



**HAL**  
open science

# Principal Geodesic Analysis of Merge Trees (and Persistence Diagrams)

Mathieu Pont, Jules Vidal, Julien Tierny

► **To cite this version:**

Mathieu Pont, Jules Vidal, Julien Tierny. Principal Geodesic Analysis of Merge Trees (and Persistence Diagrams). IEEE Transactions on Visualization and Computer Graphics, In press, 10.1109/TVCG.2022.3215001 . hal-03737499

**HAL Id: hal-03737499**

**<https://hal.science/hal-03737499>**

Submitted on 25 Jul 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Principal Geodesic Analysis of Merge Trees (and Persistence Diagrams)

Mathieu Pont, Jules Vidal and Julien Tierny

**Abstract**—This paper presents a computational framework for the Principal Geodesic Analysis of merge trees (MT-PGA), a novel adaptation of the celebrated Principal Component Analysis (PCA) framework [87] to the Wasserstein metric space of merge trees [92]. We formulate MT-PGA computation as a constrained optimization problem, aiming at adjusting a basis of orthogonal geodesic axes, while minimizing a fitting energy. We introduce an efficient, iterative algorithm which exploits shared-memory parallelism, as well as an analytic expression of the fitting energy gradient, to ensure fast iterations. Our approach also trivially extends to extremum persistence diagrams. Extensive experiments on public ensembles demonstrate the efficiency of our approach – with MT-PGA computations in the orders of minutes for the largest examples. We show the utility of our contributions by extending to merge trees two typical PCA applications. First, we apply MT-PGA to *data reduction* and reliably compress merge trees by concisely representing them by their first coordinates in the MT-PGA basis. Second, we present a *dimensionality reduction* framework exploiting the first two directions of the MT-PGA basis to generate two-dimensional layouts of the ensemble. We augment these layouts with persistence correlation views, enabling global and local visual inspections of the feature variability in the ensemble. In both applications, quantitative experiments assess the relevance of our framework. Finally, we provide a lightweight C++ implementation that can be used to reproduce our results.

**Index Terms**—Topological data analysis, ensemble data, merge trees, persistence diagrams.



## 1 INTRODUCTION

WHETHER they are acquired or simulated, modern datasets are constantly gaining in detail and complexity, given the continuous improvement of acquisition devices or computing resources. This geometrical complexity is a difficulty for interactive data analysis and interpretation. This observation motivates the development of concise yet informative data representations, capable of encoding the main features of interest and visually representing them to the users. In that regard, Topological Data Analysis (TDA) [37] has demonstrated its ability to generically, robustly and efficiently reveal implicit structural patterns hidden in complex datasets, in particular in support of analysis and visualization tasks [59]. Examples of successful applications include turbulent combustion [24], [55], [68], material sciences [43], [57], [58], nuclear energy [74], fluid dynamics [64], bioimaging [4], [20], [29], quantum chemistry [16], [51], [79] or astrophysics [108], [110]. Among the feature representations studied in TDA (see Sec. 1.1), the merge tree [27] (Fig. 2), is a popular instance in the visualization community [20], [24], [29].

In many applications, on top of the increasing geometrical data complexity, an additional challenge emerges, related to *ensemble datasets*. These describe a phenomenon not only with a single dataset, but with a collection of datasets, called *ensemble members*, in order to characterize the variability of the phenomenon under study. In principle, a topological representation (like the merge tree) can be computed for each ensemble member. While this strategy has several practical advantages (direct representations of the features of interest, reduced memory footprint), it shifts the analysis problem from an ensemble of datasets to an ensemble of merge trees. Then, a major challenge consists in designing statistical tools for such an ensemble of topological

descriptors, to support its interactive analysis and interpretation. In this direction, a series of recent works focused on the notion of *average topological descriptor* [67], [92], [116], [118], [122], with applications to ensemble summarization and clustering. However, while such averages synthesize a topological descriptor which is well *representative* of the ensemble, they do not describe the topological variability of the ensemble.

This paper addresses this issue and goes beyond simple averages by adapting the celebrated framework of Principal Component Analysis (PCA) [87] to ensembles of merge trees. For that, we introduce the novel notion of “*Merge-Tree Principal Geodesic Analysis*” (MT-PGA), which captures the most informative geodesics (i.e. analogs of straight lines on the abstract space of merge trees) given the input ensemble, hence facilitating variability analysis and visualization. In particular, we formalize the computation of an orthogonal basis of principal geodesics in the Wasserstein metric space of merge trees [92] as a constrained optimization problem (Sec. 3), inspired by previous work on the optimal transport of histograms [30], [106], which we extend and specialize to merge trees. We introduce an efficient iterative algorithm (Sec. 4), which exploits an analytic expression of the energy gradient to ensure fast iterations. Moreover, we document accelerations with shared-memory parallelism. Extensive experiments (Sec. 6) indicate that our algorithm produces bases of acceptable reconstruction quality within minutes, for real-life ensembles extracted from public benchmarks. We illustrate the utility of our contribution in two applications. First, we show that the principal geodesic bases computed by our algorithm can result in an important compression of ensembles of merge trees, while still enabling a successful post-processing for typical visualization tasks such as feature tracking or ensemble clustering. Second, we present an extended application of our work to dimensionality reduction, for the visual inspection of the ensemble variability via two-dimensional embeddings, where we show that the views

• Mathieu Pont, Jules Vidal and Julien Tierny are with the CNRS and Sorbonne Université. E-mails: {firstname.lastname}@sorbonne-universite.fr

Manuscript received April 19, 2005; revised August 26, 2015.

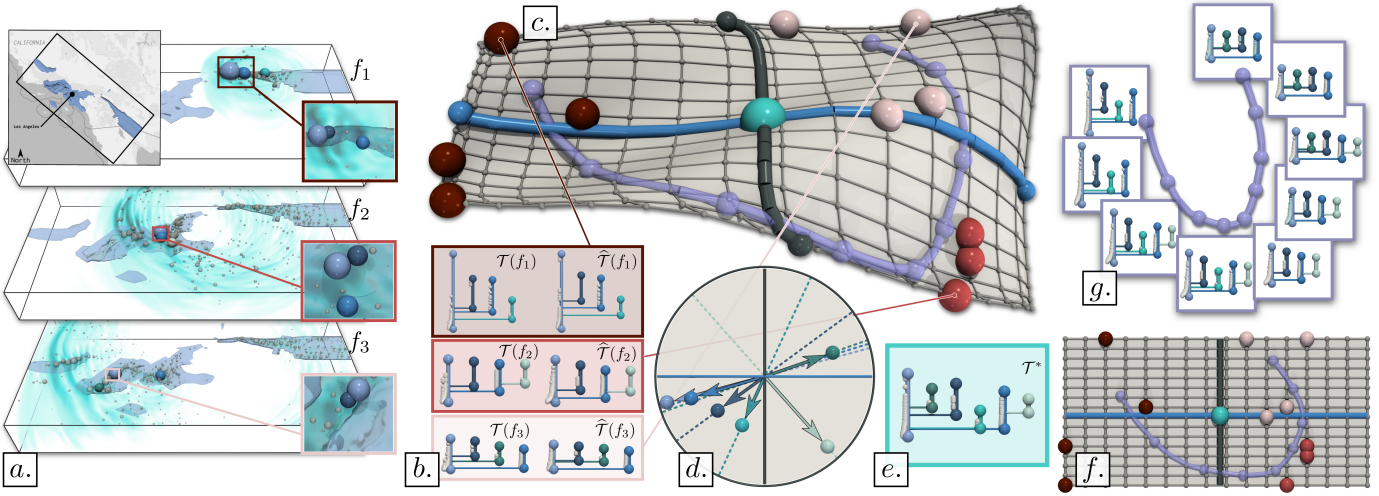


Fig. 1. Visual analysis of the *Earthquake* ensemble with Merge Tree Principal Geodesic Analysis (MT-PGA, (a): one member per ground-truth class). Our framework computes a coordinate system (c) for the Wasserstein metric space of merge trees  $\mathbb{B}$  by adjusting geodesic axes (blue and black curves, (c)) to optimize a fitting energy. This enables the adaptation to merge trees of typical applications of Principal Component Analysis, such as (i) data reduction, where the input trees are accurately reconstructed ((b), right), by simply storing their MT-PGA coordinates, or (ii) dimensionality reduction. MT-PGA enables the computation of *Principal Geodesic Surfaces* (c), which complement its planar layout (f) by better conveying visually the curved nature of  $\mathbb{B}$ . MT-PGA supports the efficient reconstruction of user-defined locations, for the interactive exploration of  $\mathbb{B}$ : the reconstruction of the purple curve (g) enables the navigation from the trees of the first cluster (dark red, (c)) to the second (orange, (c)) and third (pink, (c)) clusters. MT-PGA also introduces *Persistence Correlation Views* (d) which enable the visual identification of the features which are the most responsible for the variability in the ensemble (high correlation, near the disk boundary, (d)) as well as their direct inspection in the data (matching colors (a)).

generated by our approach preserve well the intrinsic metric between merge trees, as well as the global structure of the input ensembles. Since our framework is based on the Wasserstein distance between merge trees [92], which generalizes the Wasserstein distance between persistence diagrams [116], it trivially extends to persistence diagrams by simply adjusting a parameter.

## 1.1 Related work

The literature related to our work can be classified in three groups, reviewed in the following: (i) uncertainty visualization, (ii) ensemble visualization, and (iii) topological methods for ensembles.

**(i) Uncertainty visualization:** Variability in data can be represented in several ways. *Uncertain* datasets [1], [22], [62], [73], [83], [100] capture variability by modeling each point of the domain as a random variable, whose variability is explicitly modeled by a given probability density function (PDF). Early techniques focused on estimating the entropy of the random variables [98], their correlations [91] or their gradient variations [89]. The positional uncertainty of level sets has been studied for several interpolations and PDF models [6], [7], [8], [90], [94], [95], [96], [97], [105]. The positional uncertainty of critical points has been studied for Gaussian [70], [81], [82], [88] or uniform distributions [17], [54], [112]. A general limitation of existing methods for uncertain data is their dependence on the PDF model for which they have been specifically designed. This reduces their usability for ensemble data, where the PDF estimated from the ensemble members can follow an arbitrary, unknown model. Also, most techniques do not consider multimodal PDFs, which is however necessary when multiple trends are present in the ensemble.

**(ii) Ensemble visualization:** Ensemble datasets encode data variability by directly modeling empirical observations (i.e. the *members* of the ensemble). Current approaches to ensemble visualization typically compute some geometrical objects describing

the features of interest (level sets, streamlines, etc), for each member of the ensemble. Then, an aggregation phase estimates a *representative* object for the resulting ensemble of geometrical objects. For instance, spaghetti plots [35] are a typical example for studying level-set variability, especially for weather data [99], [104]. More specifically, box-plots [119] describe the variability of contours and curves [75]. For flow ensembles, Hummel et al. [61] introduce a Lagrangian framework for classification purposes. Clustering techniques have been introduced, to identify the main trends, and their variability, in ensembles of streamlines [44] and isocontours [45]. Regarding topological features, Favelier et al. [42] and Athawale et al. [9] introduced approaches for visualizing the variability of critical points and gradient separatrices. Lohfink et al. [71] introduced an approach for the consistent planar layout of multiple contour trees, to support effective visual comparisons between the contour trees of distinct members. Although the above techniques addressed the visualization of ensembles of topological objects, they did not focus explicitly on their statistical analysis. Principal Component Analysis (PCA) [87] is a classical approach for the analysis of variability in vectorized data (i.e. point clouds in Euclidean spaces). Extensions have been investigated for metric spaces [46], [49], [50], including transport based distances [32] between histograms [30], [106]. However, these methods are not directly applicable to merge trees. They focus on fundamentally different objects (histograms). Thus, their distances, geodesics and barycenters are defined differently (in particular in an entropic form [32], [33]) and the algorithms for their computations are drastically different (based on Sinkhorn matrix scaling [109]). Our global strategy (also based, at a high level, on an alternation of fitting and constraint enforcement, Sec. 3) can be interpreted as an extension of this line of work, but we revisit it completely, to specialize it to merge trees.

