x_{j+1})}{\sum_{i'} k_\epsilon(x_{i'}, x_{j+1})}.$$

- Because the matrix P is nonnegative with all columns summing to 1, the eigenvalues of P are constrained to lie in the unit circle in $\mathbb{C}$ and for small ϵ, we expect many eigenvalues very near to the unit circle.
- The nontrivial eigenvalues of P of **greatest magnitude** will **automatically identify the rotation angle** α.
- Related ideas in [Dellnitz/Junge'99] (Λ a root of unity), and [F/González-Tokman/Quas'14, Eisner *et al.*'15, F/Koltai'17].

Numerics in brief

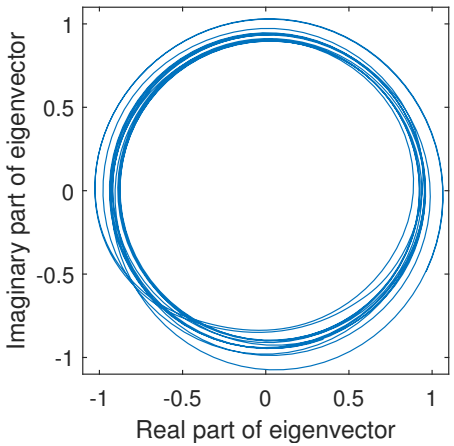
- We use a simple Gaussian collocation approach to estimate $\mathcal{P}$.
- To sketch this, suppose we have a data trajectory $x_i \in \Omega, i = 1, \ldots, N$.
- We form Gaussian kernels $k_\epsilon(x_i, y) = \exp(-\|x_i - y\|^2/\epsilon^2)$, centered on each data point x_i, for bandwidth parameter ϵ.
- We estimate the transfer operator $\mathcal{P}$ using the $N \times N$ stochastic matrix

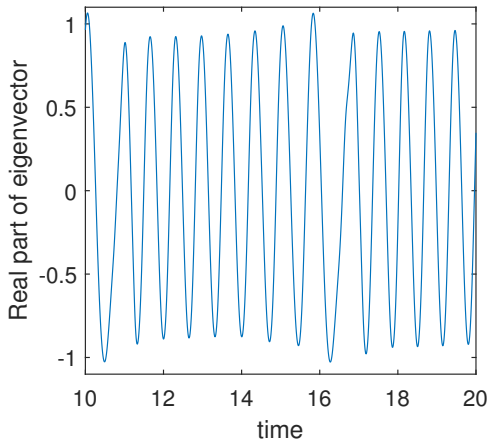
$$P_{ij} = \frac{k_\epsilon(x_i, x_{j+1})}{\sum_{i'} k_\epsilon(x_{i'}, x_{j+1})}.$$

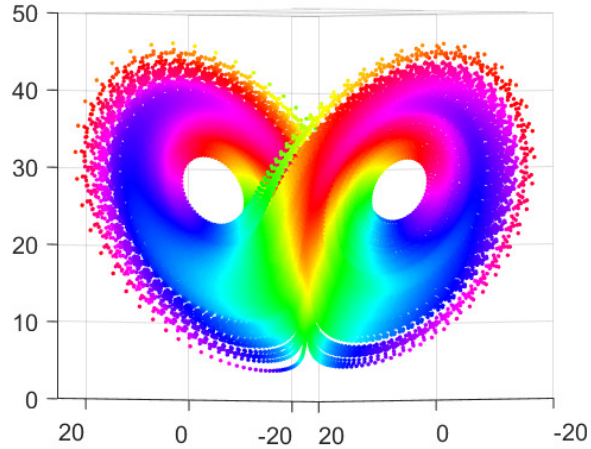
- Because the matrix P is nonnegative with all columns summing to 1, the eigenvalues of P are constrained to lie in the unit circle in $\mathbb{C}$ and for small ϵ, we expect many eigenvalues very near to the unit circle.
- The nontrivial eigenvalues of P of **greatest magnitude** will **automatically identify the rotation angle** α.
- Related ideas in [Dellnitz/Junge'99] (Λ a root of unity), and [F/González-Tokman/Quas'14, Eisner *et al.*'15, F/Koltai'17].

Illustration with Lorenz

- We made a transfer operator approximation P from a Lorenz trajectory of length 160, sampled every 0.01 time units and $\epsilon = 0.5$.
- The largest-magnitude nontrivial eigenvalue $\Lambda_1 \approx 0.9961 e^{0.013 \cdot (2\pi i)}$, corresponding to a rotation period of $1/0.013 \approx 77$ time steps of 0.01, or 0.77 time units.
- Below (left): The corresponding complex eigenvector (projection onto dominant cycle) g_1 **plotted along the data trajectory**.
- Below (right): Same as above, but now the argument of the complex eigenvector (projection) g_1 is given by colour, is **plotted on the attractor**, rather than along the data trajectory.







From [F/Giannakis/Lintner/Pike/Slawinska, *Nature Comm.*'21]

Evolving the complex eigenfunction (argument displayed)

Constructing the transfer operator from SST fields

- For observational data, we use monthly averaged, 2° SST fields from the Extended Reconstructed Sea Surface Temperature Version 4 (ERSSTv4) reanalysis product [Huang et al.'14] from 1970 to 2020.
- The number of pixels in the SST image is $d = 4868$, and we use denote by $y_t \in \mathbb{R}^d$ the d-vector of the Indo-Pacific SST values at the t-th month, for $t = 1, \ldots, T$ in months, where $T = 600$ (50 years).
- To improve state estimation from the observations y_t, we create a Takens delay vector $x_t = [y_{t-12}, y_t]$ with a single delay of 12 months (approximately one-quarter the cycle period).
- Using Gaussian kernels, we estimate $\mathcal{P}$ (stepping one month ahead) and **compute leading eigenvalues $\Lambda \in \mathbb{C}$ of P, obtaining a cycle length of $\sim$47 months.**

Constructing the transfer operator from SST fields

- For observational data, we use monthly averaged, 2° SST fields from the Extended Reconstructed Sea Surface Temperature Version 4 (ERSSTv4) reanalysis product [Huang et al.'14] from 1970 to 2020.
- The number of pixels in the SST image is $d = 4868$, and we use denote by $y_t \in \mathbb{R}^d$ the d-vector of the Indo-Pacific SST values at the t-th month, for $t = 1, \ldots, T$ in months, where $T = 600$ (50 years).
- To improve state estimation from the observations y_t, we create a Takens delay vector $x_t = [y_{t-12}, y_t]$ with a single delay of 12 months (approximately one-quarter the cycle period).
- Using Gaussian kernels, we estimate $\mathcal{P}$ (stepping one month ahead) and **compute leading eigenvalues $\Lambda \in \mathbb{C}$ of P, obtaining a cycle length of $\sim$47 months.**

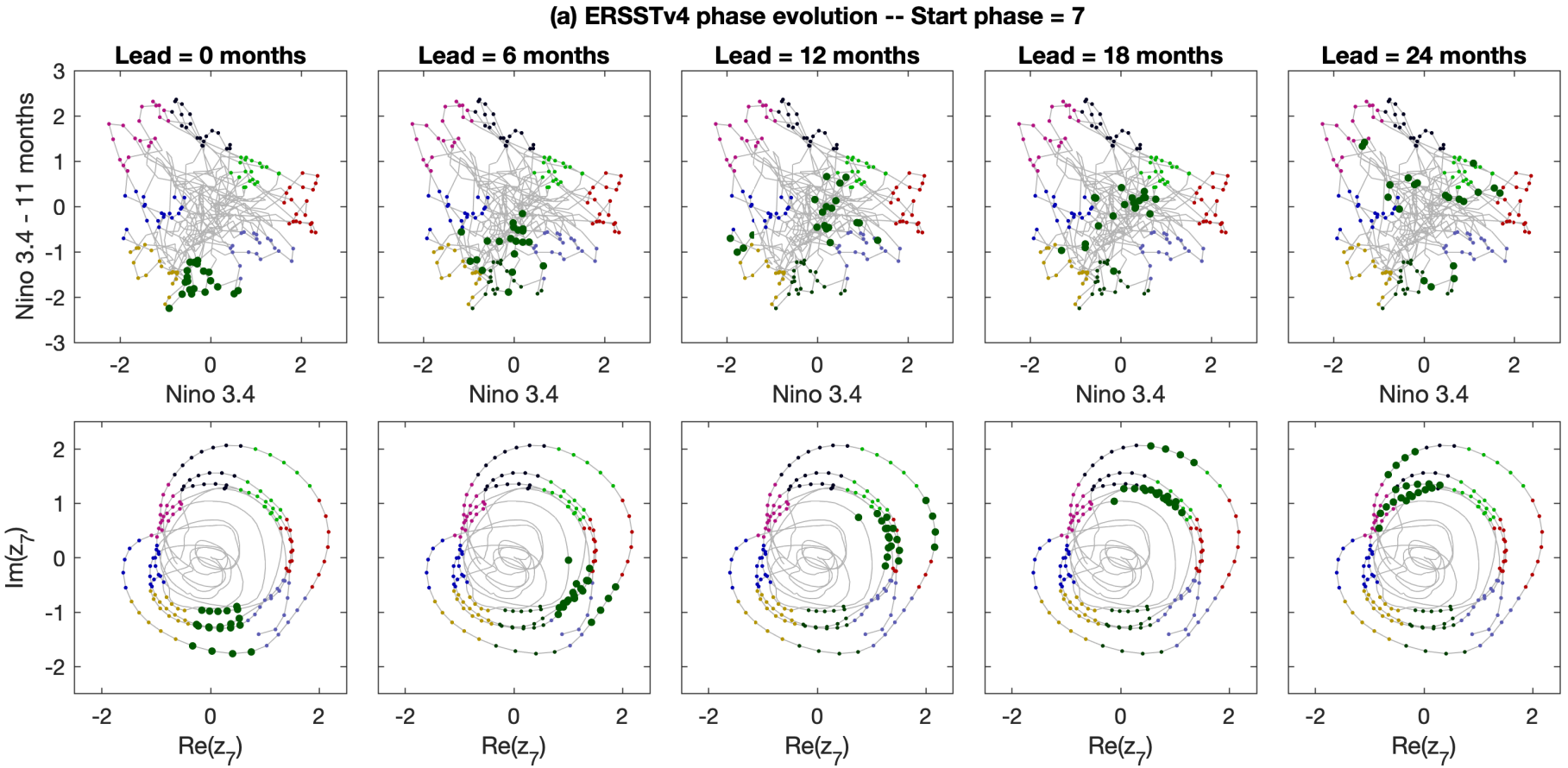
Constructing the transfer operator from SST fields

- For observational data, we use monthly averaged, 2° SST fields from the Extended Reconstructed Sea Surface Temperature Version 4 (ERSSTv4) reanalysis product [Huang et al.'14] from 1970 to 2020.
- The number of pixels in the SST image is $d = 4868$, and we use denote by $y_t \in \mathbb{R}^d$ the d-vector of the Indo-Pacific SST values at the t-th month, for $t = 1, \ldots, T$ in months, where $T = 600$ (50 years).
- To improve state estimation from the observations y_t, we create a Takens delay vector $x_t = [y_{t-12}, y_t]$ with a single delay of 12 months (approximately one-quarter the cycle period).
- Using Gaussian kernels, we estimate $\mathcal{P}$ (stepping one month ahead) and **compute leading eigenvalues $\Lambda \in \mathbb{C}$ of P, obtaining a cycle length of $\sim$47 months.**

Constructing the transfer operator from SST fields

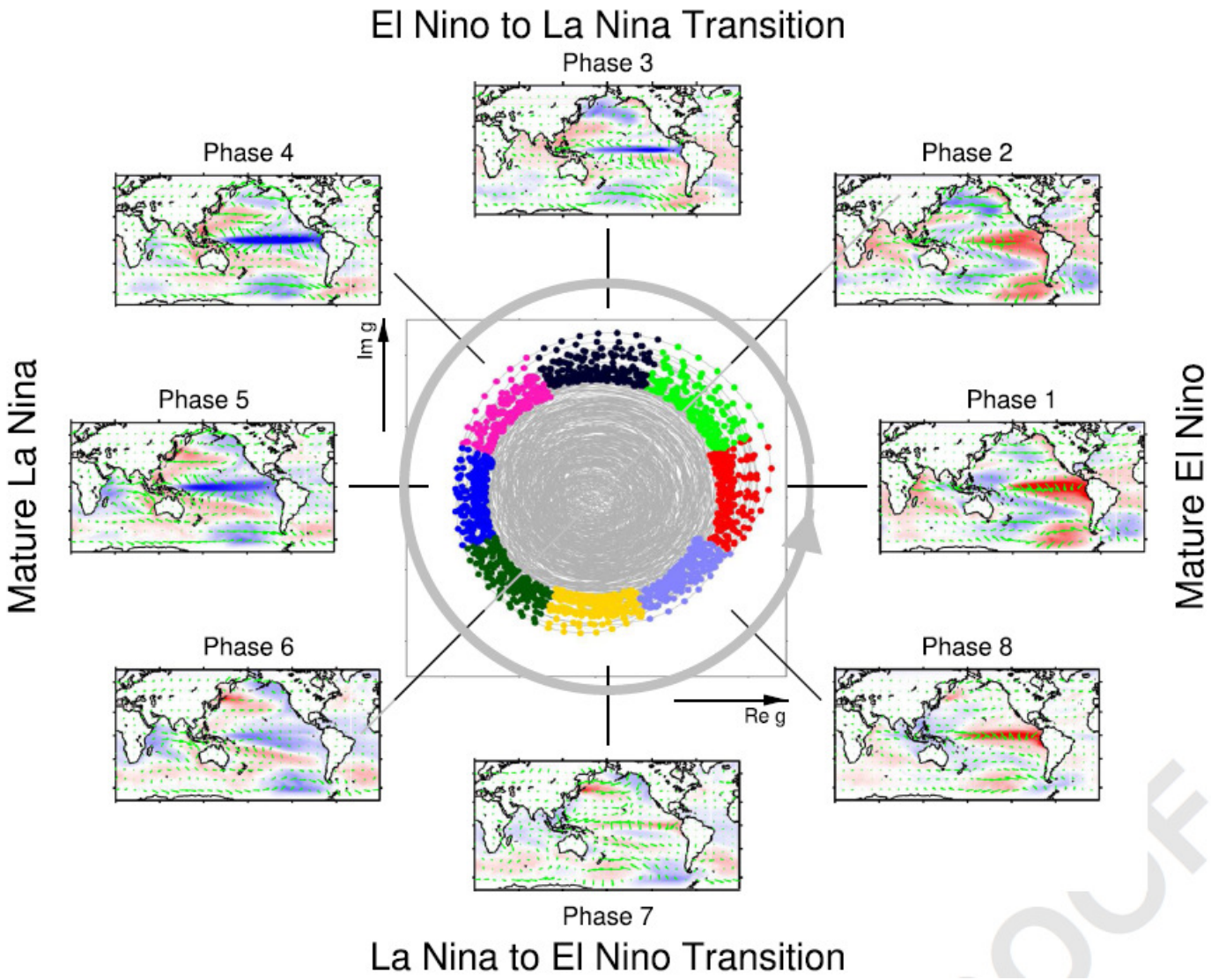
- For observational data, we use monthly averaged, 2° SST fields from the Extended Reconstructed Sea Surface Temperature Version 4 (ERSSTv4) reanalysis product [Huang et al.'14] from 1970 to 2020.
- The number of pixels in the SST image is $d = 4868$, and we use denote by $y_t \in \mathbb{R}^d$ the d-vector of the Indo-Pacific SST values at the t-th month, for $t = 1, \ldots, T$ in months, where $T = 600$ (50 years).
- To improve state estimation from the observations y_t, we create a Takens delay vector $x_t = [y_{t-12}, y_t]$ with a single delay of 12 months (approximately one-quarter the cycle period).
- Using Gaussian kernels, we estimate $\mathcal{P}$ (stepping one month ahead) and **compute leading eigenvalues $\Lambda \in \mathbb{C}$ of P, obtaining a cycle length of $\sim$47 months.**

How cyclic is ENSO?



