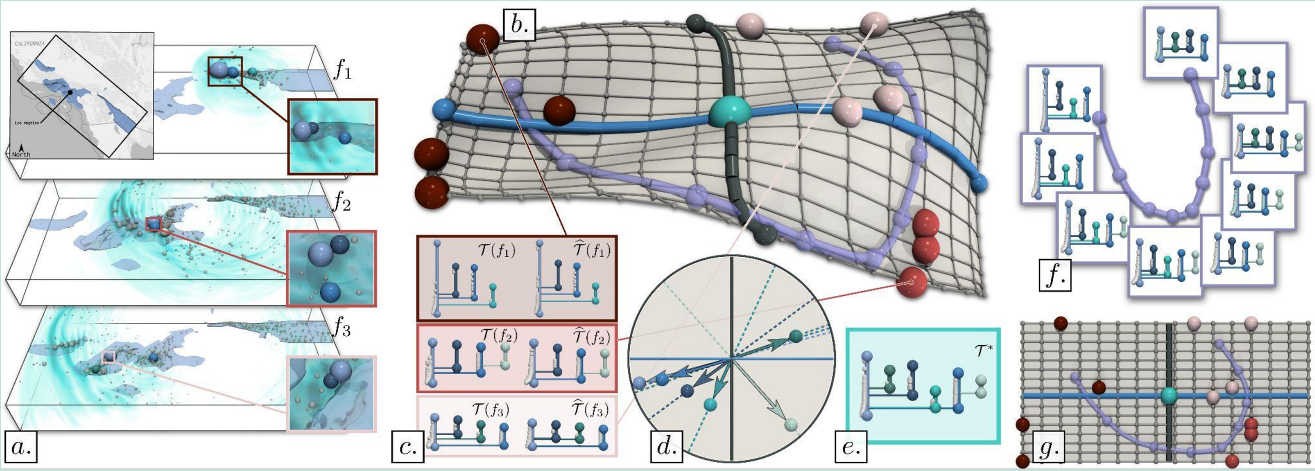
Principal Geodesic Analysis of Merge Trees (and Persistence Diagrams)

Of M. Pont, J. Vidal and J. Tierny



Samy H,
Nicolas C,
Mathias V,
Mathis Embit

MATHÉMATIQUES
VISION
APPRENTISSAGE

Ensemble datasets

Challenge : Several datasets for one phenomenon

-> Each dataset is called an ensemble member

-> Each ensemble member is represented by a Merge Tree

Merge Tree and BDT

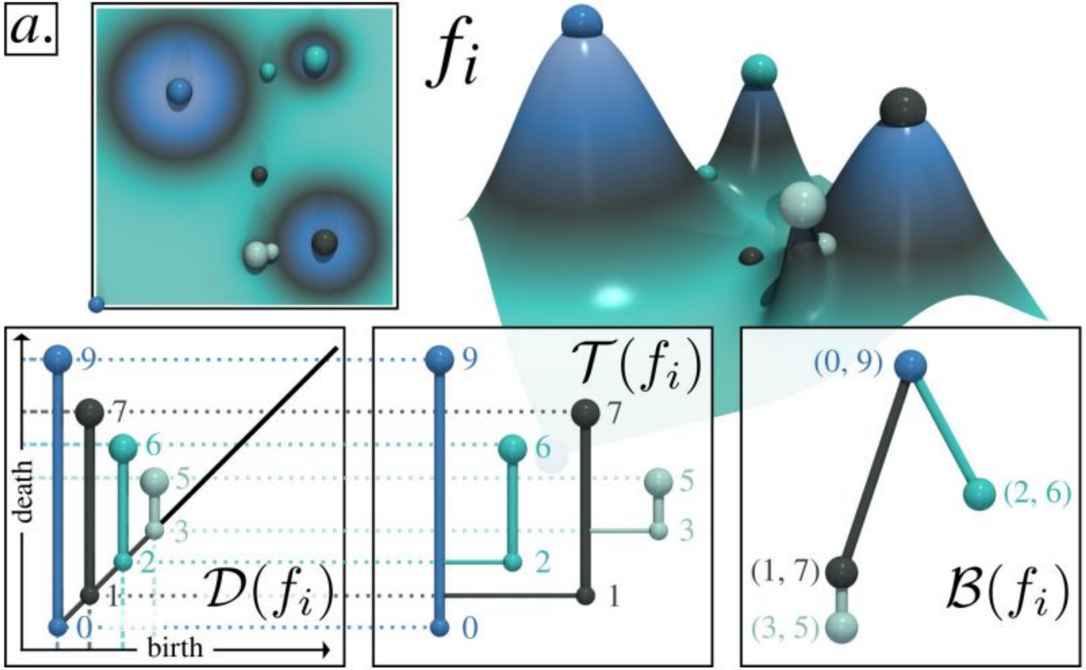
We consider a manifold and a Piecewise Linear scalar function on it