**(iii) Topological methods:** Concepts and algorithms from com-

putational topology [37] have been investigated, adapted and extended by the visualization community [59], [121]. Popular topological representations in data visualization include the persistence diagram [13], [37], [40] (Sec. 2.2), the Reeb graph [18], [36], [53], [84], [86], [115] and its variants the merge (Fig. 2) and contour trees [2], [27], [28], [52], [72], [113], or the Morse-Smale complex [23], [34], [38], [39], [47], [56], [102], [107].

As detailed in Sec. 3, the development of a computational framework for Principal Geodesic Analysis (PGA) over an ensemble of topological objects requires several key, low-level, geometrical ingredients, namely (i) a distance metric (to measure distances between objects), (ii) a geodesic estimation routine (to model the axes of the PGA basis) and (iii) a barycenter estimation routine (to compute the origin of the PGA basis). We review now the previous work related to these aspects.

Distance metrics have been studied for most of the above topological objects. Inspired by the literature in optimal transport [63], [76], the Wasserstein distance between persistence diagrams [37] (Sec. 2.2) has been extensively studied. It is based on a bipartite assignment problem, for which exact [78] and approximate [15], [65] implementations are publicly available [114]. However, the persistence diagram can lack specificity in its data characterization and more advanced topological descriptors, such as merge trees (Sec. 2.3), are often reported to differentiate datasets better [14], [77], [92], [111]. Several similarity measures have been introduced for Reeb graphs [60] and their variants [103]. However, since these measures are not metrics (the preservation of the triangle inequality is not specifically enforced), they are not conducive to the computation of geodesics. Stable distance metrics between Reeb graphs [12] and merge trees [77] have been studied from a theoretical point of view [21] but their computation, following an exponential time complexity, is not tractable for practical datasets in general. Distances with polynomial time computation algorithms have also been investigated. Beketayev et al. [14] focus on the *branch decomposition tree* (BDT, Sec. 2.3), and estimate their distances by iteratively reducing a target mismatch term over a significantly large search space. Sridharamurthy et al. [111] specialize efficient algorithms for computing constrained edit distances between trees [123] to the special case of merge trees, resulting in a distance which is computable for real-life datasets and with acceptable practical stability. Based on this edit distance, a generalization of the  $L_2$ -Wasserstein distance between persistence diagrams [116] has been introduced for merge trees [92], thereby enabling the efficient computation of distances, geodesics and barycenters of merge trees.

Regarding the estimation of a *representative* object from a set of topological representations, multiple approaches emerged recently. Several methods [67], [116], [118] have been introduced for the estimation of barycenters of persistence diagrams (or vectorized variants [3], [26]). A recent work [122] introduced a framework for computing a *1-center* of a set of merge trees (i.e. minimizing its maximum distance to the set), for an interleaving distance [48]. However, this approach requires pre-existing, reliable correspondence labels between the nodes of the input trees, which is not practical for real-life datasets (heuristics need to be considered). Also, the resulting representative merge tree is not a barycenter (it does not minimize a Fréchet energy, i.e. a sum of distances to the set). Thus, it cannot be used directly for PGA. In contrast, the barycentric framework of Pont et al. [92] automatically minimizes a Fréchet energy explicitly. Thus, the resulting barycenter merge tree can be directly used for

PGA. Another line of approaches aimed at directly applying the classical PCA (or variants from the matrix sketching literature [120]) to *vectorizations* of topological descriptors [5], [69], [101], i.e. by first converting each topological descriptor into a (high-dimensional) Euclidean vector and then leveraging traditional tools from linear algebra (e.g. classical PCA) on these vectors. However, vectorizations are in general subject to a number of limitations. They are prone to approximation errors (due to quantization and linearization artifacts), they can be difficult to revert (which challenges their usage for visualization applications) and their stability is not always established. In contrast, our approach *directly* manipulates merge trees in the Wasserstein metric space, and not linear approximations in a Euclidean space. This results in a faithful formulation of merge tree PGA, which ensures improved accuracy and interpretability (cf. experimental discussion, Sec. 6.2).

## 1.2 Contributions

This paper makes the following new contributions:

- 1) *An approach to Principal Geodesic Analysis of Merge Trees (MT-PGA)*: We formulate the definition of an orthogonal basis of principal geodesics in the Wasserstein metric space of merge trees [92] as a constrained optimization problem. Our formulation (Sec. 3) extends previous work on histograms [30], [106] and specializes it to merge trees.
- 2) *An optimization algorithm for MT-PGA*: We introduce an efficient optimization algorithm (Sec. 4) for MT-PGA. Each iteration alternates between (i) a minimization of the fitting energy of the MT-PGA basis and (ii) a constraint enforcement. Our algorithm exploits an analytic expression of the fitting energy gradient, to ensure fast iterations. We document accelerations based on shared-memory parallelism and report running times in the orders of minutes for real-life ensembles.
- 3) *An application to data reduction*: We present an application to data reduction (Sec. 5.1), where the merge trees of the input ensemble are significantly compressed, by solely storing the MT-PGA basis and the coordinates of the input merge trees in the basis. We illustrate the utility of this reduction with applications to feature tracking and ensemble clustering.
- 4) *An application to dimensionality reduction*: We present an application to dimensionality reduction (Sec. 5.2), by embedding each merge tree as a point in a planar view, based on its first two coordinates in the MT-PGA basis. We also contribute derived visualizations – the *Principal Geodesic Surface* and the *Persistence Correlation View* – which enable the visual inspection of the individual features which are the most responsible for the variability in the ensemble.
- 5) *Implementation*: We provide a C++ implementation of our algorithms that can be used for reproduction purposes: <https://github.com/MatPont/MT-PGA>

## 2 PRELIMINARIES

This section presents the theoretical background of our work. It contains definitions adapted from the Topology ToolKit [114]. It is structured as follows. First, we formalize the input data (Sec. 2.1). Second, we introduce a first topological data representation – the persistence diagram (Sec. 2.2) – which is closely related to the main representation studied in this paper – the merge tree (Sec. 2.3). Finally, we describe geometrical tools (e.g. metrics,



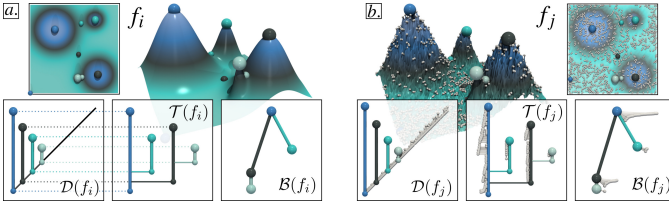


Fig. 2. Critical points (spheres, larger radius: maxima), persistence diagram (left inset), merge tree (center inset) and branch decomposition tree (right inset) of a clean (a) and noisy (b) scalar field. In both cases, four main hills are clearly represented with salient features in the persistence diagram and the merge tree. Branches with low persistence (less than 10% of the function range) are shown with small white arcs.

geodesics, barycenters) in the space of merge trees (Sec. 2.4), which have been recently generalized from the study of persistence diagrams [92]. These geometrical tools will act as building blocks in our formulation of MT-PGA (Sec. 3). We refer the reader to textbooks [37] for an introduction to computational topology.

## 2.1 Input data

The input data is an ensemble of  $N$  piecewise linear (PL) scalar fields  $f_i: \mathcal{M} \rightarrow \mathbb{R}$ , with  $i \in \{1, \dots, N\}$ , defined on a PL  $d$ -manifold  $\mathcal{M}$ , with  $d \leq 3$  in our applications. The *sub-level set* of  $f_i$ , noted  $f_{i-\infty}^{-1}(w) = \{p \in \mathcal{M} \mid f_i(p) < w\}$ , is defined as the pre-image of  $(-\infty, w)$  by  $f_i$ . The *super-level set* of  $f_i$  is defined symmetrically:  $f_{i+\infty}^{-1}(w) = \{p \in \mathcal{M} \mid f_i(p) > w\}$ . As  $w$  continuously increases, the topology of  $f_{i-\infty}^{-1}(w)$  changes at specific vertices of  $\mathcal{M}$ , called the *critical points* of  $f_i$  [11]. Critical points are classified by their *index*  $\mathcal{I}_i$ : 0 for minima, 1 for 1-saddles,  $d-1$  for  $(d-1)$ -saddles and  $d$  for maxima. In practice,  $f_i$  is enforced to contain only isolated, non-degenerate critical points [39], [41].

## 2.2 Persistence diagrams

The persistence diagram is a visual summary of the topological features (i.e. connected components, independent cycles, voids) of  $f_{i-\infty}^{-1}(w)$ . As shown in Fig. 2, it is closely related to the *merge tree*, which is the main topological data representation studied in this paper. We first describe the persistence diagram however, as the metric used in our work to measure distances between merge trees (Sec. 2.4) is a generalization of an established metric between persistence diagrams. Specifically, in the domain, each topological feature of  $f_{i-\infty}^{-1}(w)$  can be associated with a unique pair of critical points  $(c, c')$ , corresponding to its *birth* and *death*. The Elder rule [37] states that critical points can be arranged in pairs according to this observation, such that each critical point appears in only one pair  $(c, c')$ , with  $f_i(c) < f_i(c')$  and  $\mathcal{I}_i(c) = \mathcal{I}_i(c') - 1$ . For instance, if two connected components of  $f_{i-\infty}^{-1}(w)$  meet at a critical point  $c'$ , the *younger* component (created last, in  $c$ ) *dies*, in favor of the *older* one (created first). The persistence diagram  $\mathcal{D}(f_i)$  embeds each pair to a single point in 2D at coordinates  $(f_i(c), f_i(c'))$ . The *persistence* of a pair is given by its height  $f_i(c') - f_i(c)$ . The persistence diagram provides a visual overview of the features of a dataset (Fig. 2), where salient features stand out from the diagonal while pairs corresponding to noise are located near the diagonal.

## 2.3 Merge trees

In the following, we introduce the main topological data representation studied in this paper: the *merge tree*. We also describe

a specific representation of the merge tree called the *branch decomposition tree*, which can be interpreted as a generalization of the extremum persistence diagram, and which plays a central role in the computation of distances between merge trees (Sec. 2.4).

The *join tree*, noted  $\mathcal{T}^-(f_i)$ , is a visual summary of the connected components of  $f_{i-\infty}^{-1}(w)$  [27]. It is a 1-dimensional simplicial complex defined as the quotient space  $\mathcal{T}^-(f_i) = \mathcal{M} / \sim$  by the equivalence relation  $\sim$  which states that  $p_1$  and  $p_2$  are equivalent if  $f_i(p_1) = f_i(p_2)$  and if  $p_1$  and  $p_2$  belong to the same connected component of  $f_{i-\infty}^{-1}(f_i(p_1))$ .