- The phases in the upper row (lagged Niño 3.4 index) **almost randomly evolve**.
- The phases in the lower row (our eigenfunction-based phases) **neatly cycle from one phase to the next**.

El Niño—La Niña oscillation as a complex eigenfunction



Input spatio-temporal data

Build transfer/Koopman operator or generator.

Compute leading eigenvalues (Fig. 3) and eigenvectors.

Provides a projection onto the **dominant cycle** and the **cycle frequency**. The cycle is **rectified** to proceed at **constant speed** (Fig. 5).

Subdivide the extracted cycle into **phases** – **see Fig. 1 (center).** These phases are **equivariant** and **respect the cyclic dynamics** (Fig. 9).

Produces a cycle of **spatio-temporal composites**, e.g. for SST – **see Fig. 1 (outer panels).**

Summary – Part 1

- Developed a *general-purpose methodology to identify persistent cycles in high-dimensional spatio-temporal data* (or low-dimensional, like Lorenz).
- We applied this framework to **extract a canonical ENSO cycle from SST observations**.
- Our ENSO phases **much more cyclic** than those based on the standard Niño 3.4 Index; this suggests that ENSO is **more predictable than previously thought**.
- Because of our "rectification" of the cycle, we obtain **greater resolution in the transition from La Niña to El Niño**, which is difficult characterise and predict.
- Our ENSO phases **also capture independent weather/climate measurements** such as wind speed/direction, surface air temperature, and precipitation. These **independent measurements also match well with what is expected from ENSO events**.
- For these reasons, we argue that our phases are a **superior, canonical way of characterising the ENSO cycle**.
- See [F/Giannakis/Lintner/Pike/Slawinska, *Nature Comm.*'21].

Summary – Part 1

- Developed a *general-purpose methodology to identify persistent cycles in high-dimensional spatio-temporal data* (or low-dimensional, like Lorenz).
- We applied this framework to **extract a canonical ENSO cycle from SST observations**.
- Our ENSO phases **much more cyclic** than those based on the standard Niño 3.4 Index; this suggests that ENSO is **more predictable than previously thought**.
- Because of our "rectification" of the cycle, we obtain **greater resolution in the transition from La Niña to El Niño**, which is difficult characterise and predict.
- Our ENSO phases **also capture independent weather/climate measurements** such as wind speed/direction, surface air temperature, and precipitation. These **independent measurements also match well with what is expected from ENSO events**.
- For these reasons, we argue that our phases are a **superior, canonical way of characterising the ENSO cycle**.
- See [F/Giannakis/Lintner/Pike/Slawinska, *Nature Comm.*'21].

Summary – Part 1

- Developed a *general-purpose methodology to identify persistent cycles in high-dimensional spatio-temporal data* (or low-dimensional, like Lorenz).
- We applied this framework to **extract a canonical ENSO cycle from SST observations**.
- Our ENSO phases **much more cyclic** than those based on the standard Niño 3.4 Index; this suggests that ENSO is **more predictable than previously thought**.
- Because of our "rectification" of the cycle, we obtain **greater resolution in the transition from La Niña to El Niño**, which is difficult characterise and predict.
- Our ENSO phases **also capture independent weather/climate measurements** such as wind speed/direction, surface air temperature, and precipitation. These **independent measurements also match well with what is expected from ENSO events**.
- For these reasons, we argue that our phases are a **superior, canonical way of characterising the ENSO cycle**.
- See [F/Giannakis/Lintner/Pike/Slawinska, *Nature Comm.*'21].

Summary – Part 1

- Developed a *general-purpose methodology to identify persistent cycles in high-dimensional spatio-temporal data* (or low-dimensional, like Lorenz).
- We applied this framework to **extract a canonical ENSO cycle from SST observations**.
- Our ENSO phases **much more cyclic** than those based on the standard Niño 3.4 Index; this suggests that ENSO is **more predictable than previously thought**.
- Because of our "rectification" of the cycle, we obtain **greater resolution in the transition from La Niña to El Niño**, which is difficult characterise and predict.
- Our ENSO phases **also capture independent weather/climate measurements** such as wind speed/direction, surface air temperature, and precipitation. These **independent measurements also match well with what is expected from ENSO events**.
- For these reasons, we argue that our phases are a **superior, canonical way of characterising the ENSO cycle**.
- See [F/Giannakis/Lintner/Pike/Slawinska, *Nature Comm.*'21].

Summary – Part 1

- Developed a *general-purpose methodology to identify persistent cycles in high-dimensional spatio-temporal data* (or low-dimensional, like Lorenz).
- We applied this framework to **extract a canonical ENSO cycle from SST observations**.
- Our ENSO phases **much more cyclic** than those based on the standard Niño 3.4 Index; this suggests that ENSO is **more predictable than previously thought**.
- Because of our "rectification" of the cycle, we obtain **greater resolution in the transition from La Niña to El Niño**, which is difficult characterise and predict.
- Our ENSO phases **also capture independent weather/climate measurements** such as wind speed/direction, surface air temperature, and precipitation. These **independent measurements also match well with what is expected from ENSO events**.
- For these reasons, we argue that our phases are a **superior, canonical way of characterising the ENSO cycle**.
- See [F/Giannakis/Lintner/Pike/Slawinska, *Nature Comm.*'21].

Summary – Part 1

- Developed a *general-purpose methodology to identify persistent cycles in high-dimensional spatio-temporal data* (or low-dimensional, like Lorenz).
- We applied this framework to **extract a canonical ENSO cycle from SST observations**.
- Our ENSO phases **much more cyclic** than those based on the standard Niño 3.4 Index; this suggests that ENSO is **more predictable than previously thought**.
- Because of our "rectification" of the cycle, we obtain **greater resolution in the transition from La Niña to El Niño**, which is difficult characterise and predict.
- Our ENSO phases **also capture independent weather/climate measurements** such as wind speed/direction, surface air temperature, and precipitation. These **independent measurements also match well with what is expected from ENSO events**.
- For these reasons, we argue that our phases are a **superior, canonical way of characterising the ENSO cycle**.
- See [F/Giannakis/Lintner/Pike/Slawinska, *Nature Comm.*'21].

Summary – Part 1

- Developed a *general-purpose methodology to identify persistent cycles in high-dimensional spatio-temporal data* (or low-dimensional, like Lorenz).
- We applied this framework to **extract a canonical ENSO cycle from SST observations**.
- Our ENSO phases **much more cyclic** than those based on the standard Niño 3.4 Index; this suggests that ENSO is **more predictable than previously thought**.
- Because of our "rectification" of the cycle, we obtain **greater resolution in the transition from La Niña to El Niño**, which is difficult characterise and predict.
- Our ENSO phases **also capture independent weather/climate measurements** such as wind speed/direction, surface air temperature, and precipitation. These **independent measurements also match well with what is expected from ENSO events**.
- For these reasons, we argue that our phases are a **superior, canonical way of characterising the ENSO cycle**.
- See [F/Giannakis/Lintner/Pike/Slawinska, *Nature Comm.*'21].

PART II

THE BIRTH AND DEATH OF FINITE-TIME COHERENT SETS

Finite-time coherent sets

- Finite-time coherent sets have been successful at describing the **predictable backbone** of nonautonomous dynamical systems.
- If coherent sets **last throughout** a flow duration from time 0 to time τ, then all standard transfer operator [F'13] and dynamic Laplacian [F15,F/Junge'18] approaches to computing coherent sets will output a family of sets $\{A_t\}_{t=0}^{\tau} \subset M$ representing the location of a particular coherent set at time t.

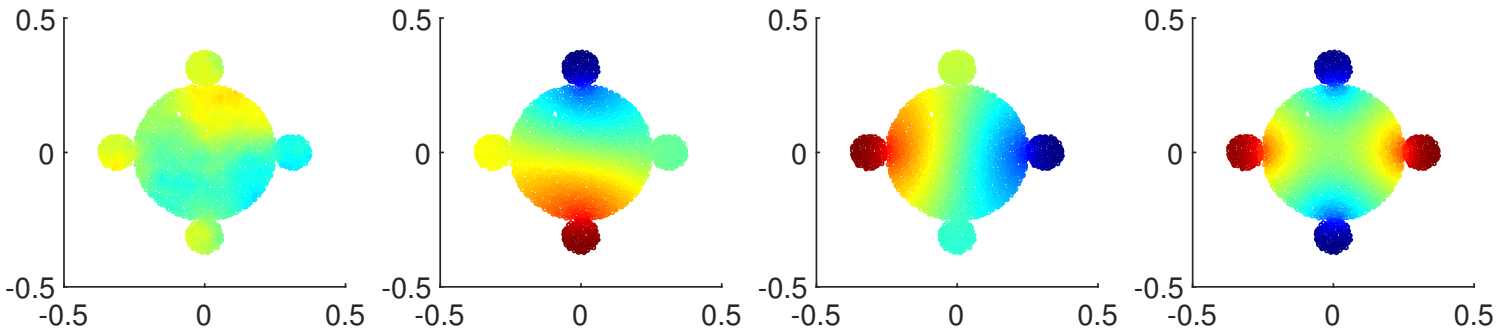
- Later in the talk we will discuss the situation where there are coherent sets that exist for only **part of** the duration $[0, \tau]$.

Setup and notation

- Let $v : [0, \tau] \times \mathbb{R}^d \to \mathbb{R}^d$ denote a time-dependent velocity field, over a finite time duration $[0, \tau]$.
- For simplicity we assume that $v(t, \cdot)$ is divergence free for all $t \in [0, \tau]$.
- Denote by $M \subset \mathbb{R}^d$ a full-dimensional, connected, compact submanifold of $\mathbb{R}^d$, representing the phase space at time $t = 0$.
- Let $\phi_t : M \to \phi_t(M)$ be the flow map generated by v from time 0 to time t, $t \in [0, \tau]$.

How to find these coherent sets?

- In classical isoperimetric theory [Chavel'84] it is known that **leading eigenfunctions** of the Laplace-Beltrami operator **encode geometric information** about a manifold and tend to "localise" (e.g. [Grebenkov/Nguyen'13]) on "poorly connected" regions, or regions from which heat slowly escapes under heat flow on the manifold.



- This idea and static isoperimetric theory can be extended to the situation of a manifold **evolving under general nonlinear dynamics**, using a *dynamic* Laplace operator [F'15,F/Kwok'20].
- The eigenfunctions of the dynamic Laplace operator will identify sets that have **small evolving boundary** relative to enclosed volume.

The dynamic Laplacian

- The dynamic Laplacian Δ^D on $L^2(M)$ has the form [F'15]:

$$\Delta^D = \frac{1}{\tau}\int_0^\tau \phi_t \circ \Delta_{\phi_t(M)} \circ \phi_{-t}\, dt = \frac{1}{\tau}\int_0^\tau \Delta_{\phi_t^* e}\, dt = \frac{1}{\tau}\int_0^\tau \Delta_{g_t}\, dt,$$

  where $g_t := \phi_t^* e$ denotes the pullback of the Euclidean metric e from the manifold $\phi_t(M)$ to the manifold M.

- The local coordinate representation of $g_t(x)$ is the $d \times d$ matrix $D\phi_t(x)^\top D\phi_t(x)$.
- Thus Δ^D is an **average of Laplace–Beltrami operators** for the Riemannian manifolds (M, g_t), $t \in [0, \tau]$.
- **The dominant coherent sets are encoded in the leading eigenfunctions of Δ^D.**

The dynamic Laplacian

- The dynamic Laplacian Δ^D on $L^2(M)$ has the form [F'15]:

$$\Delta^D = \frac{1}{\tau}\int_0^\tau \phi_t \circ \Delta_{\phi_t(M)} \circ \phi_{-t}\, dt = \frac{1}{\tau}\int_0^\tau \Delta_{\phi_t^* e}\, dt = \frac{1}{\tau}\int_0^\tau \Delta_{g_t}\, dt,$$

  where $g_t := \phi_t^* e$ denotes the pullback of the Euclidean metric e from the manifold $\phi_t(M)$ to the manifold M.
- The local coordinate representation of $g_t(x)$ is the $d \times d$ matrix $D\phi_t(x)^\top D\phi_t(x)$.
- Thus Δ^D is an **average of Laplace–Beltrami operators** for the Riemannian manifolds (M, g_t), $t \in [0, \tau]$.
- **The dominant coherent sets are encoded in the leading eigenfunctions of Δ^D.**

The dynamic Laplacian

- The dynamic Laplacian Δ^D on $L^2(M)$ has the form [F'15]:

$$\Delta^D = \frac{1}{\tau}\int_0^\tau \phi_t \circ \Delta_{\phi_t(M)} \circ \phi_{-t}\, dt = \frac{1}{\tau}\int_0^\tau \Delta_{\phi_t^* e}\, dt = \frac{1}{\tau}\int_0^\tau \Delta_{g_t}\, dt,$$

  where $g_t := \phi_t^* e$ denotes the pullback of the Euclidean metric e from the manifold $\phi_t(M)$ to the manifold M.
- The local coordinate representation of $g_t(x)$ is the $d \times d$ matrix $D\phi_t(x)^\top D\phi_t(x)$.
- Thus Δ^D is an **average of Laplace–Beltrami operators** for the Riemannian manifolds (M, g_t), $t \in [0, \tau]$.
- **The dominant coherent sets are encoded in the leading eigenfunctions of Δ^D.**