$$f : \mathcal{M} \to \mathbb{R}$$

Merge tree = M/~ where p1~p2 if f(p1) = f(p2) and p1,p2 in the same connected component of the sublevel (or superlevel) set of f at f(p1).

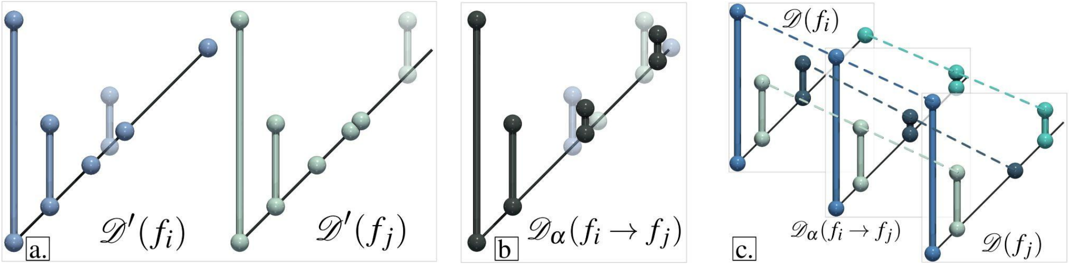
Branch Decomposition Tree (BDT) : A tree with nodes as the persistence branches and edges between adjacent branches.

Merge Tree and BDT



Wasserstein distance on persistence diagrams

- Partial assignment between
  - Off-diagonal points
    - Subset P_i
    - Complement $\overline{P_i}$
  - Bijection $\phi : P_i \to P_j$
  - Partial assignment $(\phi, \overline{P_i}, \overline{P_j}) \in \Phi$



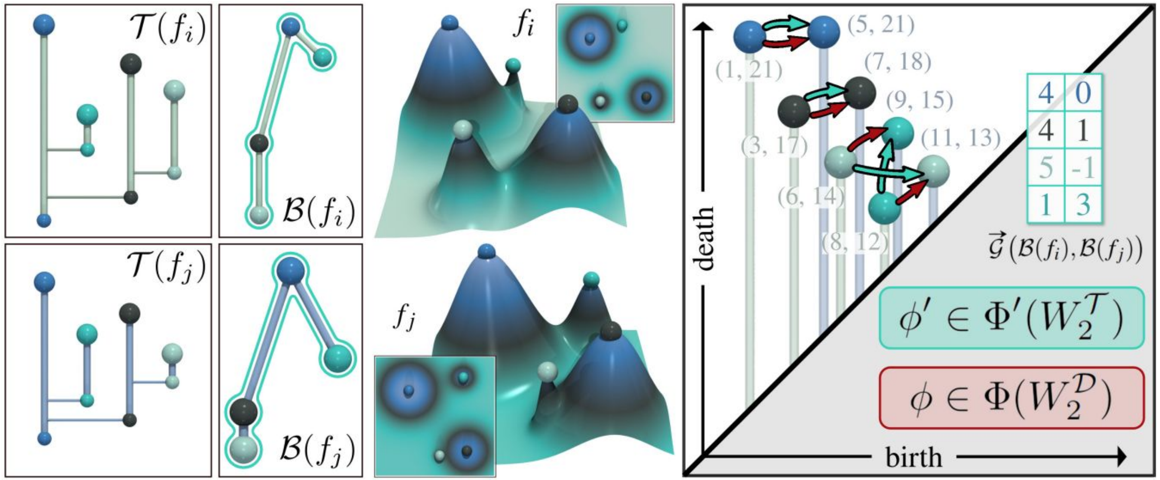
$$W_q(\mathcal{D}(f_i), \mathcal{D}(f_j)) = \min_{(\phi, \overline{P_i}, \overline{P_j}) \in \Phi} \Big(\sum_{p_i \in P_i} d_q(p_i, \phi(p_i))^q + \sum_{p_i \in \overline{P_i}} d_q(p_i, \Delta(p_i))^q + \sum_{p_j \in \overline{P_j}} d_q(\Delta(p_j), p_j)^q \Big)^{1/q}$$