The *split tree* (Fig. 2), noted  $\mathcal{T}^+(f_i)$ , is defined symmetrically and describes the connected components of the super-level set  $f_{i+\infty}^{-1}(w)$ . Each of these two *directed trees* is called a *merge tree* (MT), noted generically  $\mathcal{T}(f_i)$  in the following. Intuitively, these trees track the creation of connected components of the sub (or super) level sets at their leaves, and merge events at their interior nodes. To mitigate a phenomenon called *saddle swap*, these trees are often post-processed [92], [111], by merging adjacent saddles in the tree if their relative difference in scalar value is smaller than a threshold  $\epsilon_1 \in [0, 1]$ . MTs are often visualized according to a persistence-driven *branch decomposition* [85], to make the persistence pairs captured by the tree stand out. In this context, a *persistent branch* is a monotone path on the tree connecting the nodes corresponding to the creation and destruction (according to the Elder rule, Sec. 2.2) of a connected component of sub (or super) level set. Then, the branch decomposition provides a planar layout of the MT, where each persistent branch is represented as a vertical segment (center insets in Fig. 2). The *branch decomposition tree* (BDT), noted  $\mathcal{B}(f_i)$ , is a directed tree whose nodes are the persistent branches captured by the branch decomposition and whose arcs denote adjacency relations between them in the MT. In Fig. 2, the BDTs (right insets) can be interpreted as the dual of the branch decompositions (center insets, with matching colors): each vertical segment in the branch decomposition (center) corresponds to a node in the BDT (right) and each horizontal segment (center, denoting an adjacency relation between branches) corresponds to an arc in the BDT. The BDT can be interpreted as a generalization of the extremum persistence diagram: like  $\mathcal{D}(f_i)$ ,  $\mathcal{B}(f_i)$  describes the population of (extremum) persistence pairs present in the data. However, unlike the persistence diagram, it additionally captures adjacency relations between them (Fig. 2). Note that, the birth and death of each persistent branch  $b_i \in \mathcal{B}(f_i)$ , noted  $(x_i, y_i)$ , span by construction an interval included in that of its parent  $b'_i \in \mathcal{B}(f_i)$ :  $[x_i, y_i] \subseteq [x'_i, y'_i]$ . This *nesting property* of BDTs [92] is a direct consequence of the Elder rule (Sec. 2.2), and it plays an important role in the computation of geodesics between MTs (Sec. 2.4).

## 2.4 Wasserstein metric space

In the following, we introduce an established metric between persistence diagrams (the Wasserstein distance) which has been recently generalized to merge trees [92]. This generalization allows our PGA framework to support both persistence diagrams and merge trees (as discussed next). After introducing this metric, we detail further derived concepts, such as the notions of geodesics and Wasserstein barycenters, which will act as building blocks in our formulation of MT-PGA (Sec. 3).

To measure the distance between two diagrams  $\mathcal{D}(f_i)$  and  $\mathcal{D}(f_j)$ , a typical pre-processing step consists in augmenting each

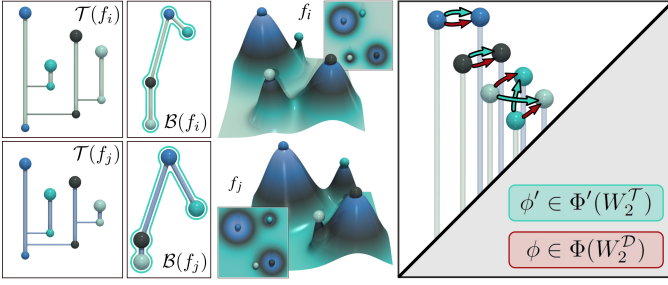


Fig. 3. The Wasserstein distance  $W_2^T$  between the BDTs  $\mathcal{B}(f_i)$  (white) and  $\mathcal{B}(f_j)$  (blue) is computed by assignment optimization (Eq. 1) in the 2D birth/death plane (cyan arrows, right), given a search space of assignments  $\Phi'$  describing partial rooted isomorphisms between  $\mathcal{B}(f_i)$  and  $\mathcal{B}(f_j)$  (cyan halo on the BDTs). For  $\varepsilon_1 = 1$ , the structure of the BDTs is fully ignored and  $W_2^T$  becomes equivalent to the classical Wasserstein distance between persistence diagrams  $W_2^D$  (red arrows), which reports a small distance in this example, despite a significant structural change in the data (two hills have been swapped from  $f_i$  to  $f_j$ ).

diagram with the diagonal projection of the off-diagonal points of the other diagram:

$$\begin{aligned} \mathcal{D}'(f_i) &= \mathcal{D}(f_i) \cup \{\Delta(p_j) \mid p_j \in \mathcal{D}(f_j)\} \\ \mathcal{D}'(f_j) &= \mathcal{D}(f_j) \cup \{\Delta(p_i) \mid p_i \in \mathcal{D}(f_i)\}, \end{aligned}$$

where  $\Delta(p_i) = (\frac{x_i+y_i}{2}, \frac{x_i+y_i}{2})$  stands for the diagonal projection of the off-diagonal point  $p_i = (x_i, y_i) \in \mathcal{D}(f_i)$ . Intuitively, this augmentation phase inserts dummy features in the diagram (with zero persistence, along the diagonal), hence preserving the topological information of the diagrams. This augmentation guarantees that the two diagrams now have the same number of points ( $|\mathcal{D}'(f_i)| = |\mathcal{D}'(f_j)|$ ), which facilitates the evaluation of their distance, as described next.

Given two points  $p_i = (x_i, y_i) \in \mathcal{D}'(f_i)$  and  $p_j = (x_j, y_j) \in \mathcal{D}'(f_j)$ , the ground distance  $d_q$  ( $q > 0$ ) in the 2D birth/death space is given by:

$$d_q(p_i, p_j) = (|x_j - x_i|^q + |y_j - y_i|^q)^{1/q} = \|p_i - p_j\|_q.$$

By convention,  $d_q(p_i, p_j)$  is set to zero between diagonal points ( $x_i = y_i$  and  $x_j = y_j$ ). Then, the  $L^q$ -Wasserstein distance  $W_q^D$  is:

$$W_q^D(\mathcal{D}'(f_i), \mathcal{D}'(f_j)) = \min_{\phi \in \Phi} \left( \sum_{p_i \in \mathcal{D}'(f_i)} d_q(p_i, \phi(p_i))^q \right)^{1/q} \quad (1)$$

where  $\Phi$  is the set of all possible assignments  $\phi$  mapping a point  $p_i \in \mathcal{D}'(f_i)$  to a point  $p_j \in \mathcal{D}'(f_j)$  (possibly its diagonal projection, indicating the destruction of the corresponding feature).

A variant of this metric, noted  $W_2^T(\mathcal{B}(f_i), \mathcal{B}(f_j))$ , has been introduced recently for BDTs [92]. Its expression is identical to Eq. 1 (for  $q = 2$ ), at the notable exception of the search space of possible assignments, noted  $\Phi' \subseteq \Phi$ , constrained to describe (rooted) partial isomorphisms [92] between  $\mathcal{B}(f_i)$  and  $\mathcal{B}(f_j)$  (cyan halo on the BDTs of Fig. 3). Intuitively,  $W_2^T$  can be understood as a variant of  $W_2^D$ , which takes into account the structures of the BDTs when evaluating candidate assignments in the optimization of Eq. 1. Since  $\Phi' \subseteq \Phi$ , we have:  $W_2^T(\mathcal{B}(f_i), \mathcal{B}(f_j)) \geq W_2^D(\mathcal{D}'(f_i), \mathcal{D}'(f_j))$ . In other words,  $W_2^T$  is more discriminative than  $W_2^D$ . In the special case where the parameter  $\varepsilon_1$  (Sec. 2.3) equals 1 (all the saddles in the MTs are merged), we have:  $W_2^T(\mathcal{B}(f_i), \mathcal{B}(f_j)) = W_2^D(\mathcal{D}'(f_i), \mathcal{D}'(f_j))$ .

In other words,  $W_2^T$  generalizes  $W_2^D$  and  $\varepsilon_1$  acts as a control parameter, which adjusts the importance of the structure of the BDTs in the metric. Then, our PGA framework will be able to support both persistence diagrams and merge trees, thanks to this metric generalization. Unless specified otherwise, we assume in the following that  $\varepsilon_1$  is set to the default recommended value:  $\varepsilon_1 = 0.05$  [92]. In the remainder, we note  $\mathbb{B}$  the metric space induced by the Wasserstein metric between BDTs.

Once a metric is established, we can now introduce the derived notions of geodesics and Wasserstein barycenters, which act as building blocks in our formulation of MT-PGA (Sec. 3).

A *geodesic* (i.e. length minimizing path on  $\mathbb{B}$ ) between two BDTs  $\mathcal{B}(f_i)$  and  $\mathcal{B}(f_j)$ , noted  $\vec{g}(\mathcal{B}(f_i), \mathcal{B}(f_j))$ , is simply given by linearly interpolating the optimal assignment  $\phi' \in \Phi'$  [92]. Technically, it is a vector in  $\mathbb{R}^{2 \times |\mathcal{B}(f_i)|}$ , obtained by concatenating the  $|\mathcal{B}(f_i)|$  two-dimensional vectors representing the optimal assignment  $\phi' \in \Phi'$  in the 2D birth/death space. Fig. 3 (right) illustrates two geodesics: one with respect to  $W_2^D$  ( $\varepsilon_1 = 1$ , red arrows), the other with respect to  $W_2^T$  ( $\varepsilon_1 = 0$ , cyan arrows). Note that an arbitrary vector in  $\mathbb{R}^{2 \times |\mathcal{B}(f_i)|}$  represents an arbitrary displacement in  $\mathbb{B}$ , which is consequently not necessarily optimal with regard to Eq. 1, and which therefore, does not necessarily represent a geodesic. To guarantee the invertibility of the interpolated BDTs into valid MTs, a local normalization, turning  $\mathcal{B}(f_i)$  into  $\mathcal{N}(\mathcal{B}(f_i))$ , needs to be introduced in a pre-processing step [92], where the persistence of each branch  $b_i \in \mathcal{B}(f_i)$  is normalized with regard to that of its parent  $b'_i \in \mathcal{B}(f_i)$ , displacing  $b_i$  in the 2D birth/space from  $(x_i, y_i)$  to  $\mathcal{N}(b_i) = (\mathcal{N}_x(b_i), \mathcal{N}_y(b_i))$ :

$$\begin{aligned} \mathcal{N}_x(b_i) &= (x_i - x'_i) / (y'_i - x'_i) \\ \mathcal{N}_y(b_i) &= (y_i - x'_i) / (y'_i - x'_i). \end{aligned} \quad (2)$$

After this pre-normalization, any interpolated BDT is reverted into a valid MT by recursively reverting Eq. 2, hence constructively enforcing the *nesting property* of BDTs ( $[x_i, y_i] \subseteq [x'_i, y'_i]$ , Sec. 2.3) thereby guaranteeing the validity of the interpolated MT. Pont et al. [92] introduce two parameters to tune the effect of this pre-normalization ( $\varepsilon_2$  balances the normalized persistence of small branches, selected via the threshold  $\varepsilon_3$ ). We set these to their default recommended values ( $\varepsilon_2 = 0.95$ ,  $\varepsilon_3 = 0.9$ ). In the remainder, we consider that all the input BDTs are normalized this way.

Once distances and geodesics for BDTs are available, the notion of *barycenter* can be introduced. Given a set  $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$ , let  $E_F(\mathcal{B})$  be the Fréchet energy of a candidate BDT  $\mathcal{B} \in \mathbb{B}$ :

$$E_F(\mathcal{B}) = \sum_{i=1}^N W_2^T(\mathcal{B}, \mathcal{B}(f_i))^2. \quad (3)$$

Then a BDT  $\mathcal{B}^* \in \mathbb{B}$  which minimizes  $E_F$  is called a *Wasserstein barycenter* of the set  $\mathcal{S}_{\mathcal{B}}$  (or its Fréchet mean under the metric  $W_2^T$ ). Eq. 3 can be optimized with an iterative algorithm [92], alternating assignment and update phases. After optimization, a barycenter MT can be obtained by recursively reverting Eq. 2.