The dynamic Laplacian

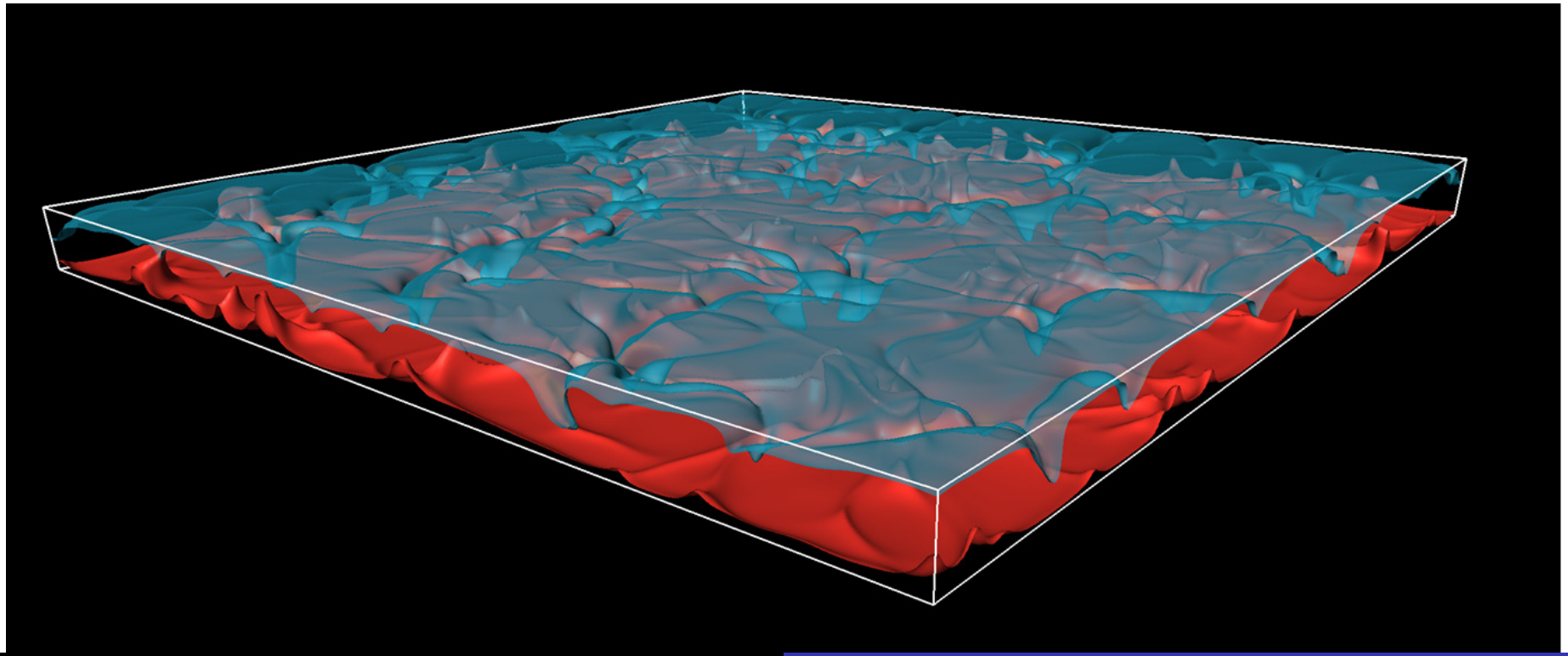
- The dynamic Laplacian Δ^D on $L^2(M)$ has the form [F'15]:

$$\Delta^D = \frac{1}{\tau}\int_0^\tau \phi_t \circ \Delta_{\phi_t(M)} \circ \phi_{-t}\, dt = \frac{1}{\tau}\int_0^\tau \Delta_{\phi_t^* e}\, dt = \frac{1}{\tau}\int_0^\tau \Delta_{g_t}\, dt,$$

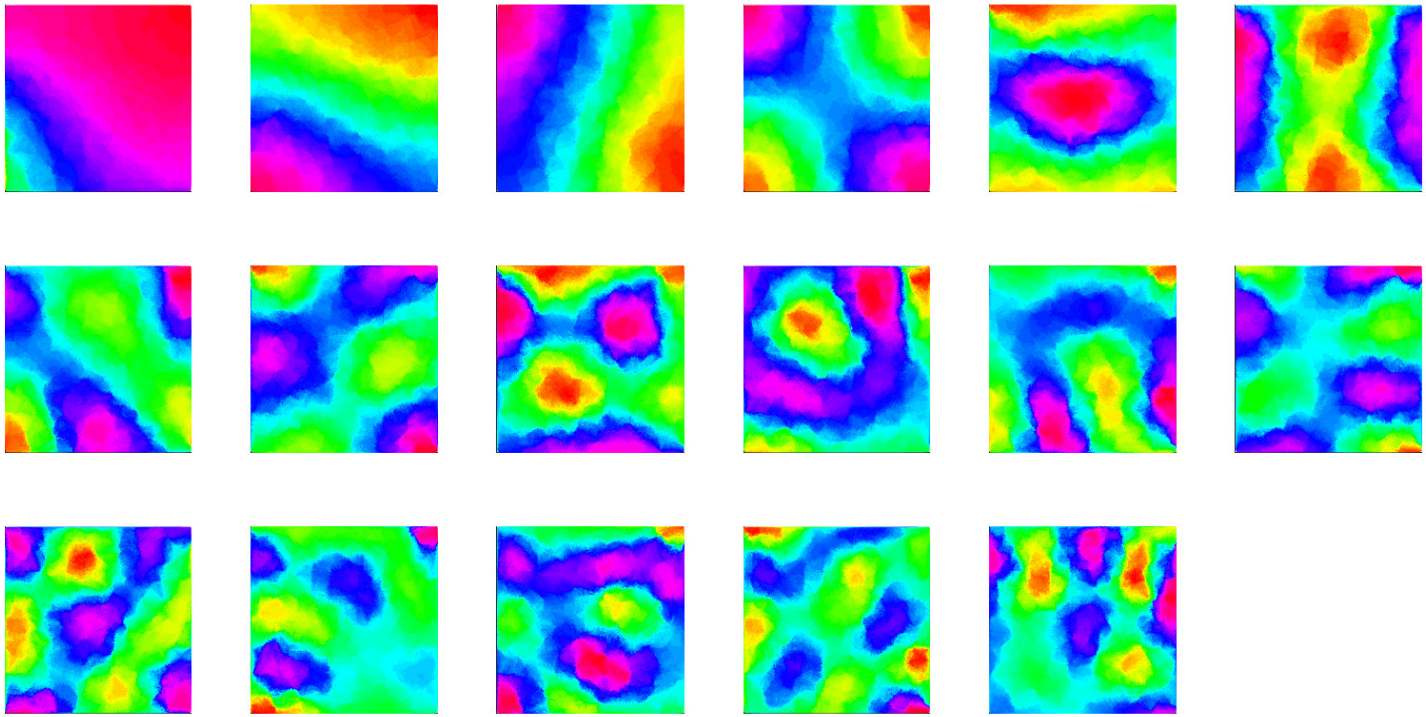
  where $g_t := \phi_t^* e$ denotes the pullback of the Euclidean metric e from the manifold $\phi_t(M)$ to the manifold M.
- The local coordinate representation of $g_t(x)$ is the $d \times d$ matrix $D\phi_t(x)^\top D\phi_t(x)$.
- Thus Δ^D is an **average of Laplace–Beltrami operators** for the Riemannian manifolds (M, g_t), $t \in [0, \tau]$.
- **The dominant coherent sets are encoded in the leading eigenfunctions of Δ^D.**

Rayleigh–Bénard convection setup

- Domain aspect ratio is 16:16:1.
- Nominal temperature range $0 \leq T \leq 1$, with the **domain floor held at $T = 1$ (hot)** and the **domain ceiling held at $T = 0$ (cold)**.
- We seek to **identify the turbulent superstructures as finite-time coherent sets** and to this end compute trajectories of 40000 tracers.
- We use a time duration of $\tau := 10.5 t_f$, where t_f is the free-fall time; thus particles undergo approximately 5 vertical loops.



Dynamic Laplacian – eigenfunctions



- A large spectral gap between the 17th and 18th eigenvalues suggests using 17 eigenfunctions.
  [Klünker/Schneider/F/Schumacher/Padberg-Gehle'20].
- Remaining issue: the 17 coherent sets are "mixed" in the 17-dimensional eigenspace.

SEBA: Multiple feature extraction from data

Let $u_1, \ldots, u_r \in \mathbb{R}^p$ be some leading eigen-(or singular) vectors of a Markov (or Laplace) operator/matrix (usually $r \ll p$).

Main idea: Rotate the r-dimensional subspace $\mathrm{sp}\{u_1, \ldots, u_r\} \subset \mathbb{R}^p$ to create a sparse basis $\mathrm{sp}\{s_1, \ldots, s_r\} \subset \mathbb{R}^p$. Each vector in this sparse basis represents a single feature.

Create a tall, thin $p \times r$ data matrix $U = [u_1|u_2|\cdots|u_r]$.
Solve an optimisation problem for the sparse data array $S = [s_1|\cdots|s_r]$:

$$\underset{S \text{ is a data array}, R \text{ is orthogonal}}{\arg\min} \left(\|U - SR\|_F^2 + \mu \|S\|_{1,1} \right),$$

where $\|\cdot\|_F$ is the Frobenius norm and $\|\cdot\|_{1,1}$ is the matrix $\ell_{1,1}$ norm.

- An **exact local optimum** can be found by alternately applying **soft thresholding** and **compact SVD** (very fast as $r \ll p$).
- The resulting algorithm is a **Sparse EigenBasis Approximation (SEBA)** of the original data subspace. [F/Rock/Sakellariou'19].
- Improves and specialises [Journée *et al.*'10, Hu *et al.*'16].

SEBA: Multiple feature extraction from data

Let $u_1, \ldots, u_r \in \mathbb{R}^p$ be some leading eigen-(or singular) vectors of a Markov (or Laplace) operator/matrix (usually $r \ll p$).

Main idea: Rotate the r-dimensional subspace $\mathrm{sp}\{u_1, \ldots, u_r\} \subset \mathbb{R}^p$ to create a sparse basis $\mathrm{sp}\{s_1, \ldots, s_r\} \subset \mathbb{R}^p$. Each vector in this sparse basis represents a single feature.

Create a tall, thin $p \times r$ data matrix $U = [u_1|u_2|\cdots|u_r]$.
Solve an optimisation problem for the sparse data array $S = [s_1|\cdots|s_r]$:

$$\underset{S \text{ is a data array}, R \text{ is orthogonal}}{\arg\min} \left(\|U - SR\|_F^2 + \mu \|S\|_{1,1} \right),$$

where $\|\cdot\|_F$ is the Frobenius norm and $\|\cdot\|_{1,1}$ is the matrix $\ell_{1,1}$ norm.

- An **exact local optimum** can be found by alternately applying **soft thresholding** and **compact SVD** (very fast as $r \ll p$).
- The resulting algorithm is a **Sparse EigenBasis Approximation (SEBA)** of the original data subspace. [F/Rock/Sakellariou'19].
- Improves and specialises [Journée *et al.*'10, Hu *et al.*'16].

SEBA: Multiple feature extraction from data

Let $u_1, \ldots, u_r \in \mathbb{R}^p$ be some leading eigen-(or singular) vectors of a Markov (or Laplace) operator/matrix (usually $r \ll p$).

Main idea: Rotate the r-dimensional subspace $\mathrm{sp}\{u_1, \ldots, u_r\} \subset \mathbb{R}^p$ to create a sparse basis $\mathrm{sp}\{s_1, \ldots, s_r\} \subset \mathbb{R}^p$. Each vector in this sparse basis represents a single feature.

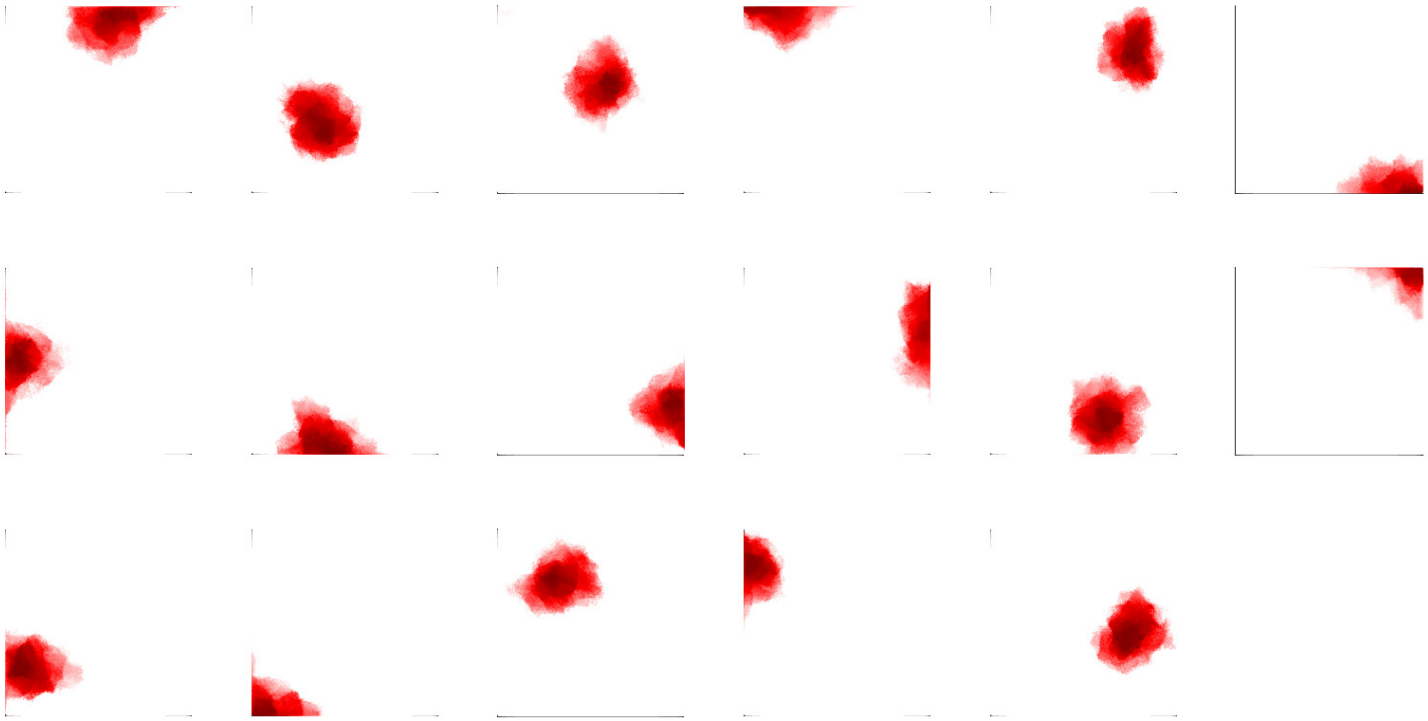
Create a tall, thin $p \times r$ data matrix $U = [u_1|u_2|\cdots|u_r]$.
Solve an optimisation problem for the sparse data array $S = [s_1|\cdots|s_r]$:

$$\underset{S \text{ is a data array}, R \text{ is orthogonal}}{\arg\min} \left(\|U - SR\|_F^2 + \mu \|S\|_{1,1} \right),$$

where $\|\cdot\|_F$ is the Frobenius norm and $\|\cdot\|_{1,1}$ is the matrix $\ell_{1,1}$ norm.

