Algorithm S1. Nonlinear Laplacian spectrum analysis (NLSA)

input : data array x of size $m \times S$
lag window q
Gaussian width ϵ
number of nearest neighbors b
number of Laplacian eigenfunctions l
output: array of spatial modes u in embedding space, of size $n \times l$, where $n = mq$
array of temporal modes v of size $s \times l$, where $s = S - 2q + 1$
vector of singular values σ of size l
arrays of spatio-temporal patterns $\{\tilde{x}^1, \dots, \tilde{x}^l\}$, each of size $m \times s$

% m: physical space dimension
% n: embedding space dimension
% S: number of input samples
% s: number of samples for which temporal modes are computed

% because of embedding and the normalization by ξ in Algorithm S2, $s < S$
% specifically, $v(i,:)$ and $\tilde{x}^k(:,i)$ correspond to $x(:, i + 2q - 1)$

begin time-lagged embedding
  % store embedding-space data in array X of size $n \times (s+1)$
  for $j \leftarrow 1 : s+1$ **do**
    for $i \leftarrow 1 : q$ **do**
      $i_1 \leftarrow (i-1) * m + 1$
      $i_2 \leftarrow i * m$
      $X(i_1 : i_2, j) \leftarrow x(1 : m, j + q - i)$
    end
  end
end

begin Laplacian eigenfunctions
  % eigenfunctions corresponding to the largest l eigenvalues of P are stored in array ϕ of size $s \times l$
  % $\phi(i,j)$ is the value of eigenfunction j evaluated at sample $X(:,i)$
  execute : algorithm S2 with inputs X, ϵ, b
  result : sparse transition probability matrix P of size $s \times s$
  vector μ of size s storing the Riemannian measure
  $\phi \leftarrow \mathsf{eigenvectors}(P, l)$ % compute the leading l eigenvectors of P
end

begin linear operator components and singular value decomposition
  % the $n \times l$ array A contains the operator components from [**8**] in the main text
  for $j \leftarrow 1 : l$ **do**
    for $i \leftarrow 1 : n$ **do**
      $A(i,j) \leftarrow \mathsf{sum}(X(i, 2 : s+1) * \mu(1 : s) * \phi(1 : s, j))$ % $*$ is element-wise array multiplication
    end
  end
  $[u, \sigma, v'] \leftarrow \mathsf{svd}(A)$ % v' has size $l \times l$
  for $k \leftarrow 1 : l$ **do**
    for $i \leftarrow 1 : s$ **do**
      $v(i,k) \leftarrow \mathsf{sum}(\phi(i, 1 : l) * v'(1 : l, k))$ % $*$ is element-wise array multiplication
    end
  end
  return u, σ, v
end

Algorithm continues on page 2.

Algorithm continued from page 1.

begin projection to physical space
    for $k \leftarrow 1 : l$ **do**
        $\tilde{x}^k(1 : m, 1 : s) \leftarrow 0$
        for $j \leftarrow 1 : s$ **do**
            $q' \leftarrow \mathsf{min}(q, s - j + 1)$ % for proper normalization near the end of the time interval
            for $i \leftarrow 1 : q'$ **do**
                $i_1 \leftarrow (i - 1) * m + 1$
                $i_2 \leftarrow i * m$
                $\tilde{x}^k(1 : m, j) \leftarrow \tilde{x}^k(1 : m, j) + u(i_1 : i_2, k) * \sigma(k) * v(j + i - 1, k)$
            end
            $\tilde{x}^k(1 : m, j) \leftarrow \tilde{x}^k(1 : m, j)/q'$
        end
        return $\tilde{x}^k$
    end
end

Algorithm S2. Transition probability matrix in diffusion map, following Coifman and Lafon [1].