### 3 FORMULATION

This section describes our novel extension of the Principal Component Analysis (PCA) framework to the Wasserstein metric space of merge trees, leading to the new notion of merge tree Principal Geodesic Analysis (MT-PGA). This section is structured as follows. First, we describe a geometric interpretation of PCA (Sec. 3.1), which our approach extends. Next, we describe how to

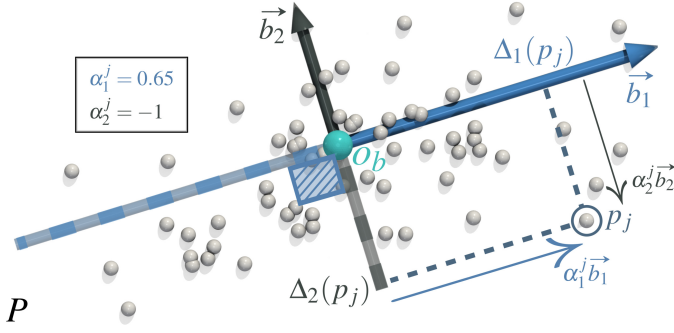


Fig. 4. The classical Principal Component Analysis (PCA) of a point cloud  $P$  in  $\mathbb{R}^d$  (white spheres) can be computed by iteratively optimizing the fitting to  $P$  of orthogonal directions (blue axis first, then black axis).

generalize each low-level geometrical tool used in PCA (Sec. 3.1) to the Wasserstein metric space of merge trees  $\mathbb{B}$  (Sec. 3.2). Finally, once these tools are available, we formalize our notion of MT-PGA as a constrained optimization problem (Sec. 3.3), for which we provide an algorithm in Sec. 4 (an overview of the algorithm is given in Sec. 4.1).

### 3.1 Geometric interpretation of PCA

Principal Component Analysis (PCA) [87] can be described with several, different, yet equivalent, formulations. In the following, we focus on a geometric interpretation (Fig. 4) which simplifies the transition to merge trees. Given a set of points  $P = \{p_1, p_2, \dots, p_N\}$  in a Euclidean space  $\mathbb{R}^d$ , PCA defines an orthogonal basis  $B_{\mathbb{R}^d} = \{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_d\}$  of vectors  $\vec{b}_i \in \mathbb{R}^d$  ( $i \in \{1, \dots, d\}$ ), with origin  $o_b \in \mathbb{R}^d$ , such that:

- 1) the origin  $o_b$  coincides with the arithmetic mean of  $P$ ,
- 2) the line  $l_i$  (defined by  $o_b$  and  $\vec{b}_i$ ) is orthogonal to all previous lines  $l_{i'}$  ( $i' \in \{1, \dots, i-1\}$ ) and minimizes its average squared distance to  $P$ .

Let  $\Delta_i(p_j)$  be the orthogonal projection of the point  $p_j$  on  $l_i$  (i.e.  $\Delta_i(p_j)$  is the closest point to  $p_j$  on  $l_i$ , Fig. 4). It can be expressed as a displacement from  $o_b$  on  $l_i$ :  $\Delta_i(p_j) = o_b + \alpha_i^j \vec{b}_i$ , with  $\alpha_i^j \in [-1, 1]$ . Then,  $B_{\mathbb{R}^d}$  can be formulated as an orthogonal basis which minimizes the following data fitting energy (with  $d' = d$  in general):

$$E_{L_2}(B_{\mathbb{R}^d}) = \sum_{j=1}^N \|p_j - (o_b + \sum_{i=1}^{d'} \alpha_i^j \vec{b}_i)\|_2^2. \quad (4)$$

In practice,  $B_{\mathbb{R}^d}$  can be optimized iteratively, one dimension at a time (starting with  $d' = 1$  and finishing with  $d' = d$ ), by finding at each iteration a vector  $\vec{b}_{d'}$  which is orthogonal to all the previous vectors in  $B_{\mathbb{R}^d}$  ( $\vec{b}_{d'} \cdot \vec{b}_{d''} = 0, \forall d'' \in \{1, \dots, d'-1\}$ ) and which minimizes  $E_{L_2}$  (Eq. 4). After all dimensions have been processed ( $d' = d$ ),  $B_{\mathbb{R}^d}$  provides a new coordinate system for  $P$  (composed of the vectors  $\vec{b}_i$  and the coordinates  $\alpha_i^j$  for each point  $p_j$ ), such that each direction of  $B_{\mathbb{R}^d}$  successively provides an optimal fit to  $P$  (Eq. 4). Note that, by construction, the variance of the projected data (i.e. the variance of  $\Delta_{d'}(p_j)$  along  $l_{d'}$ ) will be maximized for the first direction ( $d' = 1$ ) and it will keep on decreasing for increasing  $d'$ . This motivates early terminations of the optimization (for  $d' < d$ ), as the most *informative* directions are identified in the first iterations of the algorithm.

### 3.2 From PCA to MT-PGA

When the input data is not given as a point cloud in a Euclidean space (Sec. 3.1) but as an abstract set equipped with a metric, the above formulation of Principal Component Analysis (PCA) needs to be extended. Such an extension can be done with the more general notion of Principal Geodesic Analysis (PGA), which needs itself to be specifically instantiated given the specific metric space under study [30], [46], [106]. This instantiation consists in redefining the low-level geometrical tools used in PCA, but within the considered metric space. For instance, *geodesics* (length-minimizing paths) between merge trees (Sec. 2.4) extend to  $\mathbb{B}$  the notion of straight lines, and the Fréchet mean (Sec. 2.4) extends the arithmetic mean. In this section, we specifically formulate such an extension for merge trees (MTs). In particular, we formalize the following low-level geometrical notions for the Wasserstein metric space of merge trees  $\mathbb{B}$ :

- i) BDT geodesic dot product;
- ii) Orthogonal BDT geodesics;
- iii) Collinear BDT geodesics;
- iv) BDT geodesic axis;
- v) BDT geodesic axis arc-length parametrization;
- vi) BDT geodesic axis projection;
- vii) BDT geodesic axis translation;
- viii) BDT geodesic orthogonal basis.

The above notions derive sequentially from one another, to eventually result in the concept of *orthogonal bases of BDT geodesic axes*, which is precisely the main variable of the constrained optimization problem for MT-PGA formulation (Sec. 3.3).

**(i) BDT geodesic dot product.** A *geodesic* on the Wasserstein metric space of merge trees  $\mathbb{B}$  is the shortest path  $\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}')$  between its two extremity BDTs  $\mathcal{E}$  and  $\mathcal{E}'$ . As detailed in Sec. 2.4, it is an optimal assignment with regard to Eq. 1, which can be represented as a vector in  $\mathbb{R}^{2 \times |\mathcal{E}|}$ . Then, given two geodesics  $\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}') \in \mathbb{R}^{2 \times |\mathcal{E}|}$  and  $\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}'') \in \mathbb{R}^{2 \times |\mathcal{E}|}$  sharing an extremity  $\mathcal{E}$ , their *dot product*, noted  $\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}') \cdot \vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}'')$  can be naturally introduced by considering their Cartesian dot product (i.e. sum of component-wise products between two vectors in  $\mathbb{R}^{2 \times |\mathcal{E}|}$ ). Note that these two vectors must be consistently parametrized with regard to their common extremity  $\mathcal{E}$ : for both vectors, the  $i^{\text{th}}$  entry represents a 2D vector in the birth/death space modeling the optimal assignment (with regard to Eq. 1) of the  $i^{\text{th}}$  point of  $\mathcal{E}$  to points of  $\mathcal{E}'$  and  $\mathcal{E}''$  (small arrows, Fig. 3, right).

**(ii) Orthogonal BDT geodesics.** Two geodesics  $\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}')$  and  $\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}'')$  are *orthogonal* if  $\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}') \cdot \vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}'') = 0$ .

**(iii) Collinear BDT geodesics.** Two geodesics  $\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}')$  and  $\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}'')$  are *collinear* if  $\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}') = \lambda \vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}'')$ , with  $\lambda \in \mathbb{R}$ . In particular, they are *positively collinear* if  $\lambda > 0$ , and *negatively collinear* if  $\lambda < 0$ .

**(iv) BDT geodesic axis.** Now that the notion of collinear geodesics is available, we proceed to the introduction of the concept of *geodesic axis*. Given an *origin* BDT  $\mathcal{E}_O$  and two extremities  $\mathcal{E}_i$  and  $\mathcal{E}'_i$ , a *geodesic axis*, noted  $\vec{\mathcal{A}}_i$ , is defined as a pair of negatively collinear geodesics  $\vec{\mathcal{G}}_i = \vec{\mathcal{G}}(\mathcal{E}_O, \mathcal{E}_i)$ , and  $\vec{\mathcal{G}}'_i = \vec{\mathcal{G}}(\mathcal{E}_O, \mathcal{E}'_i)$ . It follows that a geodesic axis  $\vec{\mathcal{A}}_i$  is itself a geodesic between  $\mathcal{E}'_i$  and  $\mathcal{E}_i$ , which is guaranteed to pass through a given origin  $\mathcal{E}_O$ . An example of geodesic axis is given in Fig. 5 with the axis  $\vec{\mathcal{A}}_1$  (represented as a blue curve) between the BDTs  $\mathcal{E}'_1$  and  $\mathcal{E}_1$ , with origin  $\mathcal{B}^*$ . This notion will be instrumental in the definition of an orthogonal basis in  $\mathbb{B}$ , to constrain multiple geodesics to indeed pass through the origin of the basis, while still allowing for an optimization of