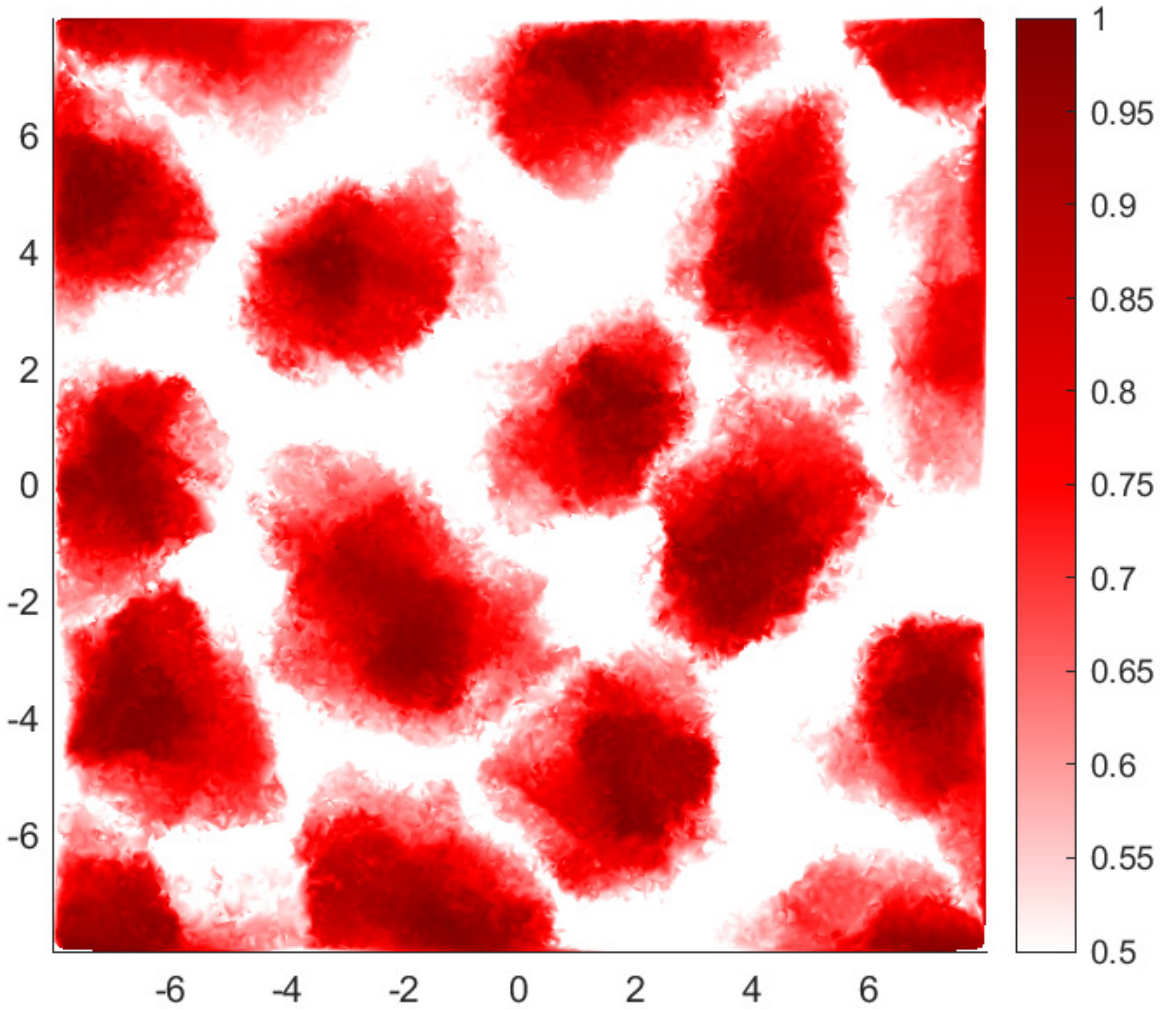
- An **exact local optimum** can be found by alternately applying **soft thresholding** and **compact SVD** (very fast as $r \ll p$).
- The resulting algorithm is a **Sparse EigenBasis Approximation (SEBA)** of the original data subspace. [F/Rock/Sakellariou'19].
- Improves and specialises [Journée *et al.*'10, Hu *et al.*'16].

Dynamic Laplacian – separating features with SEBA



- We apply Sparse EigenBasis Approximation (**SEBA**) to find a **sparse 17-dimensional basis** for the 17-dimensional eigenspace.
- **Each SEBA vector contains exactly one feature** – one coherent set.
- General SEBA code in Julia and Matlab is on **my webpage**, also `github.com/gfroyland/SEBA`.

Combining SEBA features into a single image



Plot of maximum value of the 17 SEBA vectors, interpretable as membership likelihood for each coherent set.

Nothing lasts forever

- *It is the nature of coherent sets to form, exist for some time, and then eventually decay and die.*
- For example, ocean eddies have lifetimes ranging from less than a month to over three years, with shorter lifetimes more frequent; superstructures in turbulent fluid eventually break down and reform.
- Apart from expensive trial and error on many different time windows, **how can one determine the timing of the birth and death of multiple coherent sets?**
- Attempts so far: brute force search of position and duration of window [Andrade-Canto/Karrasch/Beron-Vera'20, El Aouni'21]; fixed-length windows at different positions [Schneider/Viewig/Schumacher/Padberg-Gehle'22].
- Our approach is to use a **single wide time window** $[0, \tau]$, within which we expect coherent sets to be born, live, and die.

Nothing lasts forever

- *It is the nature of coherent sets to form, exist for some time, and then eventually decay and die.*
- For example, ocean eddies have lifetimes ranging from less than a month to over three years, with shorter lifetimes more frequent; superstructures in turbulent fluid eventually break down and reform.
- Apart from expensive trial and error on many different time windows, **how can one determine the timing of the birth and death of multiple coherent sets?**
- Attempts so far: brute force search of position and duration of window [Andrade-Canto/Karrasch/Beron-Vera'20, El Aouni'21]; fixed-length windows at different positions [Schneider/Viewig/Schumacher/Padberg-Gehle'22].
- Our approach is to use a **single wide time window** $[0, \tau]$, within which we expect coherent sets to be born, live, and die.

Nothing lasts forever

- *It is the nature of coherent sets to form, exist for some time, and then eventually decay and die.*
- For example, ocean eddies have lifetimes ranging from less than a month to over three years, with shorter lifetimes more frequent; superstructures in turbulent fluid eventually break down and reform.
- Apart from expensive trial and error on many different time windows, **how can one determine the timing of the birth and death of multiple coherent sets?**
- Attempts so far: brute force search of position and duration of window [Andrade-Canto/Karrasch/Beron-Vera'20, El Aouni'21]; fixed-length windows at different positions [Schneider/Viewig/Schumacher/Padberg-Gehle'22].
- Our approach is to use a **single wide time window** $[0, \tau]$, within which we expect coherent sets to be born, live, and die.

Nothing lasts forever

- *It is the nature of coherent sets to form, exist for some time, and then eventually decay and die.*
- For example, ocean eddies have lifetimes ranging from less than a month to over three years, with shorter lifetimes more frequent; superstructures in turbulent fluid eventually break down and reform.
- Apart from expensive trial and error on many different time windows, **how can one determine the timing of the birth and death of multiple coherent sets?**
- Attempts so far: brute force search of position and duration of window [Andrade-Canto/Karrasch/Beron-Vera'20, El Aouni'21]; fixed-length windows at different positions [Schneider/Viewig/Schumacher/Padberg-Gehle'22].
- Our approach is to use a **single wide time window** $[0, \tau]$, within which we expect coherent sets to be born, live, and die.

Nothing lasts forever

- *It is the nature of coherent sets to form, exist for some time, and then eventually decay and die.*
- For example, ocean eddies have lifetimes ranging from less than a month to over three years, with shorter lifetimes more frequent; superstructures in turbulent fluid eventually break down and reform.
- Apart from expensive trial and error on many different time windows, **how can one determine the timing of the birth and death of multiple coherent sets?**
- Attempts so far: brute force search of position and duration of window [Andrade-Canto/Karrasch/Beron-Vera'20, El Aouni'21]; fixed-length windows at different positions [Schneider/Viewig/Schumacher/Padberg-Gehle'22].
- Our approach is to use a **single wide time window** $[0, \tau]$, within which we expect coherent sets to be born, live, and die.

Time-expanding the domain

- To begin the process of relaxing materiality, we **time-expand the phase space**, giving each Riemannian manifold (M, g_t) its own t-fibre.
- Topologically, this time-expanded domain is simply

$$\mathbb{M}_0 := \bigcup_{t\in[0,\tau]} \{t\} \times M = [0,\tau] \times M.$$

In $\mathbb{M}_0$, a curve corresponding to the trajectory initialised at x_0: $\{\phi_t(x_0) : 0 \le t \le \tau\} \subset M$ is simply the "horizontal" line $\{(t, x_0) : 0 \le t \le \tau\} \subset \mathbb{M}_0$.

- We define the "co-evolved" spacetime manifold by

$$\mathbb{M}_1 := \bigcup_{t\in[0,\tau]} \{t\} \times \phi_t(M);$$

in $\mathbb{M}_1$ the trajectory of x_0 is $\{(t, \phi_t(x_0)) : 0 \le t \le \tau\} \subset \mathbb{M}_1$.

- The canonical mapping from $\mathbb{M}_0$ to the trajectory manifold $\mathbb{M}_1$, associating initial conditions with trajectories, is

$$\Phi : \mathbb{M}_0 \to \mathbb{M}_1, \quad (t, x) \mapsto (t, \phi_t(x)).$$

Time-expanding the domain

- To begin the process of relaxing materiality, we **time-expand the phase space**, giving each Riemannian manifold (M, g_t) its own t-fibre.
- Topologically, this time-expanded domain is simply

$$\mathbb{M}_0 := \bigcup_{t\in[0,\tau]} \{t\} \times M = [0,\tau] \times M.$$

In $\mathbb{M}_0$, a curve corresponding to the trajectory initialised at x_0: $\{\phi_t(x_0) : 0 \le t \le \tau\} \subset M$ is simply the "horizontal" line $\{(t, x_0) : 0 \le t \le \tau\} \subset \mathbb{M}_0$.

- We define the "co-evolved" spacetime manifold by

$$\mathbb{M}_1 := \bigcup_{t\in[0,\tau]} \{t\} \times \phi_t(M);$$

in $\mathbb{M}_1$ the trajectory of x_0 is $\{(t, \phi_t(x_0)) : 0 \le t \le \tau\} \subset \mathbb{M}_1$.

- The canonical mapping from $\mathbb{M}_0$ to the trajectory manifold $\mathbb{M}_1$, associating initial conditions with trajectories, is

$$\Phi : \mathbb{M}_0 \to \mathbb{M}_1, \quad (t, x) \mapsto (t, \phi_t(x)).$$

Time-expanding the domain

- To begin the process of relaxing materiality, we **time-expand the phase space**, giving each Riemannian manifold (M, g_t) its own t-fibre.
- Topologically, this time-expanded domain is simply

$$\mathbb{M}_0 := \bigcup_{t\in[0,\tau]} \{t\} \times M = [0,\tau] \times M.$$

In $\mathbb{M}_0$, a curve corresponding to the trajectory initialised at x_0: $\{\phi_t(x_0) : 0 \le t \le \tau\} \subset M$ is simply the "horizontal" line $\{(t, x_0) : 0 \le t \le \tau\} \subset \mathbb{M}_0$.

- We define the "co-evolved" spacetime manifold by

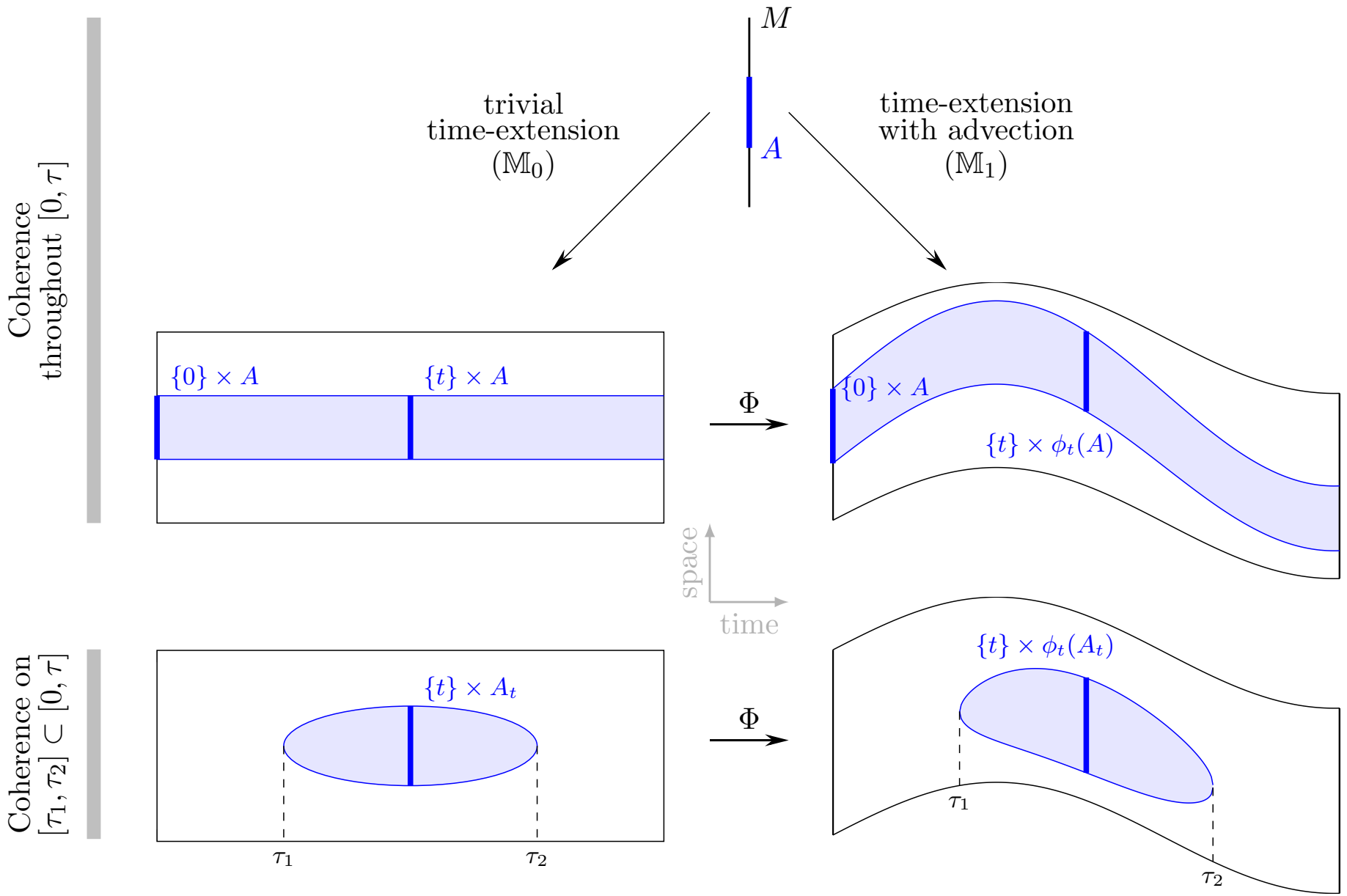
$$\mathbb{M}_1 := \bigcup_{t\in[0,\tau]} \{t\} \times \phi_t(M);$$

in $\mathbb{M}_1$ the trajectory of x_0 is $\{(t, \phi_t(x_0)) : 0 \le t \le \tau\} \subset \mathbb{M}_1$.