input : data array X of size $n \times (s+1)$
Gaussian width ϵ
number of nearest neighbors b
output: sparse transition probability matrix P of size $s \times s$
Riemannian measure, stored in vector μ of size s

begin local velocity in embedding space for Gaussian width normalization
  % local velocity stored in vector ξ of size s
  % norm returns the norm of an n-dimensional vector in embedding space
  for $i \leftarrow 2$ **to** $s+1$ **do**
    $\xi(i-1) \leftarrow \mathsf{norm}\big(X(1:n,i) - X(1:n,i-1)\big)$
  end
end

begin distances and indices of b nearest neighbors
  % distances and indices are stored in arrays D and N of size $s \times b$
  for $i \leftarrow 1$ **to** s **do**
    for $j \leftarrow 1$ **to** s **do**
      $d(j) \leftarrow \mathsf{norm}\big(X(1:n,i+1) - X(1:n,j+1)\big)$
    end
    $[D(i,1:b), N(i,1:b)] \leftarrow \mathsf{partialSort}(d,b)$
    % $0 = D(i,1) < D(i,2) \leq \cdots \leq D(i,b)$ are the distances to the b nearest neighbors of sample i
    % $N(i,:)$ are the corresponding nearest-neighbor indices; i.e., $D(i,j) = d(N(j))$
  end
end

begin sparse weight matrix W
  % W has size $s \times s$
  $W(:,:) \leftarrow 0$ % initialize W to zero
  for $i \leftarrow 1 : s$ **do**
    for $j \leftarrow 1$ **to** b **do**
      $W(i,N(i,j)) \leftarrow \mathsf{exp}(-D(i,j)^2/(\epsilon * \xi(i) * \xi(j)))$
    end
  end
  $W \leftarrow \mathsf{sym}(W)$ % symmetrization is performed here
  % $\mathsf{sym}(W)(i,j) = W(j,i)$ if $W(i,j) = 0$, otherwise $\mathsf{sym}(W)(i,j) = W(i,j)$
  % symmetrized W has at least b nonzero elements per row
end

Algorithm continues on page 4.

Algorithm continued from page 3.

begin convert W to a transition probability matrix
    for $j \leftarrow 1 : s$ **do**
        $Q(j) \leftarrow \mathsf{sum}(W(:, j))$ % Q is a vector of size s
    end
    % normalize the rows and columns of W by Q
    for $i \leftarrow 1 : s$ **do**
        for $j \leftarrow 1 : s$ **do**
            $W(i, j) \leftarrow W(i, j)/Q(i)/Q(j)$
        end
    end
    % normalize the rows of W by the degree (connectivity) associated with W
    for $i \leftarrow 1 : s$ **do**
        $Q(i) \leftarrow \mathsf{sum}(W(i, 1 : s))$ % $Q(i)$ is the degree of sample i
    end
    for $i \leftarrow 1 : s$ **do**
        $P(i, 1 : s) = W(i, 1 : s)/Q(i)$
    end
    return P % P is a transition probability matrix because $\mathsf{sum}(P(i, :)) = 1$
    $\mu(1 : s) \leftarrow Q(1 : s)/\mathsf{sum}(Q(1 : s))$
    return μ % μ is a vector of size s with the property $\mu P = \mu$
end

Movie S1. Spatio-temporal patterns $\sum_k \tilde{x}_t^k$ of the upper 300 m temperature anomaly field (annual mean subtracted at each gridpoint, color-coded in $^\circ$C) evaluated using **[9]** and **[1]** for a 100-year portion of the data set. (a) Raw data. (b) the PDO mode from SSA. (c–f) NLSA patterns using (c) the annual modes, $k \in \{1, 2\}$ (see Figs. 1 and 2); (d) the leading-low-frequency (PDO) mode, $k = 3$; the semi-annual modes, $k \in \{6, 7\}$; (d) the leading two intermittent (Kuroshio) modes, $k \in \{9, 10\}$. The starting time of this animation is the same as in Figure 2.

Movie S2. 1000-day trajectories for the zonal modes $\{x_1, x_4\}$ in (a) the full model (Table S1), showing approximate locations of the zonal and blocked states; (b–d) reduced models constructed by projection onto the NLSA modes u_k from **[9]**.

1. **Coifman, R. R & Lafon, S. (2006) Diffusion maps. Appl. Comput. Harmon. Anal. 21, 5–30.**