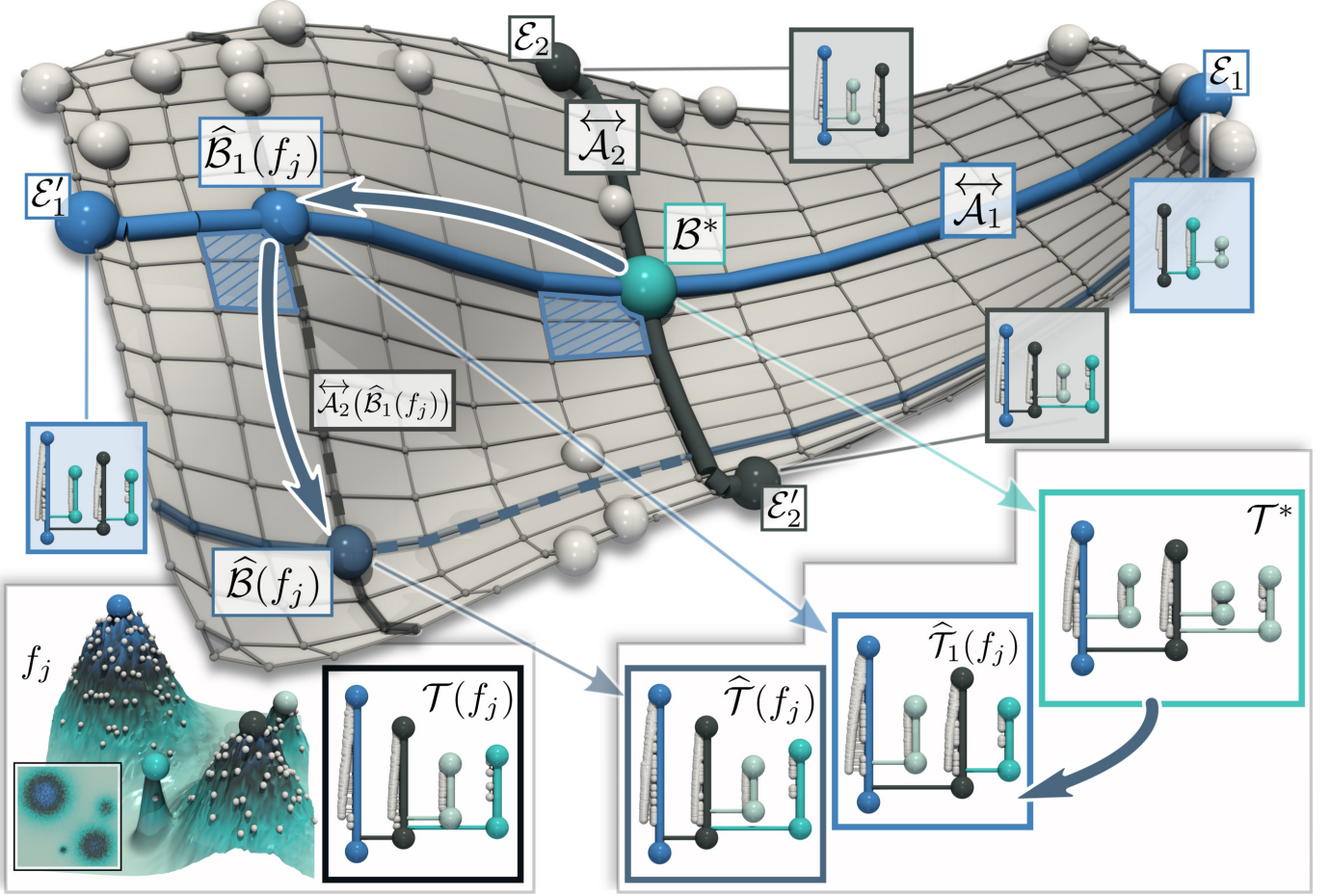


Fig. 5. Approach overview: Our merge tree Principal Geodesic Analysis (MT-PGA) defines a basis  $\mathcal{B}_{\mathbb{B}}$  on the Wasserstein metric space of BDTs  $\mathbb{B}$  (grey surface). It is computed by iteratively optimizing a fitting energy (Eq. 8) to an input ensemble of BDTs (white spheres) for orthogonal *geodesic* axes (blue:  $\overleftrightarrow{\mathcal{A}}_1$ , black:  $\overleftrightarrow{\mathcal{A}}_2$ ). Each axis (e.g.  $\overleftrightarrow{\mathcal{A}}_1$ ) is defined as a pair of *geodesics* between its extremities ( $\mathcal{E}_1$  and  $\mathcal{E}'_1$ ) and the Fréchet mean of the ensemble (cyan:  $\mathcal{B}^*$ ). After the optimization of the MT-PGA basis, any input BDT  $\mathcal{B}(f_j)$  can be reconstructed into  $\widehat{\mathcal{B}}(f_j)$  (dark blue sphere) by successive geodesic displacements (dark blue arrows, Eq. 9) along *translations* of the axes (e.g.  $\overleftrightarrow{\mathcal{A}}_2(\widehat{\mathcal{B}}_1(f_j))$ , black dashed curve), given the coordinates  $\alpha^j$  of  $\mathcal{B}(f_j)$  in the basis. While our framework manipulates BDTs, each of them can be directly inverted (transparent arrows) back to MTs (insets), resulting in reconstructions ( $\widehat{\mathcal{T}}(f_j)$ ) that are visually similar to the input ( $\mathcal{T}(f_j)$ ).

their extremities (Sec. 4). Each geodesic axis  $\overleftrightarrow{\mathcal{A}}_i$  is associated to a *direction vector*:  $\vec{\mathcal{V}}_i = \vec{\mathcal{G}}_i - \vec{\mathcal{G}}'_i = \vec{\mathcal{G}}(\mathcal{E}'_i, \mathcal{E}_i)$ .

(v) **BDT geodesic axis arc-length parametrization.** By construction, any BDT  $\mathcal{B}$  located on an axis  $\overleftrightarrow{\mathcal{A}}_i$  can be expressed with the following arc-length parametrization  $\overleftrightarrow{\mathcal{A}}_i(\mathcal{B})$  (with  $\alpha_i \in [0, 1]$ ):

$$\mathcal{B} = \mathcal{E}_O + \overleftrightarrow{\mathcal{A}}_i(\mathcal{B}) = \mathcal{E}_O + \alpha_i \times \vec{\mathcal{G}}_i + (1 - \alpha_i) \times \vec{\mathcal{G}}'_i. \quad (5)$$

This arc-length parametrization is the analog for the metric space  $\mathbb{B}$  of the Euclidean line parametrization used in PCA (Eq. 4).

(vi) **BDT geodesic axis projection.** The *projection*  $\mathcal{B}_{\overleftrightarrow{\mathcal{A}}_i}$  of an arbitrary BDT  $\mathcal{B}$  on the axis  $\overleftrightarrow{\mathcal{A}}_i$  is its closest BDT on  $\overleftrightarrow{\mathcal{A}}_i$ . It minimizes the following projection energy:

$$\begin{aligned} \mathcal{B}_{\overleftrightarrow{\mathcal{A}}_i} &= \arg \min_{\mathcal{B}' \in \overleftrightarrow{\mathcal{A}}_i} (W_2^T(\mathcal{B}, \mathcal{B}')) \\ &= \arg \min_{\mathcal{B}' \in \overleftrightarrow{\mathcal{A}}_i} (W_2^T(\mathcal{B}, \mathcal{E}_O + \overleftrightarrow{\mathcal{A}}_i(\mathcal{B}'))). \end{aligned} \quad (6)$$

An example of projection is given in Fig. 5, with the blue sphere noted  $\widehat{\mathcal{B}}_1(f_j)$ , which is the projection on the axis  $\overleftrightarrow{\mathcal{A}}_1$  (blue

curve) of an input BDT  $\mathcal{B}(f_j)$ . The projection  $\mathcal{B}_{\overleftrightarrow{\mathcal{A}}_i}$  of  $\mathcal{B}$  on  $\mathcal{A}_i$  will act as the analog, for the metric space  $\mathbb{B}$ , of the projection  $\Delta_i(p_j)$  of a point  $p_j$  along a line  $l_i$  in the Euclidean PCA (Sec. 3.1).

(vii) **BDT geodesic axis translation.** Let  $\overleftrightarrow{\mathcal{A}}_i$  and  $\overleftrightarrow{\mathcal{A}}_j$  be two axes sharing the same origin  $\mathcal{E}_O$ . The axis  $\overleftrightarrow{\mathcal{A}}_j(\mathcal{B})$  is called the *translation* of  $\overleftrightarrow{\mathcal{A}}_j$  along  $\overleftrightarrow{\mathcal{A}}_i$ , with origin  $\mathcal{B} \in \overleftrightarrow{\mathcal{A}}_i$  if:  $\overleftrightarrow{\mathcal{A}}_j(\mathcal{B}) = ((\mathcal{B}, \mathcal{B} + \vec{\mathcal{G}}_j), (\mathcal{B}, \mathcal{B} + \vec{\mathcal{G}}'_j))$ . Intuitively, a translated axis  $\overleftrightarrow{\mathcal{A}}_j(\mathcal{B})$  is a BDT axis, which is parallel to the axis  $\overleftrightarrow{\mathcal{A}}_j$ , and which passes through a specific BDT  $\mathcal{B}$ . In Fig. 5, an example is given with the axis  $\overleftrightarrow{\mathcal{A}}_2(\widehat{\mathcal{B}}_1(f_j))$  (black dashed curve), which is the translation of the axis  $\overleftrightarrow{\mathcal{A}}_2$  (black curve) at the BDT  $\widehat{\mathcal{B}}_1(f_j)$  (blue sphere). This notion of translated axis is needed when reconstructing a BDT given its coordinates in the MT-PGA basis, as detailed next.

(viii) **BDT geodesic orthogonal basis.** We now introduce the notion of orthogonal basis in the metric space of merge trees  $\mathbb{B}$ , which will be the variable at the core of the constrained optimization problem for MT-PGA formulation (Sec. 3.3). Let  $\mathcal{B}_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d'}\}$  be an *orthogonal basis* of  $d'$  geodesic axes (i.e. with pairwise orthogonal direction vectors  $\{\vec{\mathcal{V}}_1, \vec{\mathcal{V}}_2, \dots, \vec{\mathcal{V}}_{d'}\}$ ),

**Algorithm 1** Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

**Input:** Set of BDTs  $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$ .  
**Output1:** Basis origin  $\mathcal{B}^*$ ;  
**Output2:** Basis geodesic axes  $B_{\mathbb{B}} = \{\overleftarrow{\mathcal{A}}_1, \overleftarrow{\mathcal{A}}_2, \dots, \overleftarrow{\mathcal{A}}_{d_{\max}}\}$ ;  
**Output3:** Coordinates  $\alpha^j \in [0, 1]^{d_{\max}}$  of the input BDTs in  $B_{\mathbb{B}}$  (with  $j \in \{1, 2, \dots, N\}$ ).

- 1:  $\mathcal{B}^* \leftarrow \text{WassersteinBarycenter}(\mathcal{S}_{\mathcal{B}})$ ;
- 2: **for**  $d' \in \{1, 2, \dots, d_{\max}\}$  **do**
- 3:  $\overleftarrow{\mathcal{A}}_{d'} \leftarrow \text{InitializeGeodesicAxis}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}})$ ;
- 4: **while**  $E_{W_2^T}(B_{\mathbb{B}})$  decreases **do**
- 5: // Optimize the current geodesic axis  $\overleftarrow{\mathcal{A}}_{d'}$  (Sec. 4.2)
- 6:  $\alpha_{d'}^{j \in \{1, 2, \dots, N\}} \leftarrow \text{ProjectTrees}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;
- 7:  $E_{W_2^T}(B_{\mathbb{B}}) \leftarrow \text{EvaluateFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;
- 8:  $\overleftarrow{\mathcal{A}}_{d'} \leftarrow \text{OptimizeFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;
- 9: // Enforce the constraints (Sec. 4.3)
- 10: **while**  $\overleftarrow{\mathcal{A}}_{d'}$  evolves **do**
- 11:  $\overleftarrow{\mathcal{A}}_{d'} \leftarrow \text{EnforceGeodesics}(\mathcal{B}^*, \overleftarrow{\mathcal{A}}_{d'})$ ;
- 12:  $\overleftarrow{\mathcal{A}}_{d'} \leftarrow \text{EnforceNegativeCollinearity}(\overleftarrow{\mathcal{A}}_{d'})$ ;
- 13:  $\overleftarrow{\mathcal{A}}_{d'} \leftarrow \text{EnforceOrthogonality}(B_{\mathbb{B}}, \overleftarrow{\mathcal{A}}_{d'})$ ;
- 14: **end while**
- 15: **end while**
- 16: **end for**

with origin  $\mathcal{E}_O$ . Let  $\widehat{\mathcal{B}}_1$  be the projection of an arbitrary BDT  $\mathcal{B}$  on  $\overleftarrow{\mathcal{A}}_1$  ( $\widehat{\mathcal{B}}_1 = \mathcal{B} \xrightarrow{\overleftarrow{\mathcal{A}}_1} = \mathcal{B} \xrightarrow{\overleftarrow{\mathcal{A}}_1}(\mathcal{E}_O)$ , e.g. blue sphere on  $\overleftarrow{\mathcal{A}}_1$ , Fig. 5, noted  $\widehat{\mathcal{B}}_1(f_j)$ ). Now, let  $\widehat{\mathcal{B}}_2$  be the projection of  $\mathcal{B}$  on the translation of  $\overleftarrow{\mathcal{A}}_2$  along  $\overleftarrow{\mathcal{A}}_1$ , with origin  $\widehat{\mathcal{B}}_1$ :  $\widehat{\mathcal{B}}_2$  is the BDT of the translated axis  $\overleftarrow{\mathcal{A}}_2(\widehat{\mathcal{B}}_1)$  which is the closest to  $\mathcal{B}$  (i.e.  $\widehat{\mathcal{B}}_2 = \mathcal{B} \xrightarrow{\overleftarrow{\mathcal{A}}_2(\widehat{\mathcal{B}}_1)}$ , e.g. dark blue sphere on  $\overleftarrow{\mathcal{A}}_2(\widehat{\mathcal{B}}_1)$  in Fig. 5, noted  $\widehat{\mathcal{B}}(f_j)$ ).