- The canonical mapping from $\mathbb{M}_0$ to the trajectory manifold $\mathbb{M}_1$, associating initial conditions with trajectories, is

$$\Phi : \mathbb{M}_0 \to \mathbb{M}_1, \quad (t, x) \mapsto (t, \phi_t(x)).$$

The time-expanded constructions on $\mathbb{M}_0$ and $\mathbb{M}_1$



Time expansion and the inflated dynamic Laplace operator

- Recall that Δ^D is an operator on M.
- We now time expand M and create an operator on $\mathbb{M}_0 := [0, \tau] \times M$.
- The obvious thing to do is to apply Δ_{g_t} to the t^{th} copy of M, i.e. to $\{t\} \times M \subset \mathbb{M}_0$, for each $t = [0, \tau]$.
- Finally, we apply diffusion in the temporal direction to **dynamically connect the distinct t-fibres**; this diffusion will control how much we relax materiality.

In summary, we define a Laplace–Beltrami operator $\Delta_{G_a} : L^2(\mathbb{M}_0, G_a) \to L^2(\mathbb{M}_0, G_a)$, where the metric G_a is Euclidean in the time direction and g_t on the t-fibre:

$$\Delta_{G_a} F(t, \cdot) = a^2 \partial_{tt} F(t, \cdot) + \Delta_{g_t} F(t, \cdot).$$

which we call the **inflated dynamic Laplace operator**.

Time expansion and the inflated dynamic Laplace operator

- Recall that Δ^D is an operator on M.
- We now time expand M and create an operator on $\mathbb{M}_0 := [0, \tau] \times M$.
- The obvious thing to do is to apply Δ_{g_t} to the t^{th} copy of M, i.e. to $\{t\} \times M \subset \mathbb{M}_0$, for each $t = [0, \tau]$.
- Finally, we apply diffusion in the temporal direction to **dynamically connect the distinct t-fibres**; this diffusion will control how much we relax materiality.

In summary, we define a Laplace–Beltrami operator $\Delta_{G_a} : L^2(\mathbb{M}_0, G_a) \to L^2(\mathbb{M}_0, G_a)$, where the metric G_a is Euclidean in the time direction and g_t on the t-fibre:

$$\Delta_{G_a} F(t, \cdot) = a^2 \partial_{tt} F(t, \cdot) + \Delta_{g_t} F(t, \cdot).$$

which we call the **inflated dynamic Laplace operator**.

Time expansion and the inflated dynamic Laplace operator

- Recall that Δ^D is an operator on M.
- We now time expand M and create an operator on $\mathbb{M}_0 := [0, \tau] \times M$.
- The obvious thing to do is to apply Δ_{g_t} to the t^{th} copy of M, i.e. to $\{t\} \times M \subset \mathbb{M}_0$, for each $t = [0, \tau]$.
- Finally, we apply diffusion in the temporal direction to **dynamically connect the distinct t-fibres**; this diffusion will control how much we relax materiality.

In summary, we define a Laplace–Beltrami operator $\Delta_{G_a} : L^2(\mathbb{M}_0, G_a) \to L^2(\mathbb{M}_0, G_a)$, where the metric G_a is Euclidean in the time direction and g_t on the t-fibre:

$$\Delta_{G_a} F(t, \cdot) = a^2 \partial_{tt} F(t, \cdot) + \Delta_{g_t} F(t, \cdot).$$

which we call the **inflated dynamic Laplace operator**.

The spectrum of the inflated dynamic Laplacian

Let the spectrum of the dynamic Laplacian Δ^D be $0 \geq \lambda_1^D > \lambda_2^D > \cdots$ and the spectrum of the *inflated* dynamic Laplacian Δ_{G_a} be $0 \geq \Lambda_{1,a} > \Lambda_{2,a} > \cdots$. Let's compare:

Theorem [F/Koltai, *Comm. Pure Appl. Math.'22*]

1. For each $k \geq 1$ and $a > 0$ one has $\lambda_k^D \leq \Lambda_{k,a}$.
   (For **any** $a > 0$ *the dynamic Laplacian eigenfunctions always experience* **more mixing** *than the eigenfunctions of the inflated dynamic Laplacian).*
2. For each $k \geq 1$, $\Lambda_{k,a}$ are nonincreasing in $a \geq 0$.
   *(***Increasing** *the parameter a leads to* **more mixing***...because greater temporal connectivity inhibits spatial adaptation of an eigenfunction).*
3. For each $k \geq 1$, $\lim_{a\to\infty} \Lambda_{k,a} = \lambda_k^D$.
   *(***Convergence** *of inflated dynamic Laplacian eigenvalues to the eigenvalues of the dynamic Laplacian as $a \to \infty$).* **Thus, a is a tuning parameter** interpolating between **completely non-material** $(a = 0)$ **and fully material** $(a = \infty)$ **coherent sets.**

The spectrum of the inflated dynamic Laplacian

Let the spectrum of the dynamic Laplacian Δ^D be $0 \geq \lambda_1^D > \lambda_2^D > \cdots$ and the spectrum of the *inflated* dynamic Laplacian Δ_{G_a} be $0 \geq \Lambda_{1,a} > \Lambda_{2,a} > \cdots$. Let's compare:

Theorem [F/Koltai, *Comm. Pure Appl. Math.'22*]

1. For each $k \geq 1$ and $a > 0$ one has $\lambda_k^D \leq \Lambda_{k,a}$.
(For **any** *$a > 0$ the dynamic Laplacian eigenfunctions always experience* **more mixing** *than the eigenfunctions of the inflated dynamic Laplacian).*
2. For each $k \geq 1$, $\Lambda_{k,a}$ are nonincreasing in $a \geq 0$.
*(***Increasing** *the parameter a leads to* **more mixing***...because greater temporal connectivity inhibits spatial adaptation of an eigenfunction).*
3. For each $k \geq 1$, $\lim_{a\to\infty} \Lambda_{k,a} = \lambda_k^D$.
*(***Convergence** *of inflated dynamic Laplacian eigenvalues to the eigenvalues of the dynamic Laplacian as $a \to \infty$).* **Thus, a is a tuning parameter** interpolating between **completely non-material ($a = 0$) and fully material ($a = \infty$) coherent sets.**

The spectrum of the inflated dynamic Laplacian

Let the spectrum of the dynamic Laplacian Δ^D be $0 \geq \lambda_1^D > \lambda_2^D > \cdots$ and the spectrum of the *inflated* dynamic Laplacian Δ_{G_a} be $0 \geq \Lambda_{1,a} > \Lambda_{2,a} > \cdots$. Let's compare:

Theorem [F/Koltai, *Comm. Pure Appl. Math.'22*]

1. For each $k \geq 1$ and $a > 0$ one has $\lambda_k^D \leq \Lambda_{k,a}$.
   (For **any** $a > 0$ *the dynamic Laplacian eigenfunctions always experience* **more mixing** *than the eigenfunctions of the inflated dynamic Laplacian).*
2. For each $k \geq 1$, $\Lambda_{k,a}$ are nonincreasing in $a \geq 0$.
   *(***Increasing** *the parameter a leads to* **more mixing***...because greater temporal connectivity inhibits spatial adaptation of an eigenfunction).*
3. For each $k \geq 1$, $\lim_{a\to\infty} \Lambda_{k,a} = \lambda_k^D$.
   *(***Convergence** *of inflated dynamic Laplacian eigenvalues to the eigenvalues of the dynamic Laplacian as $a \to \infty$).* **Thus, a is a tuning parameter** interpolating between **completely non-material ($a = 0$) and fully material ($a = \infty$) coherent sets.**

The spectrum of the inflated dynamic Laplacian

Let the spectrum of the dynamic Laplacian Δ^D be $0 \geq \lambda_1^D > \lambda_2^D > \cdots$ and the spectrum of the *inflated* dynamic Laplacian Δ_{G_a} be $0 \geq \Lambda_{1,a} > \Lambda_{2,a} > \cdots$. Let's compare:

Theorem [F/Koltai, *Comm. Pure Appl. Math.'22*]

1. For each $k \geq 1$ and $a > 0$ one has $\lambda_k^D \leq \Lambda_{k,a}$.
   (For **any** $a > 0$ *the dynamic Laplacian eigenfunctions always experience* **more mixing** *than the eigenfunctions of the inflated dynamic Laplacian).*
2. For each $k \geq 1$, $\Lambda_{k,a}$ are nonincreasing in $a \geq 0$.
   *(***Increasing** *the parameter a leads to* **more mixing***...because greater temporal connectivity inhibits spatial adaptation of an eigenfunction).*
3. For each $k \geq 1$, $\lim_{a\to\infty} \Lambda_{k,a} = \lambda_k^D$.
   *(***Convergence** *of inflated dynamic Laplacian eigenvalues to the eigenvalues of the dynamic Laplacian as $a \to \infty$).* **Thus, a is a tuning parameter** interpolating between **completely non-material ($a = 0$) and fully material ($a = \infty$) coherent sets.**

An example

- In the flow data below, given by 400 trajectories, sampled at 100 time instants, coherence is present for 60% **of the flow time**.
- From the eigenfunctions of the inflated dynamic Laplacian, we will **automatically find when these coherent sets exist** and **where they are in phase space**.

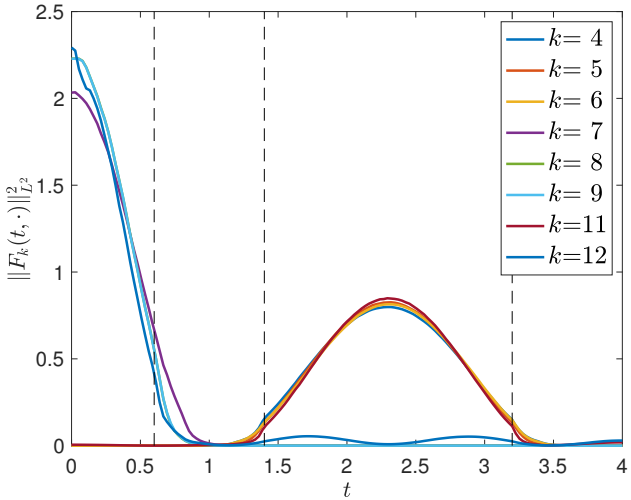
An example

- In the flow data below, given by 400 trajectories, sampled at 100 time instants, coherence is present for 60% **of the flow time**.
- From the eigenfunctions of the inflated dynamic Laplacian, we will **automatically find when these coherent sets exist** and **where they are in phase space**.

Slices of eigenvectors on $\mathbb{M}_0$

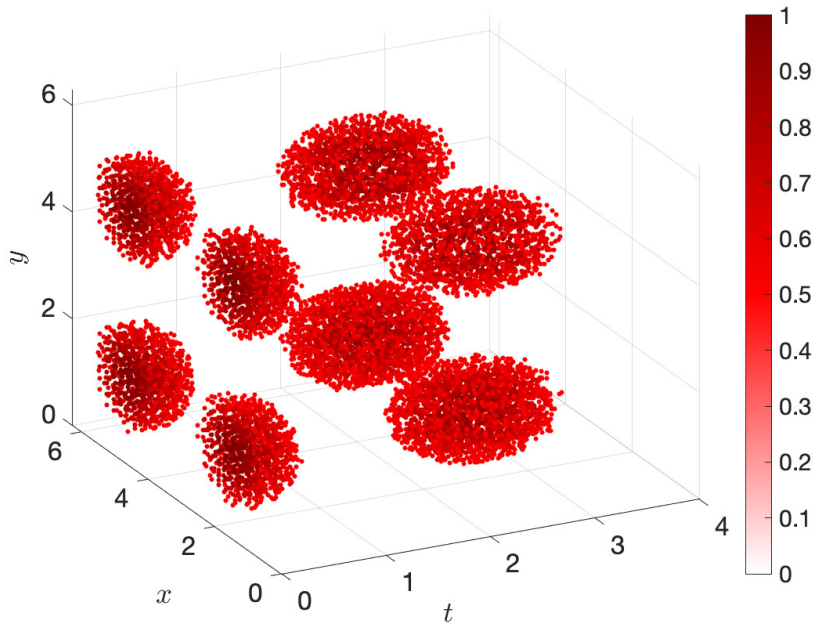
(a) First 4 spatial eigenfunctions, shown on timeslices of maximal norm.

(b) Next 4 spatial eigenfunctions, shown on timeslices of maximal norm.



(c) Left: Eigenvector timeslice norms.

Trajectory points in co-evolved space $\mathbb{M}_1$ coloured by SEBA output



- We use SEBA to automatically find a sparse basis for the leading 8 spatial eigenfunctions.
- The **8 separately supported vectors** (each vector is supported on one coherent set in space-time) **are superimposed in this figure**.
- Note that SEBA knows nothing about "space" and "time" coordinates in our dynamical system.

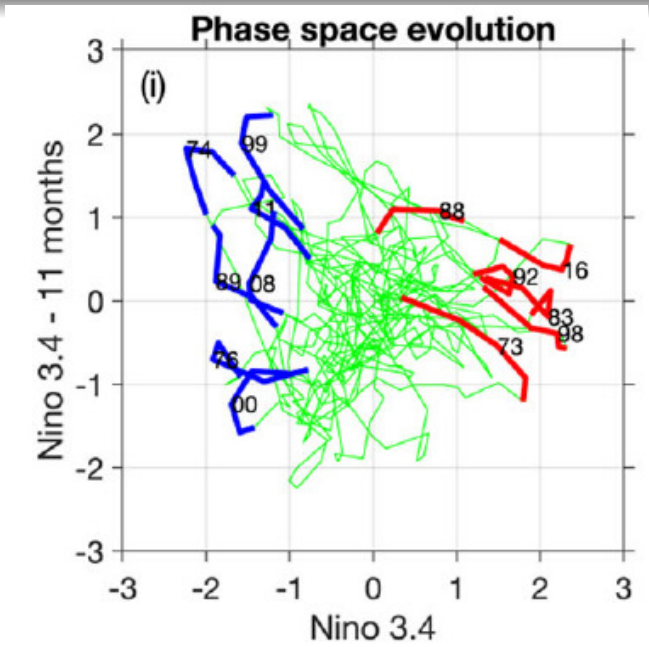
Left: composite SEBA function timeslices through time
Right: trajectories coloured by SEBA through time.