Wasserstein distance on Merge trees

- Same, for q = 2 but $\phi : P_i \to P_j$ must be a partial rooted isomorphism

Partial isomorphism : Operates only on a subset P_i - no difference

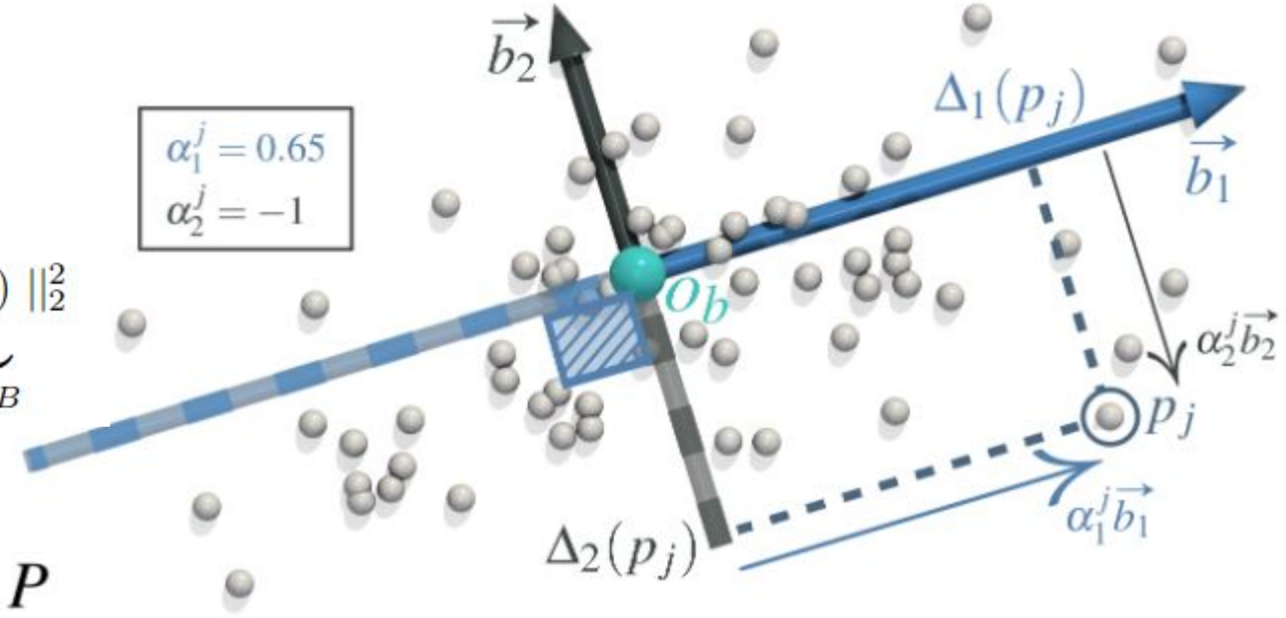
Rooted isomorphism : Conserves adjacency relationship.



Recall PCA in R^d

$$P = \{p_1, \ldots, p_N\}, p_j \in \mathbb{R}^d, \forall j$$
$$B = \{v_1, \ldots, v_q\}, v_i \cdot v_j = 0, \forall i \neq j$$

$$E_{L_2}(B) = \sum_{j=1}^{N} ||p_j - \underbrace{(\mathbf{o} + \sum_{i=1}^{q} \alpha_i^j v_i)}_{\text{Proj. of } p_j \text{ onto } B}||_2^2$$



From PCA to MT-PGA

A **geodesic** is a shortest path between two MT $\mathcal{G}(\mathcal{E},\mathcal{E}') = \underset{\phi' \in \Phi'}{\operatorname{argmin}} W_q^{\mathcal{T}}(\mathcal{E},\mathcal{E}')$

.