By recursion, we can then define the *translated projection* of  $\mathcal{B}$  for the axis  $\overleftarrow{\mathcal{A}}_{d'}$ , noted  $\widehat{\mathcal{B}}_{d'}$ , as  $\widehat{\mathcal{B}}_{d'} = \mathcal{B} \xrightarrow{\overleftarrow{\mathcal{A}}_{d'}(\widehat{\mathcal{B}}_{d'-1})}$  (with  $\widehat{\mathcal{B}}_0 = \mathcal{E}_O$ ).

Intuitively, the translated projection  $\widehat{\mathcal{B}}_{d'}$  of  $\mathcal{B}$  is the closest BDT to  $\mathcal{B}$  (dark blue sphere in Fig. 5), which can be obtained by a sequence of  $d'$  geodesic displacements (dark blue arrows in Fig. 5) along the translated axes of the orthogonal basis  $B_{\mathbb{B}}$ .

By using Eq. 5,  $\widehat{\mathcal{B}}_{d'}$  can be associated with a collection of arc-length parameterizations  $\{\overleftarrow{\mathcal{A}}_1(\widehat{\mathcal{B}}_1), \overleftarrow{\mathcal{A}}_2(\widehat{\mathcal{B}}_2), \dots, \overleftarrow{\mathcal{A}}_{d'}(\widehat{\mathcal{B}}_{d'})\}$ , s.t.:

$$\widehat{\mathcal{B}}_{d'} = \mathcal{E}_O + \sum_{i=1}^{d'} \overleftarrow{\mathcal{A}}_i(\widehat{\mathcal{B}}_i). \quad (7)$$

$\widehat{\mathcal{B}}_{d'}$  can then be interpreted as the *reconstruction* of  $\mathcal{B}$  (up to the dimension  $d'$ ), given the basis  $B_{\mathbb{B}}$ : it is obtained by starting from the origin  $\mathcal{E}_O$  of the basis, and successively minimizing the projection energy (Eq. 10) along translated axes of  $B_{\mathbb{B}}$ , yielding a coordinate vector  $\alpha \in [0, 1]^{d'}$  (Eq. 5) which can be interpreted as the coordinates of  $\mathcal{B}$  in  $B_{\mathbb{B}}$ .

### 3.3 MT-PGA formulation

Now that the above low-level geometrical tools have been introduced for the Wasserstein metric space  $\mathbb{B}$ , we can formulate the MT-PGA by direct analogy to the classical PCA (Sec. 3.1).

Given a set  $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$  of input BDTs, let  $\mathcal{B}^*$  be their Fréchet mean (Eq. 3). Then, similarly to PCA, MT-PGA defines an orthogonal basis  $B_{\mathbb{B}} = \{\overleftarrow{\mathcal{A}}_1, \overleftarrow{\mathcal{A}}_2, \dots, \overleftarrow{\mathcal{A}}_{2 \times |\mathcal{B}^*|}\}$  of geodesic axes  $\overleftarrow{\mathcal{A}}_i$  ( $i \in \{1, \dots, 2 \times |\mathcal{B}^*|\}$ ) s.t.:

- 1) the origin  $\mathcal{E}_O$  of  $B_{\mathbb{B}}$  coincides with the Fréchet mean  $\mathcal{B}^*$  of  $\mathcal{S}_{\mathcal{B}}$ ,
- 2) the axis  $\overleftarrow{\mathcal{A}}_i$  is orthogonal to all axes  $\overleftarrow{\mathcal{A}}_{i'}$  ( $i' \in \{1, \dots, i-1\}$ ) and it minimizes its average squared Wasserstein distance to  $\mathcal{S}_{\mathcal{B}}$ .

Then,  $B_{\mathbb{B}}$  can be formulated as an orthogonal basis which minimizes the following (non-convex) fitting energy (with  $d' = 2 \times |\mathcal{B}^*|$  in general):

$$E_{W_2^T}(B_{\mathbb{B}}) = \sum_{j=1}^N W_2^T(\mathcal{B}(f_j), \mathcal{B}^* + \sum_{i=1}^{d'} \overleftarrow{\mathcal{A}}_i(\widehat{\mathcal{B}}_i(f_j)))^2. \quad (8)$$

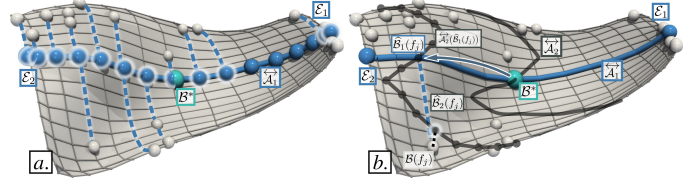


Fig. 6. Computing the *translated projection* of each input BDT  $\mathcal{B}(f_j)$  (white sphere). For the first axis (a),  $N_2$  samples evenly spaced on  $\overleftarrow{\mathcal{A}}_1$  are considered (blue spheres). Then, for each  $\mathcal{B}(f_j)$ , the sample which minimizes its  $W_2^T$  distance to  $\mathcal{B}(f_j)$  is selected as  $\widehat{\mathcal{B}}_1(f_j)$  (dashes). Next (b), the axis  $\overleftarrow{\mathcal{A}}_2$  is *translated* to  $\widehat{\mathcal{B}}_1(f_j)$  along  $\overleftarrow{\mathcal{A}}_1$  (arrow) yielding the translated axis  $\overleftarrow{\mathcal{A}}_2(\widehat{\mathcal{B}}_1(f_j))$ , which is also evenly sampled. The sample of  $\overleftarrow{\mathcal{A}}_2(\widehat{\mathcal{B}}_1(f_j))$  minimizing its  $W_2^T$  distance to  $\mathcal{B}(f_j)$  is selected as  $\widehat{\mathcal{B}}_2(f_j)$ .

The above equation is a direct analogy to the classical PCA (Eq. 4): the  $L_2$  norm is replaced by the Wasserstein distance  $W_2^T$ , the arithmetic mean  $o_b$  by  $\mathcal{B}^*$  and the term  $\alpha_i^j \overleftarrow{b}_i$  by  $\overleftarrow{\mathcal{A}}_i(\widehat{\mathcal{B}}_i(f_j))$ .

## 4 ALGORITHM

This section presents our novel algorithm, based on the constrained minimization of Eq. 8, for the estimation of an orthogonal basis  $B_{\mathbb{B}}$  for the Principal Geodesic Analysis of merge trees.

### 4.1 Overview

Alg. 1 provides an overview of our approach. Our algorithm takes an ensemble  $\mathcal{S}_{\mathcal{B}}$  of BDTs as an input and provides three outputs: (i) the basis origin  $\mathcal{B}^*$ , (ii) its  $d_{\max}$  geodesic axes  $\{\overleftarrow{\mathcal{A}}_1, \overleftarrow{\mathcal{A}}_2, \dots, \overleftarrow{\mathcal{A}}_{d_{\max}}\}$  (where  $1 \leq d_{\max} \leq 2 \times |\mathcal{B}^*|$  is an input parameter) and (iii) the coordinates  $\alpha^j \in [0, 1]^{d_{\max}}$  of the input BDTs in  $B_{\mathbb{B}}$  (Eq. 5, with  $j \in \{1, 2, \dots, N\}$ ).

First, the origin of the basis  $B_{\mathbb{B}}$  is computed as the Wasserstein barycenter  $\mathcal{B}^*$  of  $\mathcal{S}_{\mathcal{B}}$  (line 1). This is done with an iterative algorithm [92], which initializes  $\mathcal{B}^*$  at the BDT in  $\mathcal{S}_{\mathcal{B}}$  minimizing Eq. 3, and which further minimizes Eq. 3 by iteratively alternating assignment and update phases (see [92]). After this optimization has completed, only the  $N_1$  most persistent branches are kept in  $\mathcal{B}^*$ , where  $N_1$  is an input parameter controlling the memory footprint of the basis (see Sec. 4.5).

Next, the MT-PGA basis is computed by adapting the iterative strategy described in the case of PCA (Sec. 3.1). In particular, geodesic axes are computed one dimension at a time (starting with  $d' = 1$  and finishing with  $d' = d_{\max}$ , for loop, lines 2 to 16).

In particular, at each step  $d'$ :

- 1) The geodesics  $\overleftarrow{\mathcal{G}}_{d'}$  and  $\overleftarrow{\mathcal{G}}_{d'}'$  defining the axis  $\overleftarrow{\mathcal{A}}_{d'}$  are optimized in order to minimize Eq. 8 (line 8). This optimization (described in Sec. 4.2 and illustrated in Fig. 6) involves the translated projection  $\widehat{\mathcal{B}}_{d'}(f_j)$  of each input BDT  $\mathcal{B}(f_j)$  (line 6), providing an estimation of its coordinate  $\alpha_{d'}^j$  (Eq. 5).
- 2) The axis  $\overleftarrow{\mathcal{A}}_{d'}$  is updated into a valid solution (as described in Sec. 4.3 and illustrated in Fig. 7), by enforcing each constraint one after the other (lines 11 to 13).

This alternated procedure optimization/constraint is iterated until  $E_{W_2^T}(B_{\mathbb{B}})$  does not evolve significantly anymore (see Sec. 4.2). Once this is achieved, the current axis  $\overleftarrow{\mathcal{A}}_{d'}$  is considered as finalized and  $d'$  is incremented until  $\overleftarrow{\mathcal{A}}_{d_{\max}}$  is finalized.

## 4.2 Geodesic axis optimization

This section describes the optimization of a single axis  $\overleftrightarrow{\mathcal{A}}_{d'}$ .

First, the geodesic  $\overrightarrow{\mathcal{G}}_{d'}$  of  $\overleftrightarrow{\mathcal{A}}_{d'}$  is initialized by considering as its extremity  $\mathcal{E}_{d'}$  the input BDT in  $\mathcal{S}_{\mathcal{B}}$  which maximizes its Wasserstein distance to all the previous geodesic axes  $\overleftrightarrow{\mathcal{A}}_{d''}$ ,  $\forall d'' \in \{1, \dots, d' - 1\}$  (to  $\mathcal{B}^*$  when  $d' = 1$ ). The second geodesic  $\overrightarrow{\mathcal{G}}_{d'}$  is initialized at  $-\overrightarrow{\mathcal{G}}_{d'}$ .

Next, both  $\overrightarrow{\mathcal{G}}_{d'}$  and  $\overrightarrow{\mathcal{G}}_{d'}^*$  are optimized, to minimize Eq. 8. This is achieved with an iterative algorithm (Alg. 1, *while* loop, line 4) which, similarly to the minimization of the Fréchet energy [92], alternates an (i) *Assignment* and an (ii) *Update* phase. Intuitively, the (i) *Assignment* phase will compute geodesics (dashed blue curves, Fig. 6a) between the input BDTs (white spheres, Fig. 6a) and the axis to optimize (plain blue curve, Fig. 6a), while the (ii) *Update* phase will optimize freely the axis under these optimal assignments. After the (ii) *Update*, the initial assignments may no longer be optimal (since the axis has changed) and the *Assignment/Update* sequence needs to be iterated again, as detailed next.