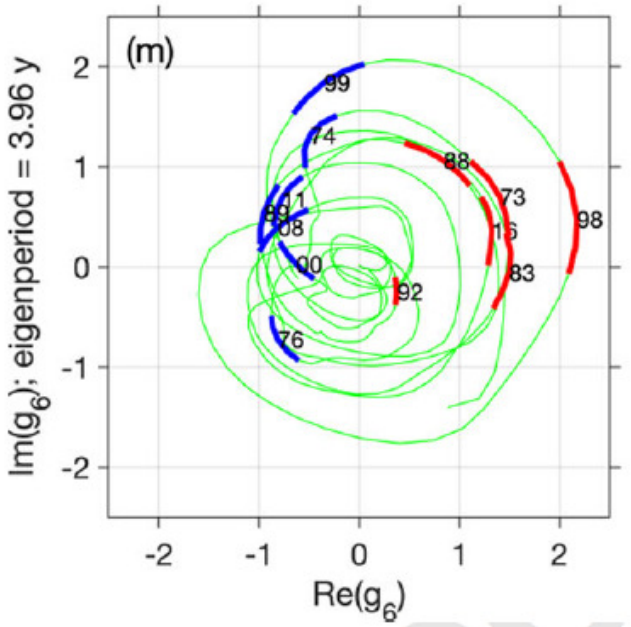
Summary – Part II

- The eigenfunctions of the **dynamic Laplacian** are an effective way to extract finite-time coherent sets from sparse trajectory data.
- An efficient FEM-based approach [F/Junge'18] handles sparse trajectory data very robustly (code at `github.com/gaioguy/FEMDL`).
- When many coherent sets are present, the **SEBA algorithm** [F/Rock/Sakellariou'19] disentangles the eigenfunctions arising from the dynamic Laplacian.
- SEBA is a **general-purpose post-processing step** for the output of any type of **spectral clustering**.
- When coherent sets are emerging and vanishing, the eigenfunctions of the **inflated dynamic Laplacian** [F/Koltai'22] (to appear in *Comm.Pure.Appl.Math*), clearly detect the birth and death phenomena as well as spatial evolution during the coherent set lifetimes.
- Currently extending and specialising these algorithms for application to (i) geophysical flows, (ii) general high-dimensional datasets, and (iii) time-varying networks.

Plotting the complex eigenvector

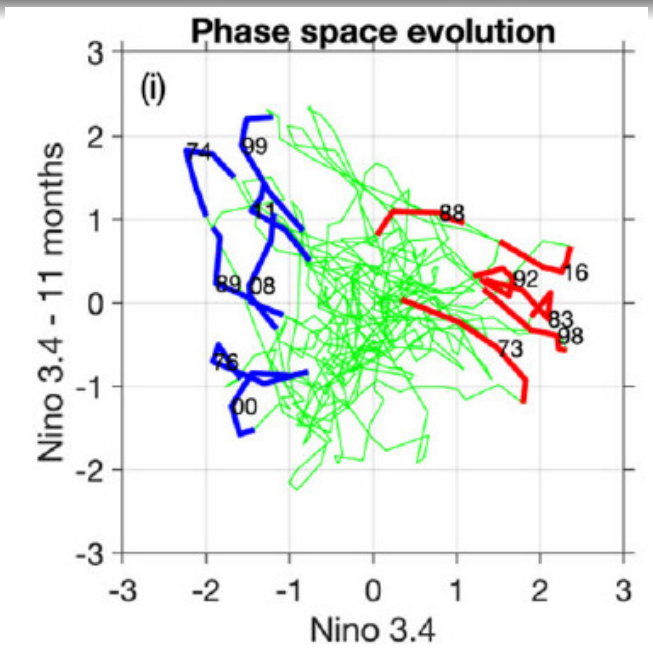
- *Upper:* a lagged Niño 3.4 index (lagged by 1 year = 1/4 period). Note that by definition **strong El Niño is the far right** and **strong La Niña is the far left**. *Lower:* plot of eigenfunction $g(\tilde{y}_t)$.
- Those parts of the domain corresponding to **large magnitude values of the eigenfunctions** correspond to a **strong (slowly decaying) ENSO cycle**.
- The transition from **El Niño to La Niña is faster** than from La Niña to El Niño [An/Kim'18].
- Our eigenfunction **"rectifies" the ENSO cycle** so that in the lower figure, one **proceeds at constant speed** and La Niña will appear "earlier" in the northwest of the plot.

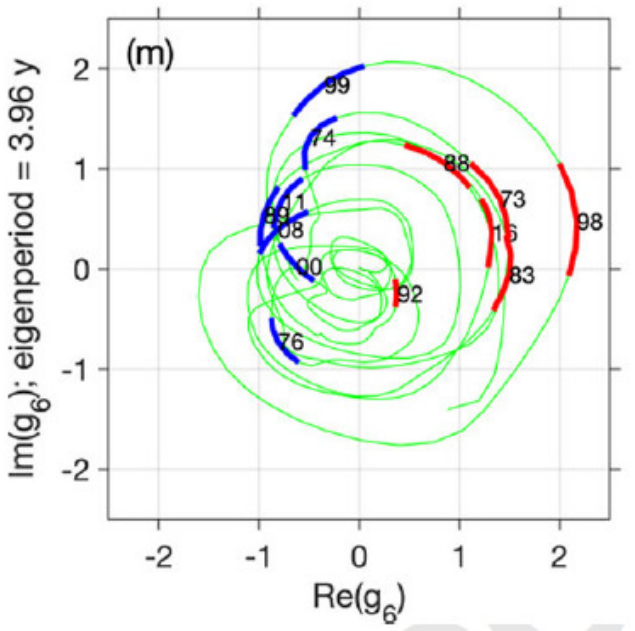




Plotting the complex eigenvector

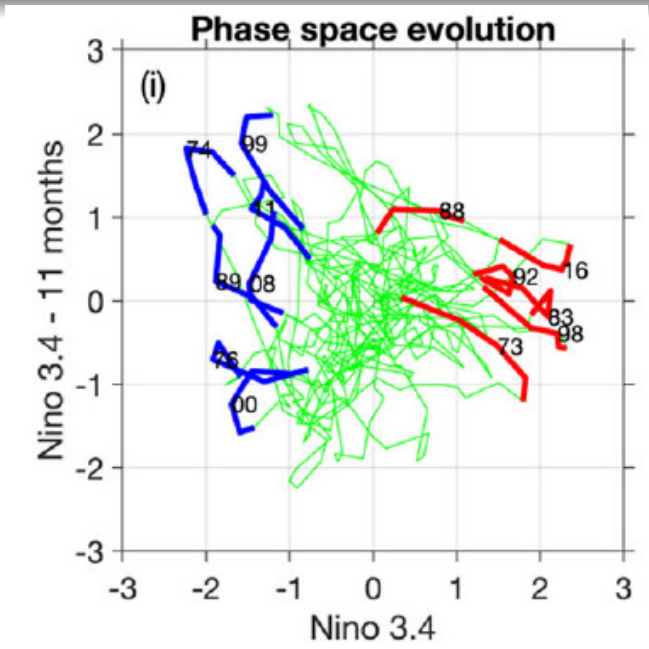
- *Upper:* a lagged Niño 3.4 index (lagged by 1 year = 1/4 period). Note that by definition **strong El Niño is the far right** and **strong La Niña is the far left**. *Lower:* plot of eigenfunction $g(\tilde{y}_t)$.
- Those parts of the domain corresponding to **large magnitude values of the eigenfunctions** correspond to a **strong (slowly decaying) ENSO cycle**.
- The transition from **El Niño to La Niña is faster** than from La Niña to El Niño [An/Kim'18].
- Our eigenfunction **"rectifies" the ENSO cycle** so that in the lower figure, one **proceeds at constant speed** and La Niña will appear "earlier" in the northwest of the plot.

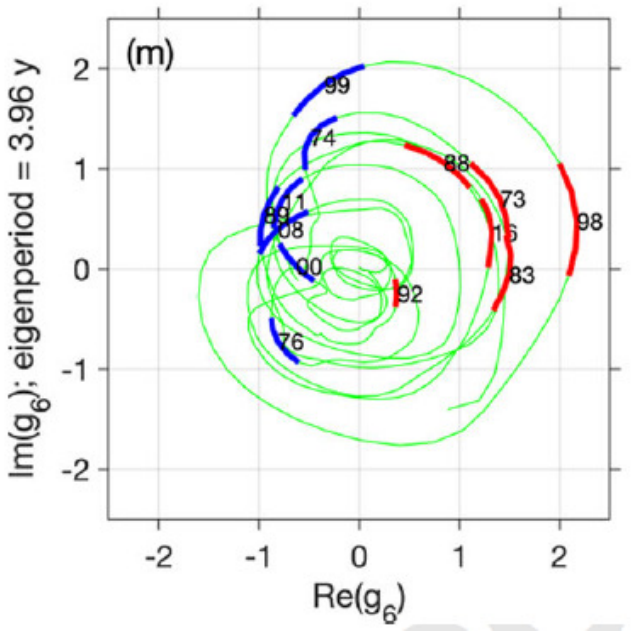




Plotting the complex eigenvector

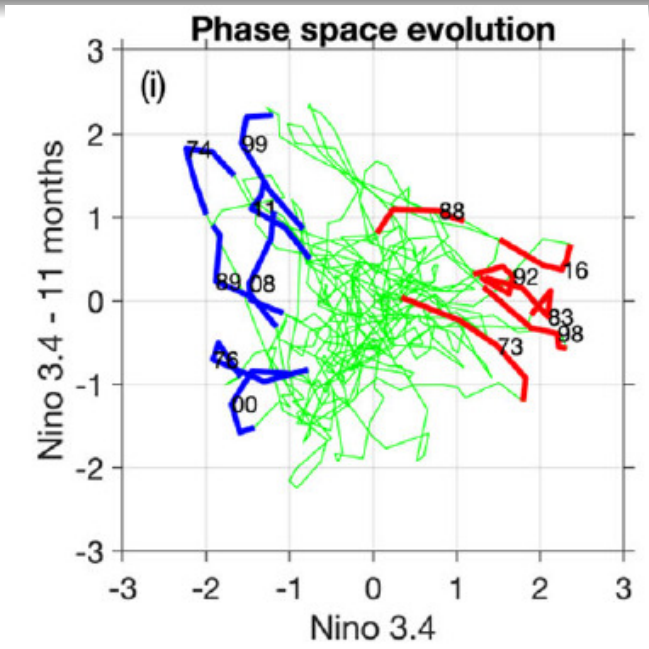
- *Upper:* a lagged Niño 3.4 index (lagged by 1 year = 1/4 period). Note that by definition **strong El Niño is the far right** and **strong La Niña is the far left**. *Lower:* plot of eigenfunction $g(\tilde{y}_t)$.
- Those parts of the domain corresponding to **large magnitude values of the eigenfunctions** correspond to a **strong (slowly decaying) ENSO cycle**.
- The transition from **El Niño to La Niña is faster** than from La Niña to El Niño [An/Kim'18].
- Our eigenfunction **"rectifies" the ENSO cycle** so that in the lower figure, one **proceeds at constant speed** and La Niña will appear "earlier" in the northwest of the plot.