Geodesic dot product:

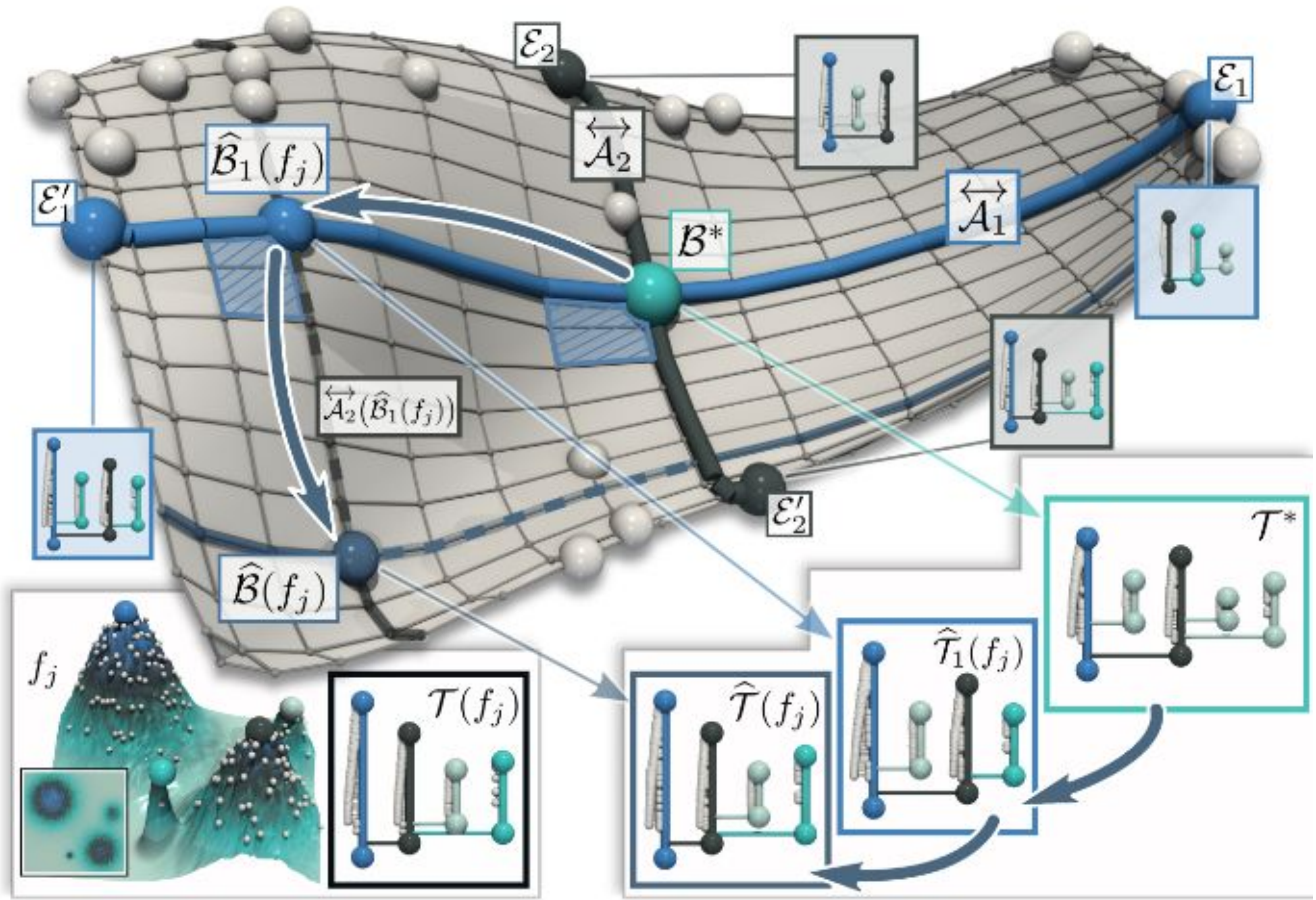
- Share extremity point
- Cartesian dot product in $\mathbb{R}^{2\times|\mathcal{E}|}$

Orthogonal → $\mathcal{G}(\mathcal{E},\mathcal{E}') \cdot \mathcal{G}(\mathcal{E},\mathcal{E}'') = 0$