The *Assignment* phase (i) computes a geodesic between each input BDT  $\mathcal{B}(j)$  and its translated projection  $\widehat{\mathcal{B}}_{d'}(f_j)$  (Fig. 6). In particular, these translated projections (Alg. 1, line 6) are estimated in practice by sampling the translated axis  $\overleftrightarrow{\mathcal{A}}_{d'}(\widehat{\mathcal{B}}_{d'-1}(f_j))$  along  $N_2$  evenly spaced samples and by selecting, for each  $\mathcal{B}(f_j)$ , the sample  $\widehat{\mathcal{B}}_{d'}(f_j)$  which minimizes Eq. 10.

Next, the *Update* phase (ii) consists in optimizing  $\overrightarrow{\mathcal{G}}_{d'}$  and  $\overrightarrow{\mathcal{G}}_{d'}^*$  to minimize Eq. 8 under the assignments computed by the *Assignment* phase (i). As detailed in Appendix A, for fixed assignments, the fitting energy  $E_{W_2^T}$  is convex with the variables  $\overrightarrow{\mathcal{G}}_{d'}$  and  $\overrightarrow{\mathcal{G}}_{d'}^*$ . We provide in Appendix A the analytic expression of the gradient of this energy and we derive the analytic expressions of  $\overrightarrow{\mathcal{G}}_{d'}$  and  $\overrightarrow{\mathcal{G}}_{d'}^*$  which minimize Eq. 8.

This overall *Assignment/Update* sequence is then iterated (Alg. 1, *while* loop, line 4). Each iteration decreases  $E_{W_2^T}$  constructively: while the *Update* phase (ii) minimizes it under the current assignment, the next *Assignment* phase (i) further improves (by construction) the assignments, hence decreasing  $E_{W_2^T}$  overall. In our implementation, the algorithm stops when the fitting energy has decreased by less than 1% between two consecutive iterations.

## 4.3 Constraints

The previous section described an algorithm for optimizing the geodesic axis  $\overleftrightarrow{\mathcal{A}}_{d'}$ , in order to minimize the fitting energy  $E_{W_2^T}$

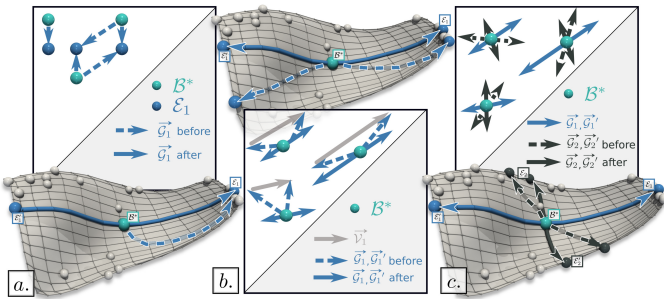


Fig. 7. Constraint enforcements: (a) Geodesic enforcement guarantees that the vector  $\overrightarrow{\mathcal{G}}_1$  describes an assignment which minimizes Eq. 1; (b) Collinearity enforcement constructively aligns  $\overrightarrow{\mathcal{G}}_1$  and  $\overrightarrow{\mathcal{G}}_1^*$ ; (c) Orthogonality enforcement updates  $\overrightarrow{\mathcal{G}}_2$  and  $\overrightarrow{\mathcal{G}}_2^*$  via *Gram-Schmidt* orthogonalization.

(Eq. 8). However, this algorithm did not consider yet key constraints present in the definition of MT-PGA (Sec. 3.2), such as axis orthogonality. We describe now our strategy for extending the above algorithm with a succession of constraint enforcements.

(i) **Geodesic enforcement.** After the axis  $\overleftrightarrow{\mathcal{A}}_{d'}$  has been optimized (Sec. 4.2) to minimize the fitting energy  $E_{W_2^T}$  (Eq. 8), its associated vectors  $\overrightarrow{\mathcal{G}}_{d'} \in \mathbb{R}^{2 \times |\mathcal{B}^*|}$  and  $\overrightarrow{\mathcal{G}}_{d'}^* \in \mathbb{R}^{2 \times |\mathcal{B}^*|}$  may represent displacements in the 2D birth/death space which are no longer geodesics in  $\mathbb{B}$  (as illustrated in Fig. 7a with the dashed arrows). Concretely, this situation occurs if the assignment described by the (optimized) vector  $\overrightarrow{\mathcal{G}}_{d'}$  between the barycenter  $\mathcal{B}^*$  and the vector extremity  $\mathcal{E}_{d'}$  is no longer optimal with regard to  $W_2^T$  (Eq. 1), as  $\overrightarrow{\mathcal{G}}_{d'}$  has been optimized freely when minimizing  $E_{W_2^T}$ . This can be easily corrected by re-computing the optimal assignment between  $\mathcal{B}^*$  and  $\mathcal{E}_{d'}$ , yielding an updated vector  $\overrightarrow{\mathcal{G}}_{d'}$ , now describing a valid geodesic between  $\mathcal{B}^*$  and  $\mathcal{E}_{d'}$ . In Fig. 7a, this is illustrated by the switch from the dashed arrows to the plain blue arrows. The same correction is applied to second vector of the axis,  $\overrightarrow{\mathcal{G}}_{d'}^*$ .

(ii) **Negative collinearity enforcement.** Up to now, the two geodesics  $\overrightarrow{\mathcal{G}}_{d'}$  and  $\overrightarrow{\mathcal{G}}_{d'}^*$  of the axis  $\overleftrightarrow{\mathcal{A}}_{d'}$  have been optimized independently. Then, they may not be negatively collinear and thus, they may not describe a valid geodesic axis, as defined in Sec. 3.2. Let  $\beta' = \|\overrightarrow{\mathcal{G}}_{d'}\| / (\|\overrightarrow{\mathcal{G}}_{d'}\| + \|\overrightarrow{\mathcal{G}}_{d'}^*\|)$  be the ratio describing the relative norm of  $\overrightarrow{\mathcal{G}}_{d'}$  with regard to  $\overleftrightarrow{\mathcal{A}}_{d'}$ . Negative collinearity is now constructively enforced by updating  $\overrightarrow{\mathcal{G}}_{d'}$  and  $\overrightarrow{\mathcal{G}}_{d'}^*$  (Fig. 7b) such that  $\overrightarrow{\mathcal{G}}_{d'} \leftarrow \beta' \times \overrightarrow{\mathcal{V}}_{d'}$  and  $\overrightarrow{\mathcal{G}}_{d'}^* \leftarrow -(1 - \beta') \times \overrightarrow{\mathcal{V}}_{d'}$ , where  $\overrightarrow{\mathcal{V}}_{d'}$  is the direction vector of  $\overleftrightarrow{\mathcal{A}}_{d'}$  (Sec. 3.2,  $\overrightarrow{\mathcal{V}}_{d'} = \overrightarrow{\mathcal{G}}_{d'} - \overrightarrow{\mathcal{G}}_{d'}^*$ ).

(iii) **Orthogonality enforcement.** We now describe the enforcement of the orthogonality of  $\overleftrightarrow{\mathcal{A}}_{d'}$  to all previous axes in the basis. In particular, given the direction vector  $\overrightarrow{\mathcal{V}}_{d'} = \overrightarrow{\mathcal{G}}_{d'} - \overrightarrow{\mathcal{G}}_{d'}^*$ , we want to update  $\overrightarrow{\mathcal{G}}_{d'}$  and  $\overrightarrow{\mathcal{G}}_{d'}^*$ , such that  $\overrightarrow{\mathcal{V}}_{d'} \cdot \overrightarrow{\mathcal{V}}_{d''} = 0$ , for all  $d'' \in \{1, 2, \dots, d' - 1\}$ . Let  $\mathcal{P}_{\overrightarrow{\mathcal{V}}_{d''}}(\overrightarrow{\mathcal{V}}_{d'}) = ((\overrightarrow{\mathcal{V}}_{d'} \cdot \overrightarrow{\mathcal{V}}_{d''}) / (\overrightarrow{\mathcal{V}}_{d''} \cdot \overrightarrow{\mathcal{V}}_{d''})) \times \overrightarrow{\mathcal{V}}_{d''}$  be the projection of  $\overrightarrow{\mathcal{V}}_{d'}$  onto the direction spanned by  $\overrightarrow{\mathcal{V}}_{d''}$ . The orthogonality of  $\overrightarrow{\mathcal{V}}_{d'}$  to all vectors  $\overrightarrow{\mathcal{V}}_{d''}$  is enforced (Fig. 7c) via *Gram-Schmidt* orthogonalization [31], by updating  $\overrightarrow{\mathcal{V}}_{d'}$  as follows:  $\overrightarrow{\mathcal{V}}_{d'} \leftarrow \overrightarrow{\mathcal{V}}_{d'} - \sum_{d''=1}^{d'-1} \mathcal{P}_{\overrightarrow{\mathcal{V}}_{d''}}(\overrightarrow{\mathcal{V}}_{d'})$ .

Note that the above constraints may go against each other: e.g. after orthogonality enforcement (iii),  $\overrightarrow{\mathcal{G}}_{d'}$  may no longer represent a valid geodesic (i). Thus, we iterate the above succession of constraint enforcements, until  $\overleftrightarrow{\mathcal{A}}_{d'}$  no longer evolves significantly (Alg. 1, line 10). In practice, using a constant number of iterations (specifically 4) is sufficient to obtain satisfactory results (see Sec. 6.3 for further details regarding convergence).

## 4.4 From BDTs to MTs

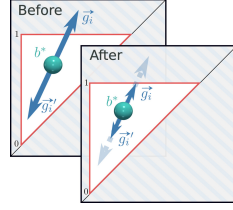
Given an MT-PGA basis, for an arbitrary coordinate vector  $\alpha \in [0, 1]^{d_{\max}}$ , the resulting BDT  $\widehat{\mathcal{B}}$  can be reconstructed (Sec. 3.2) by considering a BDT isomorphic to  $\mathcal{B}^*$  such that:

$$\widehat{\mathcal{B}} = \widehat{\mathcal{B}}_{d_{\max}} = \mathcal{B}^* + \sum_{i=1}^{d_{\max}} \alpha_i \times \overrightarrow{\mathcal{G}}_i + (1 - \alpha_i) \times \overrightarrow{\mathcal{G}}_i^*. \quad (9)$$

Applying the above computation embeds each branch  $b$  of  $\widehat{\mathcal{B}}$  as a point  $(x_b, y_b)$  in the 2D birth/death space (Fig. 3, right). Since we consider that the input BDTs have been pre-normalized (Sec. 2.4), a valid MT  $\widehat{\mathcal{T}}$  can in principle be reverted from  $\widehat{\mathcal{B}}$ , by recursively reverting Eq. 2, assuming that  $[x_b, y_b] \subseteq [0, 1]$  (to enforce the nesting property of BDTs, Sec. 2.3). However, the latter assumption is not explicitly enforced in our constrained optimization (Secs. 4.2 and 4.3). Thus, in our entire



framework (optimization included), when re-constructing a BDT (Eq. 9), for each branch  $b^* \in \mathcal{B}^*$ , the corresponding direction vector  $\vec{v}_i = \vec{g}_i - \vec{g}_i'$  (with  $\vec{g}_i$  and  $\vec{g}_i'$  being the entries of  $\vec{\mathcal{G}}_i$  and  $\vec{\mathcal{G}}_i'$  corresponding to the branch  $b^*$ ) is temporarily scaled down if needed (inset, right) by locally renormalizing  $\alpha_i$ , to guarantee that the extremities  $e_i$  ( $\alpha_i = 1$ ) and  $e_i'$  ( $\alpha_i = 0$ ) describe valid normalized birth/death locations (i.e.  $x_{e_i} < y_{e_i}$  and  $[x_{e_i}, y_{e_i}] \subseteq [0, 1]$ ).