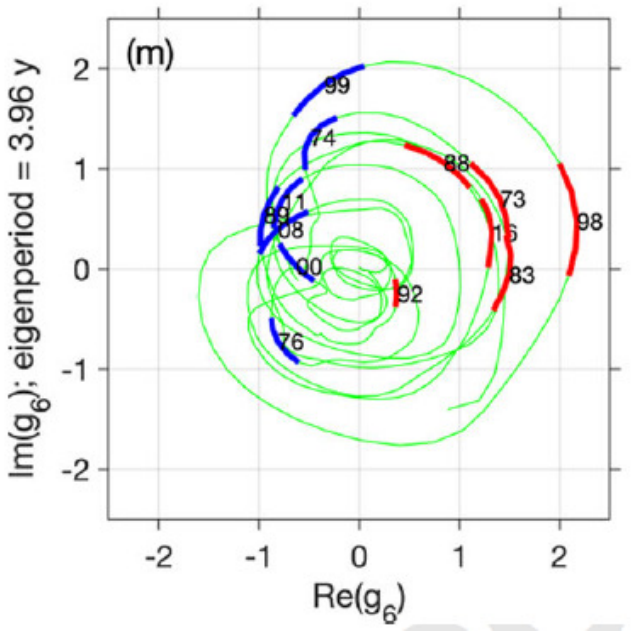




Plotting the complex eigenvector

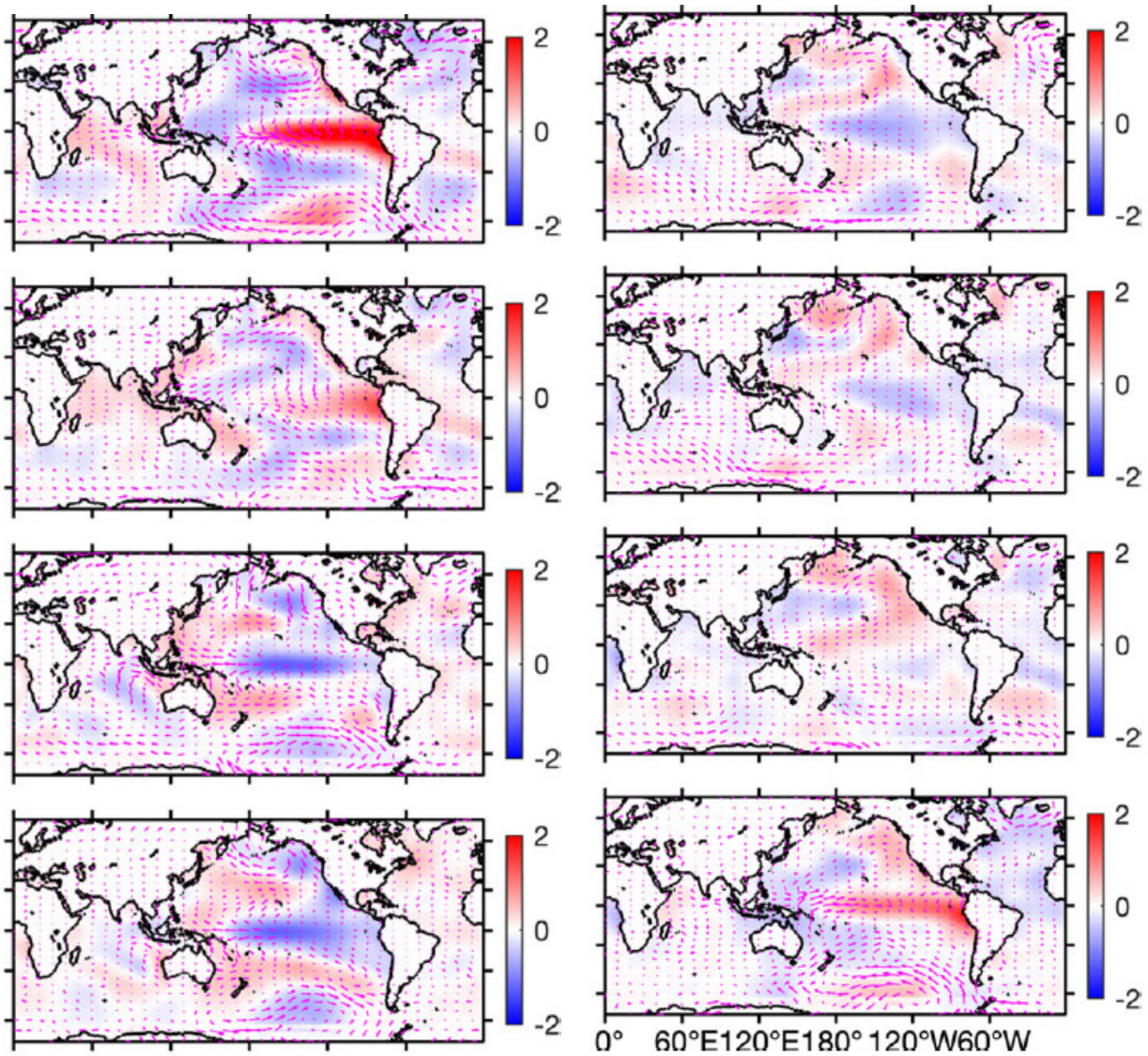
- *Upper:* a lagged Niño 3.4 index (lagged by 1 year = 1/4 period). Note that by definition **strong El Niño is the far right** and **strong La Niña is the far left**. *Lower:* plot of eigenfunction $g(\tilde{y}_t)$.
- Those parts of the domain corresponding to **large magnitude values of the eigenfunctions** correspond to a **strong (slowly decaying) ENSO cycle**.
- The transition from **El Niño to La Niña is faster** than from La Niña to El Niño [An/Kim'18].
- Our eigenfunction **"rectifies" the ENSO cycle** so that in the lower figure, one **proceeds at constant speed** and La Niña will appear "earlier" in the northwest of the plot.



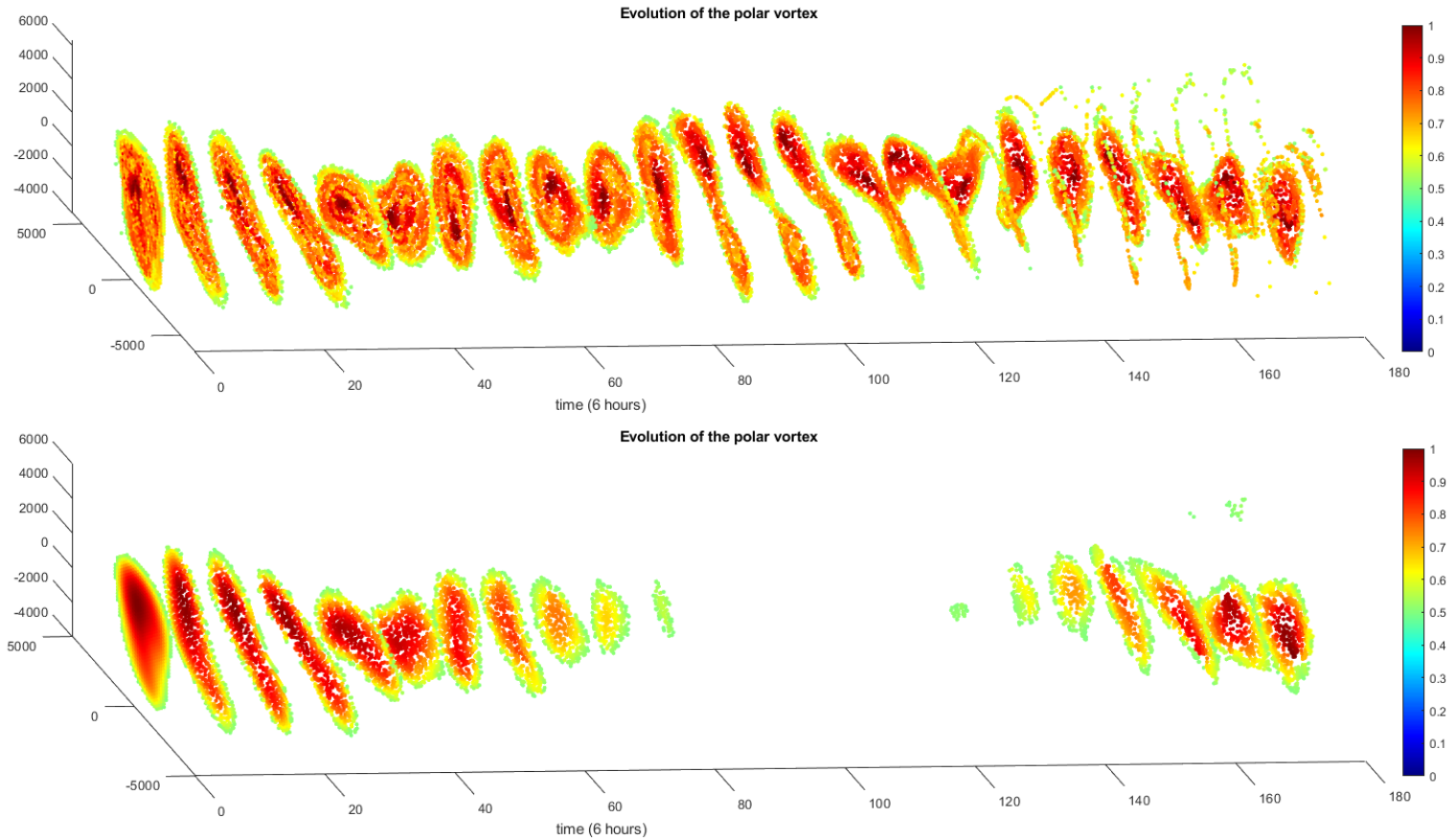


Phases as SST fields

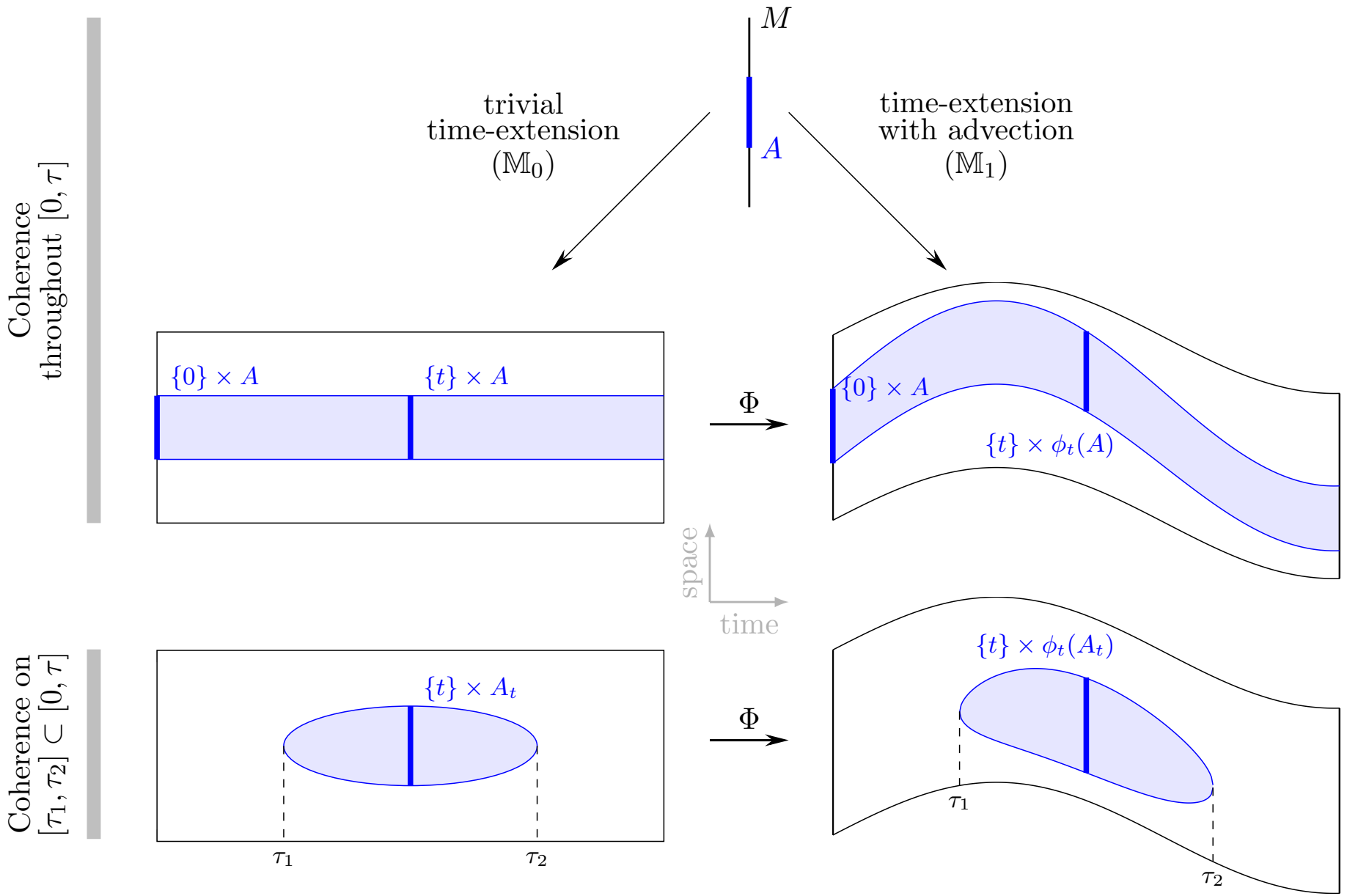
By computing the **average SST anomaly in each of our 8 phases**, we obtain 8 SST anomaly fields equally spaced around our **canonical ENSO cycle**.



Polar vortex breakup



The time-expanded constructions on $\mathbb{M}_0$ and $\mathbb{M}_1$

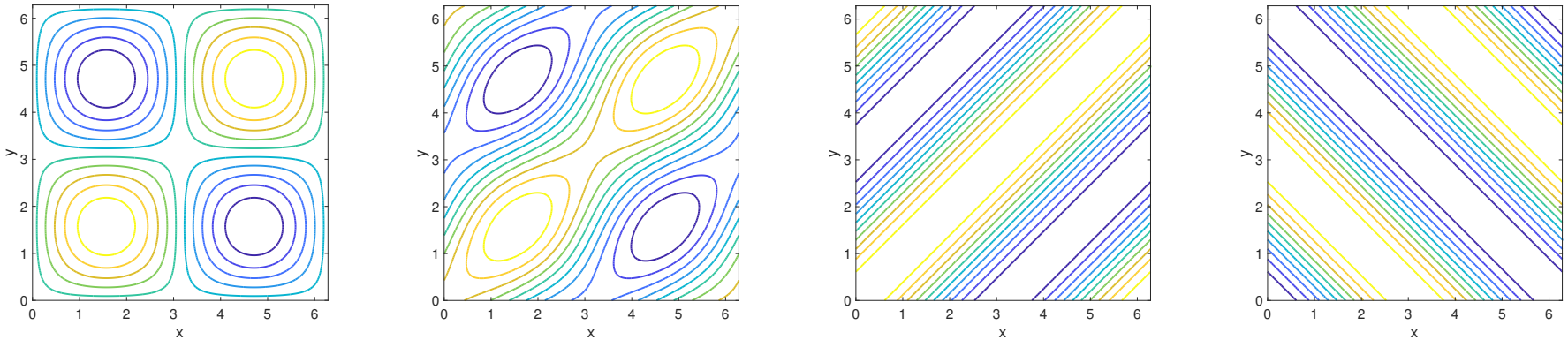


An example

We consider the Childress-Soward flow, a two-dimensional velocity field [Childress/Soward'89] $v : \mathbb{T}^2 \to \mathbb{R}^2$, parameterised by $A > 0$ and $-1 \leq r \leq 1$:

$$v = A \cdot \left(\frac{\partial \psi_r}{\partial y}, -\frac{\partial \psi_r}{\partial x} \right),$$

with streamfunction $\psi_r(x, y) = \sin x \sin y + r \cos x \cos y$.
For $r \approx 0$ the flow has **four vortices**, for $|r| \approx 1$ the flow is a **diagonal shear**; otherwise it possesses an intermediate "cat's-eye" structure.



Computation

- In low dimensions we estimate Δ^D by **meshing the location of trajectories at each time instant** t in a discrete subset of $[0, \tau]$ and applying a bespoke finite-element approach [F/Junge'18].
- The eigenproblem $\Delta^D f = \frac{1}{\tau}\int_0^\tau \Delta_{g_t}\, dt = \lambda f$ becomes a discrete generalised eigenproblem $\frac{1}{T}\sum_{t=1}^T D_t = \lambda \frac{1}{T}\sum_{t=1}^T M_t$.
- (The entries of D_t and M_t can be associated with **edge weights** and **vertex weights** of the graph defined by the mesh).
- **Code** for the meshing and construction of the D_t and M_t is available at `github.com/gaioguy/femdl`.
- To estimate the *inflated dynamic Laplacian on time-expanded space*, we take the matrices D_t and M_t, $t = 1, \ldots, T$ and "**linearly interpolate**" **across the time dimension** [F/Koltai'22].
- Alternatively (and relevant for high-dimensional data) one can use a specialised diffusion-maps approach to compute eigenfunctions of the inflated dynamic Laplacian [Atnip/F/Koltai, in prep.].