Collinear→ $\mathcal{G}(\mathcal{E},\mathcal{E}') = \lambda \mathcal{G}(\mathcal{E},\mathcal{E}'')$

Geodesic axis is a pair of negatively collinear geodesics through the origin $\mathcal{E}_o$ → It's a geodesic and defines a direction vector $\mathcal{V}_i = \mathcal{G}(\mathcal{E}_i,\mathcal{E}_i')$ Between the extremities

From PCA to MT-PGA



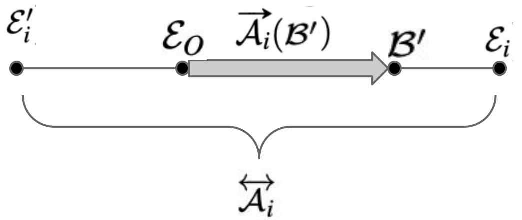
From PCA to MT-PGA

- BDT($\mathcal{B}$) geodesic axis ($\overleftrightarrow{\mathcal{A}_i}$) projection

$$\mathcal{B}_{\overleftrightarrow{\mathcal{A}_i}} = \underset{\mathcal{B}' \in \overleftrightarrow{\mathcal{A}_i}}{\arg\min} \left(W_2^{\mathcal{T}} \left(\mathcal{B}, \mathcal{B}' \right) \right)$$

- BDT ($\mathcal{B}' \in \overleftrightarrow{\mathcal{A}_i}$) geodesic axis arc-length parametrization $\overrightarrow{\mathcal{G}_i'} = \overrightarrow{\mathcal{G}}(\mathcal{E}_O, \mathcal{E}_i')$ $\overrightarrow{\mathcal{G}_i} = \overrightarrow{\mathcal{G}}(\mathcal{E}_O, \mathcal{E}_i)$

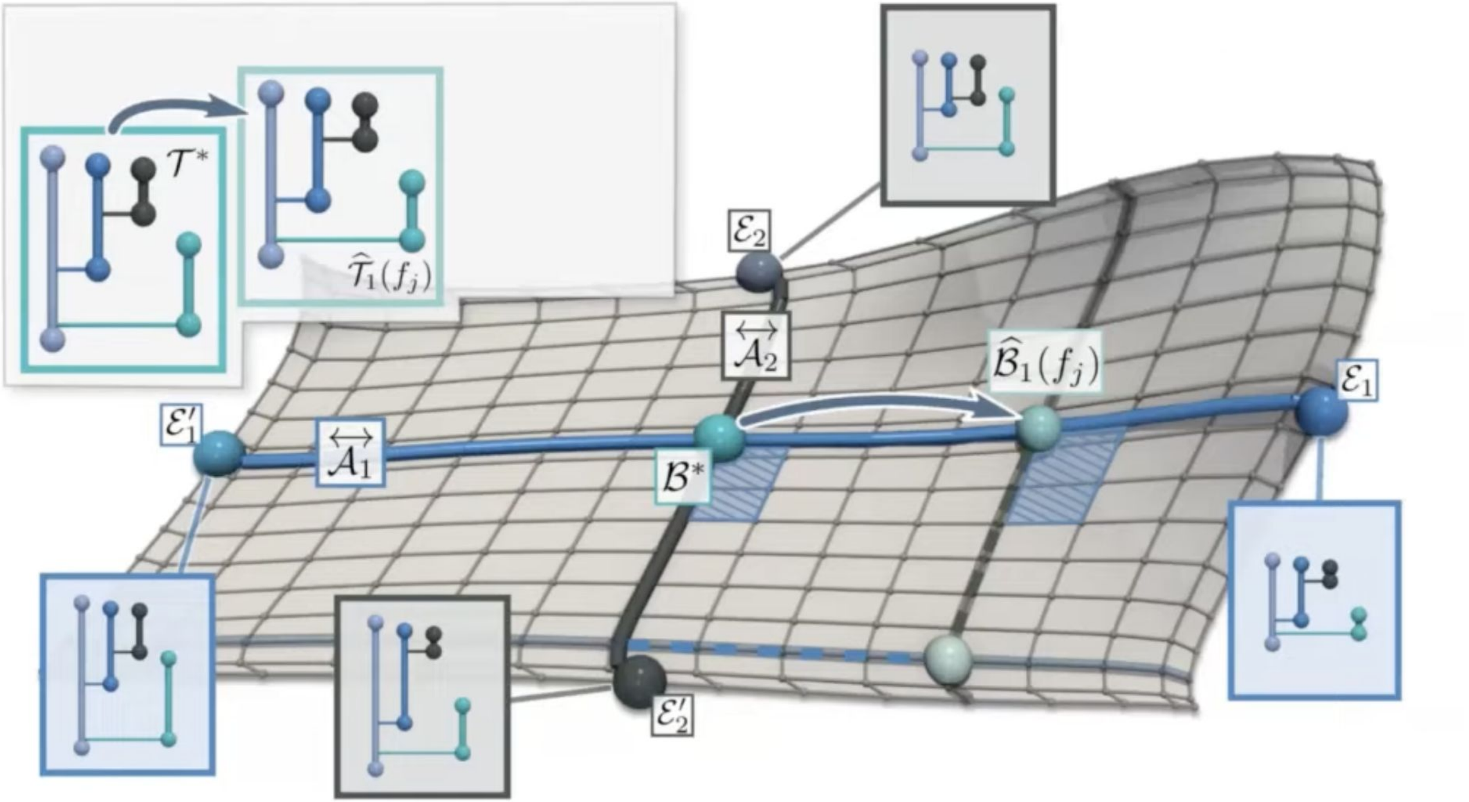
$$\mathcal{B}' = \mathcal{E}_O + \overrightarrow{\mathcal{A}_i}(\mathcal{B}') = \mathcal{E}_O + \alpha_i \times \overrightarrow{\mathcal{G}_i} + (1 - \alpha_i) \times \overrightarrow{\mathcal{G}_i'}.$$



- BDT($\mathcal{B}$) geodesic axis ($\overleftrightarrow{\mathcal{A}_i}$) projection

$$\mathcal{B}_{\overleftrightarrow{\mathcal{A}_i}} = \underset{\mathcal{B}' \in \overleftrightarrow{\mathcal{A}_i}}{\arg\min} \left(W_2^{\mathcal{T}} \left(\mathcal{B}, \mathcal{E}_O + \overrightarrow{\mathcal{A}_i}(\mathcal{B}') \right) \right)$$

- BDT axis translation: $\overleftrightarrow{\mathcal{A}}_j(\mathcal{B}) = ((\mathcal{B}, \mathcal{B} + \overrightarrow{\mathcal{G}_j}), (\mathcal{B}, \mathcal{B} + \overrightarrow{\mathcal{G}'_j}))$



From PCA to MT-PGA

- **BDT geodesic orthogonal basis**

  Orthogonal basis of d' geodesic axes $B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}_1}, \overleftrightarrow{\mathcal{A}_2}, \dots, \overleftrightarrow{\mathcal{A}_{d'}}\}$ with origine $\mathcal{E}_O$

  Project an arbitrary BDT $\mathcal{B}$ on $\overleftrightarrow{\mathcal{A}_1}$: $\widehat{\mathcal{B}}_1 = \mathcal{B}_{\overleftrightarrow{\mathcal{A}_1}} = \mathcal{B}_{\overleftrightarrow{\mathcal{A}_1}(\mathcal{E}_O)}$

  Recursively: $\widehat{\mathcal{B}}_{d'} = \mathcal{B}_{\overleftrightarrow{\mathcal{A}_{d'}}(\widehat{\mathcal{B}}_{d'-1})} \Longrightarrow \widehat{\mathcal{B}}_{d'} = \mathcal{E}_O + \sum_{i=1}^{d'} \overrightarrow{\mathcal{A}_i}(\widehat{\mathcal{B}}_i)$

- **MT-PGA formulation: distance to the constructed basis of BDT**

  $\mathcal{B}^*$: Wasserstein barycenter

$$\boxed{E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}}) = \sum_{j=1}^{N} W_2^{\mathcal{T}}\left(\mathcal{B}(f_j), \mathcal{B}^* + \sum_{i=1}^{d'} \overrightarrow{\mathcal{A}_i}(\widehat{\mathcal{B}}_i(f_j))\right)^2}$$