Computation

- In low dimensions we estimate Δ^D by **meshing the location of trajectories at each time instant** t in a discrete subset of $[0, \tau]$ and applying a bespoke finite-element approach [F/Junge'18].
- The eigenproblem $\Delta^D f = \frac{1}{\tau} \int_0^\tau \Delta_{g_t} \, dt = \lambda f$ becomes a discrete generalised eigenproblem $\frac{1}{T} \sum_{t=1}^{T} D_t = \lambda \frac{1}{T} \sum_{t=1}^{T} M_t$.
- (The entries of D_t and M_t can be associated with **edge weights** and **vertex weights** of the graph defined by the mesh).
- **Code** for the meshing and construction of the D_t and M_t is available at `github.com/gaioguy/femdl`.
- To estimate the *inflated dynamic Laplacian on time-expanded space*, we take the matrices D_t and M_t, $t = 1, \ldots, T$ and **"linearly interpolate" across the time dimension** [F/Koltai'22].
- Alternatively (and relevant for high-dimensional data) one can use a specialised diffusion-maps approach to compute eigenfunctions of the inflated dynamic Laplacian [Atnip/F/Koltai, in prep.].

Computation

- In low dimensions we estimate Δ^D by **meshing the location of trajectories at each time instant** t in a discrete subset of $[0, \tau]$ and applying a bespoke finite-element approach [F/Junge'18].
- The eigenproblem $\Delta^D f = \frac{1}{\tau} \int_0^\tau \Delta_{g_t} \, dt = \lambda f$ becomes a discrete generalised eigenproblem $\frac{1}{T} \sum_{t=1}^{T} D_t = \lambda \frac{1}{T} \sum_{t=1}^{T} M_t$.
- (The entries of D_t and M_t can be associated with **edge weights** and **vertex weights** of the graph defined by the mesh).
- **Code** for the meshing and construction of the D_t and M_t is available at `github.com/gaioguy/femdl`.
- To estimate the *inflated dynamic Laplacian on time-expanded space*, we take the matrices D_t and M_t, $t = 1, \ldots, T$ and **"linearly interpolate" across the time dimension** [F/Koltai'22].
- Alternatively (and relevant for high-dimensional data) one can use a specialised diffusion-maps approach to compute eigenfunctions of the inflated dynamic Laplacian [Atnip/F/Koltai, in prep.].

Computation

- In low dimensions we estimate Δ^D by **meshing the location of trajectories at each time instant** t in a discrete subset of $[0, \tau]$ and applying a bespoke finite-element approach [F/Junge'18].
- The eigenproblem $\Delta^D f = \frac{1}{\tau} \int_0^\tau \Delta_{g_t} \, dt = \lambda f$ becomes a discrete generalised eigenproblem $\frac{1}{T} \sum_{t=1}^{T} D_t = \lambda \frac{1}{T} \sum_{t=1}^{T} M_t$.
- (The entries of D_t and M_t can be associated with **edge weights** and **vertex weights** of the graph defined by the mesh).
- **Code** for the meshing and construction of the D_t and M_t is available at `github.com/gaioguy/femdl`.
- To estimate the *inflated dynamic Laplacian on time-expanded space*, we take the matrices D_t and M_t, $t = 1, \ldots, T$ and **"linearly interpolate" across the time dimension** [F/Koltai'22].
- Alternatively (and relevant for high-dimensional data) one can use a specialised diffusion-maps approach to compute eigenfunctions of the inflated dynamic Laplacian [Atnip/F/Koltai, in prep.].

Computation

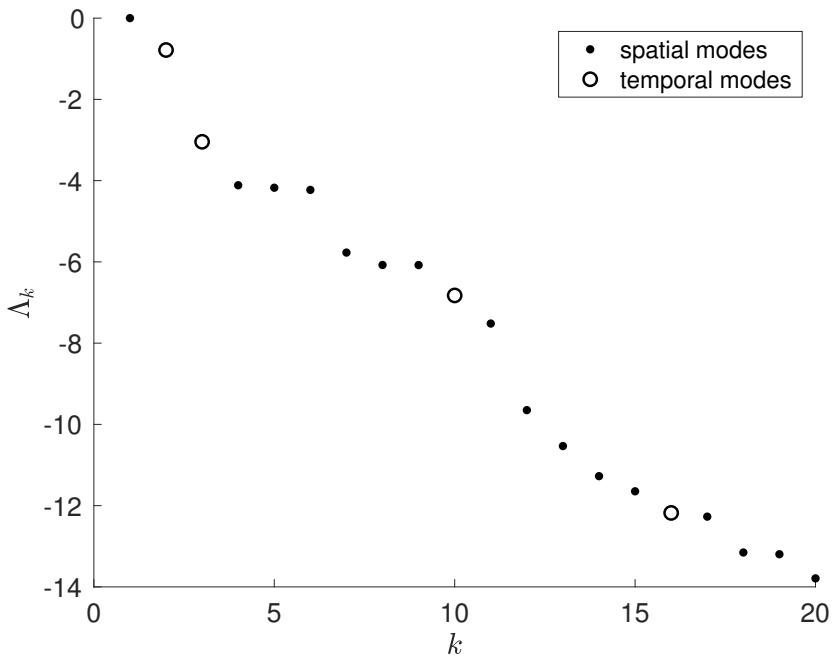
- In low dimensions we estimate Δ^D by **meshing the location of trajectories at each time instant** t in a discrete subset of $[0, \tau]$ and applying a bespoke finite-element approach [F/Junge'18].
- The eigenproblem $\Delta^D f = \frac{1}{\tau} \int_0^\tau \Delta_{g_t} \, dt = \lambda f$ becomes a discrete generalised eigenproblem $\frac{1}{T} \sum_{t=1}^T D_t = \lambda \frac{1}{T} \sum_{t=1}^T M_t$.
- (The entries of D_t and M_t can be associated with **edge weights** and **vertex weights** of the graph defined by the mesh).
- **Code** for the meshing and construction of the D_t and M_t is available at `github.com/gaioguy/femdl`.
- To estimate the *inflated dynamic Laplacian on time-expanded space*, we take the matrices D_t and M_t, $t = 1, \ldots, T$ and **"linearly interpolate" across the time dimension** [F/Koltai'22].
- Alternatively (and relevant for high-dimensional data) one can use a specialised diffusion-maps approach to compute eigenfunctions of the inflated dynamic Laplacian [Atnip/F/Koltai, in prep.].

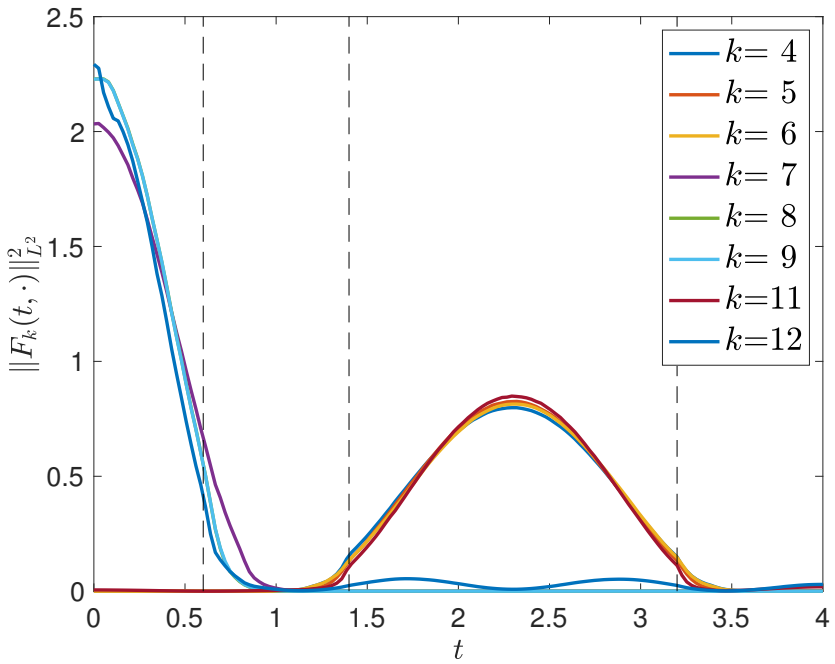
Computation

- In low dimensions we estimate Δ^D by **meshing the location of trajectories at each time instant** t in a discrete subset of $[0, \tau]$ and applying a bespoke finite-element approach [F/Junge'18].
- The eigenproblem $\Delta^D f = \frac{1}{\tau} \int_0^\tau \Delta_{g_t} \, dt = \lambda f$ becomes a discrete generalised eigenproblem $\frac{1}{T} \sum_{t=1}^T D_t = \lambda \frac{1}{T} \sum_{t=1}^T M_t$.
- (The entries of D_t and M_t can be associated with **edge weights** and **vertex weights** of the graph defined by the mesh).
- **Code** for the meshing and construction of the D_t and M_t is available at `github.com/gaioguy/femdl`.
- To estimate the *inflated dynamic Laplacian on time-expanded space*, we take the matrices D_t and M_t, $t = 1, \ldots, T$ and **"linearly interpolate" across the time dimension** [F/Koltai'22].
- Alternatively (and relevant for high-dimensional data) one can use a specialised diffusion-maps approach to compute eigenfunctions of the inflated dynamic Laplacian [Atnip/F/Koltai, in prep.].

Spectrum and eigenfunctions of Δ_{G_a}

- We use 400 trajectories and 100 time steps over the interval $[0, 4]$ (videos) and 900 trajectories and 150 time steps (still images).
- The plot on the left shows the numerical spectrum of the inflated dynamic Laplacian (separated into temporal and spatial eigenvalues).
- Together with the spatial eigenvalue 0 we expect **8 dominant coherent sets** that **come and go** during times $[0, 4]$.
- The plot on the right shows the norm of the t-fibre, indicating **when** the coherence is present.





Other contributions in [F/Koltai,subm]

In addition to theory and numerics for **automatically detecting partially present coherent sets** and **timing of the loss of coherence:**

1. In [F/Koltai'22] we introduce *inflated dynamic Cheeger and Sobolev constants* (which are the standard constants associated with Δ_{G_a}) and prove a theorem relating these to dynamic Cheeger and Sobolev constants [F'15]. These geometric results **formalises trade-off of the regularity of the boundaries of semi-material coherent sets:** *more material means more irregular coherent sets.*
2. We collapse the spatial dimensions to produce a one-dimensional Sturm–Liouville ODE (in the time-coordinate) that provides
   - **finer information on spatial norms of eigenfunctions over time**.
   - a natural **quantification of instantaneous mixing experienced** by an eigenfunction and thus **identification of time intervals where mixing is greater than or less than average**.
3. As $a \to \infty$ we recover the process generated by Δ^D (time-copied on $\mathbb{M}_0$) from the SDEs generated by the Δ_{G_a} on the pullback space $\mathbb{M}_0$ and the co-evolved space $\mathbb{M}_1$ (cf. [Karrasch/Schilling,subm]).

Other contributions in [F/Koltai,subm]

In addition to theory and numerics for **automatically detecting partially present coherent sets** and **timing of the loss of coherence:**

1. In [F/Koltai'22] we introduce *inflated dynamic Cheeger and Sobolev constants* (which are the standard constants associated with Δ_{G_a}) and prove a theorem relating these to dynamic Cheeger and Sobolev constants [F'15]. These geometric results **formalises trade-off of the regularity of the boundaries of semi-material coherent sets:** *more material means more irregular coherent sets.*
2. We collapse the spatial dimensions to produce a one-dimensional Sturm–Liouville ODE (in the time-coordinate) that provides
   - **finer information on spatial norms of eigenfunctions over time**.
   - a natural **quantification of instantaneous mixing experienced** by an eigenfunction and thus **identification of time intervals where mixing is greater than or less than average**.
3. As $a \to \infty$ we recover the process generated by Δ^D (time-copied on $\mathbb{M}_0$) from the SDEs generated by the Δ_{G_a} on the pullback space $\mathbb{M}_0$ and the co-evolved space $\mathbb{M}_1$ (cf. [Karrasch/Schilling,subm]).

Other contributions in [F/Koltai,subm]

In addition to theory and numerics for **automatically detecting partially present coherent sets** and **timing of the loss of coherence:**

1. In [F/Koltai'22] we introduce *inflated dynamic Cheeger and Sobolev constants* (which are the standard constants associated with Δ_{G_a}) and prove a theorem relating these to dynamic Cheeger and Sobolev constants [F'15]. These geometric results **formalises trade-off of the regularity of the boundaries of semi-material coherent sets:** *more material means more irregular coherent sets.*
2. We collapse the spatial dimensions to produce a one-dimensional Sturm–Liouville ODE (in the time-coordinate) that provides
   - **finer information on spatial norms of eigenfunctions over time**.
   - a natural **quantification of instantaneous mixing experienced** by an eigenfunction and thus **identification of time intervals where mixing is greater than or less than average**.
3. As $a \to \infty$ we recover the process generated by Δ^D (time-copied on $\mathbb{M}_0$) from the SDEs generated by the Δ_{G_a} on the pullback space $\mathbb{M}_0$ and the co-evolved space $\mathbb{M}_1$ (cf. [Karrasch/Schilling,subm]).

Other contributions in [F/Koltai,subm]

In addition to theory and numerics for **automatically detecting partially present coherent sets** and **timing of the loss of coherence:**

1. In [F/Koltai'22] we introduce *inflated dynamic Cheeger and Sobolev constants* (which are the standard constants associated with Δ_{G_a}) and prove a theorem relating these to dynamic Cheeger and Sobolev constants [F'15]. These geometric results **formalises trade-off of the regularity of the boundaries of semi-material coherent sets:** *more material means more irregular coherent sets.*
2. We collapse the spatial dimensions to produce a one-dimensional Sturm–Liouville ODE (in the time-coordinate) that provides
   - **finer information on spatial norms of eigenfunctions over time**.
   - a natural **quantification of instantaneous mixing experienced** by an eigenfunction and thus **identification of time intervals where mixing is greater than or less than average**.
3. As $a \to \infty$ we recover the process generated by Δ^D (time-copied on $\mathbb{M}_0$) from the SDEs generated by the Δ_{G_a} on the pullback space $\mathbb{M}_0$ and the co-evolved space $\mathbb{M}_1$ (cf. [Karrasch/Schilling,subm]).

Eigenfunctions of the inflated dynamic Laplacian

$$\Delta_{G_a} F(t,\cdot) = a^2 \partial_{tt} F(t,\cdot) + \Delta_{g_t} F(t,\cdot).$$

- Eigenfunctions[1] of Δ_{G_a} with **no spatial dependence** are easy to identify: $F^{\text{temp}}_{k,a}(t,x) := \cos(ak\pi t/\tau)$, $k \geq 1$, with eigenvalue $\Lambda^{\text{temp}}_{k,a} := -(a\pi k/\tau)^2$.
- We call these **temporal eigenfunctions**; they tell us nothing about coherent sets in space.
- The temporal eigenfunctions generate the eigenspace $\mathbb{S}^{\text{temp}}$.
- We are primarily interested in the orthogonal complement $\mathbb{S}^{\text{spat}} := (\mathbb{S}^{\text{temp}})^{\perp}$, containing the remaining **spatial eigenfunctions** (with non-trivial spatial dependence).

[1] applying Neumann b.c.s on the "time faces".

Upper/Lower: 2nd efunc./trajectories. Left/Right: $\mathbb{M}_0/\mathbb{M}_1$