Algorithm

Finding an orthogonal basis of principal geodesics in the Wasserstein metric space of MTs

⇔

Constrained optimization problem

$$\min_{B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}_1}, \overleftrightarrow{\mathcal{A}_2}, \dots, \overleftrightarrow{\mathcal{A}_{2\times|\mathcal{B}^*|}}\}} \quad E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}}) = \sum_{j=1}^{N} W_2^{\mathcal{T}}\left(\mathcal{B}(f_j), \mathcal{B}^* + \sum_{i=1}^{d'} \overrightarrow{\mathcal{A}_i}(\widehat{\mathcal{B}}_i(f_j))\right)^2$$

subject to
- geodesics
- negative collinearity
- orthogonality

Algorithm

- For d’ in {1, …, d_max}
  - While $E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}})$ decreases
    - Find A_d’ such that it minimizes $E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}})$
    - enforce the 3 constraints on A_d’

As constraint enforcements are not necessarily compatible together -> while loop until convergence:

while $\overleftrightarrow{\mathcal{A}_{d'}}$ evolves **do**

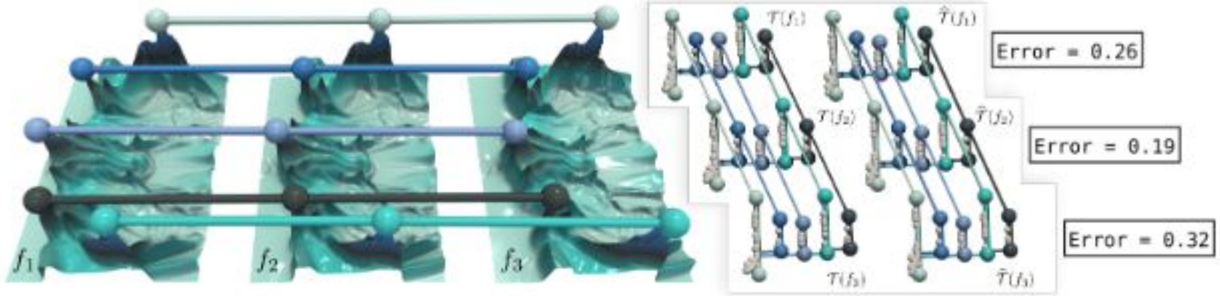
$\overleftrightarrow{\mathcal{A}_{d'}} \leftarrow \text{EnforceGeodesics}(\mathcal{B}^*, \overleftrightarrow{\mathcal{A}_{d'}});$

$\overleftrightarrow{\mathcal{A}_{d'}} \leftarrow \text{EnforceNegativeCollinearity}(\overleftrightarrow{\mathcal{A}_{d'}});$

$\overleftrightarrow{\mathcal{A}_{d'}} \leftarrow \text{EnforceOrthogonality}(B_{\mathbb{B}}, \overleftrightarrow{\mathcal{A}_{d'}});$

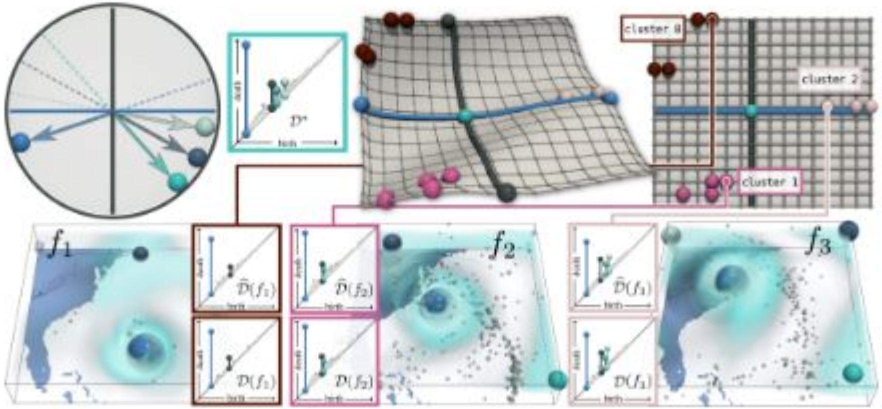
Applications

1. Data reduction → Lossy compression



2. Dimensionality reduction

3. Clustering



Thank you!