#### 4.5 Computational parameters

As described in previous work [92], the Wasserstein metric  $W_2^T$  is subject to three parameters ( $\varepsilon_1$ ,  $\varepsilon_2$  and  $\varepsilon_3$ , Sec. 2.4), for which we use the recommended default values [92]. Otherwise, when switching  $\varepsilon_1$  to 1,  $W_2^T$  becomes equivalent to  $W_2^D$  (Sec. 2.4) and our framework computes then a PGA basis of extremum persistence diagrams (PD-PGA for short).

Our algorithm itself (Alg. 1) is subject to two parameters. (i)  $N_1$  controls the size of  $\mathcal{B}^*$ , and hence the memory footprint of the MT-PGA basis. In practice, we set  $N_1$  to a conservative value: 20% of the total number of branches in the ensemble ( $\sum_{j=1}^N |\mathcal{B}(f_j)|$ ). (ii)  $N_2$  controls the number of samples on each geodesic axis, to balance accuracy and speed. In practice, we set  $N_2$  to 16.

## 5 APPLICATIONS

This section illustrates the utility of our framework in concrete visualization tasks: data reduction and dimensionality reduction.

### 5.1 Data reduction

Like any other data representation, merge trees can benefit from lossy compression, to facilitate their storage or transfer. This can be particularly useful in scenarios where the scalar fields of the ensemble cannot all be stored permanently (because of their size or the induced IO bottleneck) but are represented instead individually by a topological signature (e.g. a persistence diagram or a merge tree), yielding an ensemble of signatures. This can be the case for instance during large data acquisition campaigns, or large-scale simulations, where a topological signature can be typically computed in-situ [10] to represent each time step [25]. In this scenario, lossy compression is useful to facilitate the manipulation

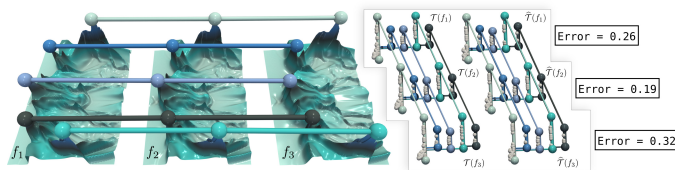


Fig. 8. Tracking features (the five most persistent maxima, spheres) in time-varying (from left to right) 2D data (ion density during universe formation [80]). The computed tracking [92] is identical, when using the input original merge trees (inset, left) or their versions compressed by MT-PGA (inset, right), which are highly similar visually. Here, since the input ensemble has few, small BDTs, the default reduction parameters ( $d_{max} = 3$  and  $N_1 = 20\%$ ) resulted in a modest compression factor (5.12). The reported relative reconstruction error (right) is given by the distance  $W_2^T$  between a tree and its reconstruction, divided by the maximum pairwise distance  $W_2^T$  observed in the input ensemble.

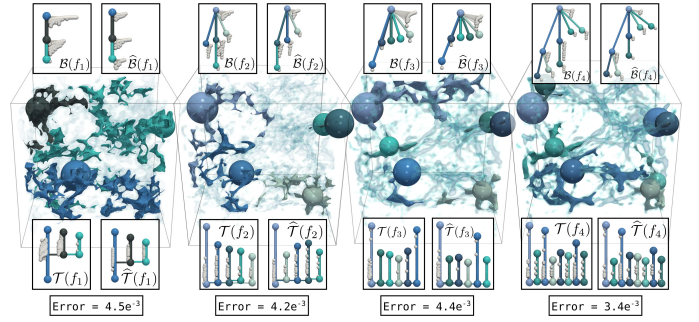


Fig. 9. Topological clustering of the 40 members of a cosmology ensemble [80] (one member is represented for each cluster). The computed clustering [92] is identical, when using the input original merge trees (bottom left inset) or their versions compressed by MT-PGA (bottom right inset). The optimal assignment between the left and right trees is visualized by the matching colors and the corresponding BDTs are reported at the top. Since this ensemble has many members, the default reduction parameters ( $d_{max} = 3$  and  $N_1 = 20\%$ ) resulted in an important compression factor (19.27). This compression still provided compressed trees (bottom right inset) that are visually similar to the input trees (bottom left inset): the number of prominent features (thick cylinders) are well preserved. This is confirmed with the reconstructed BDTs (top right inset) which are mostly isomorphic to the input BDTs (thick cylinders of matching color). The reported relative reconstruction error (bottom) is given by the distance  $W_2^T$  between a tree and its reconstruction, divided by the maximum pairwise distance  $W_2^T$  observed in the input ensemble.

(i.e. storage and transfer) of the resulting ensemble of merge trees. Previous work [92] has investigated the lossy compression of a temporal sequence of merge trees. In this section, we extend this work to arbitrary ensembles of merge trees thanks to MT-PGA.

Given its coordinates  $\alpha^j \in [0, 1]^{d_{max}}$ , each BDT  $\mathcal{B}(f_j)$  of  $\mathcal{S}_{\mathcal{B}}$  can be reliably estimated with Eq. 9. Thus, we present now an application to data reduction where the input ensemble of BDTs is compressed, by only storing to disk: (i) the origin  $\mathcal{B}^*$  of  $\mathcal{B}_{\mathbb{B}}$ , (ii) its axes  $\{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_{d_{max}}\}$  and (iii) the  $N$  BDT coordinates  $\alpha^j \in [0, 1]^{d_{max}}$ . The compression quality can be controlled with two input parameters (Sec. 4.1). (i)  $d_{max}$  controls the number of axes (and thus the ability of the basis to capture mild variabilities). (ii)  $N_1$  controls the size of  $\mathcal{B}^*$  (and thus the ability of the basis to capture small features). The reconstruction error (Eq. 8) will be minimized for large values of both parameters, while the compression factor will be maximized for low values. In our reduction experiments, we set  $d_{max}$  to 3 and  $N_1$  to its default value (Sec. 4.5). Figs. 8 and 9 show two examples of visualization applications (feature tracking and ensemble clustering, replicated from [92]) where the BDTs compressed with the above strategy ( $\hat{\mathcal{B}}(f_j)$ ) have been used as an input. In both cases, the output is identical to the outcome obtained with the *original* BDTs. This shows the viability of our reconstructed BDTs (and MTs) and demonstrates the utility of our reduction scheme. A detailed analysis of the resulting compression factors and reconstruction errors for all our test ensembles is available in Appendix B.

### 5.2 Dimensionality reduction

The MT-PGA basis  $\mathcal{B}_{\mathbb{B}}$  can also be used to generate 2D layouts of the ensemble, for its global visual inspection. This is achieved by embedding each input BDT  $\mathcal{B}(f_j)$  as a point in the 2D plane, given its first two coordinates in  $\mathcal{B}_{\mathbb{B}}$ . To prevent an artificial anisotropic distortion ( $\alpha^j \in [0, 1]^2$ ), we scale the coordinates of each BDT,

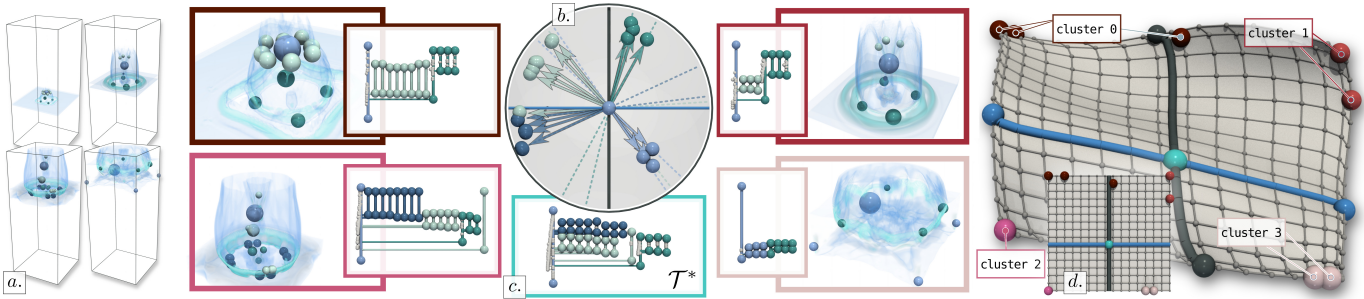


Fig. 10. Feature variability interpretation with *Persistence Correlation Views* (PCV) on the *Ionization (3D)* ensemble. Left (a): Four members of the ensemble (one per ground-truth class). Right (d): *Principal Geodesic Surface* (PGS) and its planar layout, computed by MT-PGA (sphere color: ground-truth classes). In the correlation view (b), several features (i.e. branches) are located near the disk boundary, indicating high correlations. The dark green features are highly correlated with the direction  $(0, \alpha_2)$ , given their location (near the black axis, upper half). Back in the data, this indicates that the corresponding features (basis of the ionization front, green spheres in the data and merge tree insets) will be the most persistent for the ensemble members with high  $\alpha_2$  values (clusters 0 and 1, (d)). Light green features in the PCV (upper left quadrant, (b)) are highly correlated with the direction  $(-\alpha_1, \alpha_2)$ : these features (front extremities, insets) will be the most persistent in the top left corner of the PGS (d, cluster 0). This is confirmed visually in the data (zoom insets), where the maxima are represented by spheres of matching color (the radius denotes persistence).

to account for the variation in length of the axes, and we embed each BDT at coordinates  $(\alpha_1^j \times W_2^T(\mathcal{E}_1, \mathcal{E}'_1), \alpha_2^j \times W_2^T(\mathcal{E}_2, \mathcal{E}'_2))$ . This results in a summarization view of the overall ensemble, which groups together BDTs which are close (given the metric  $W_2^T$ ) and which exhibit a similar variability with regard to  $\mathcal{B}^*$ . Note that, given an arbitrary point of the 2D layout, its BDT (and MT) can be efficiently reconstructed with Eq. 9. This enables interactive navigations within the ensemble (Fig. 1g). We augment our planar layouts with the following two improvements, to further characterize the global and local variability in the ensemble.

(i) **Principal geodesic surface.** To visually convey the curved (i.e. non-Euclidean) nature of the Wasserstein metric space  $\mathbb{B}$ , we introduce a 3D embedding of our planar layouts, which we call *Principal Geodesic Surface* (PGS). First, our 2D layout is sampled along a  $N_x \times N_y$  regular grid  $G_{\mathbb{B}}$  (in practice,  $N_x = N_y = N_2$ ). For each vertex of  $G_{\mathbb{B}}$ , a BDT can be reconstructed (Eq. 9), enabling the computation of a distance matrix  $\mathbb{D}_{G_{\mathbb{B}}}$ , where  $\mathbb{D}_{G_{\mathbb{B}}}(i, j)$  is the  $W_2^T$  distance between the vertices  $i$  and  $j$  of  $G_{\mathbb{B}}$ . Then,  $G_{\mathbb{B}}$  is embedded in 3D by multidimensional scaling [66] of  $\mathbb{D}_{G_{\mathbb{B}}}$ . The resulting PGS provides the same visual information as the planar layout, but in the form of a surface in 3D, parameterized by  $\alpha_1$  and  $\alpha_2$  (Fig. 5), which conveys visually the curved nature of  $\mathbb{B}$ .

(ii) **Persistence correlation view.** As detailed in Appendix C, we compute the correlation  $\rho(p_i, \alpha_k)$  between the coordinate  $\alpha_k$  and the persistence of the  $i^{\text{th}}$  branch (i.e. the branch in the input BDTs mapped to the  $i^{\text{th}}$  branch of  $\mathcal{B}^*$  given the optimal assignment induced by  $W_2^T$ , Eq. 1). Then, the  $i^{\text{th}}$  branch of the barycenter BDT  $\mathcal{B}^*$  can be embedded in a *Persistence Correlation View* (PCV), by placing an arrow between the origin and the point  $(\rho_{p_i, \alpha_1}, \rho_{p_i, \alpha_2})$ . This enables a *local* interpretation of the feature variability within the ensemble. Specifically, it enables the visual identification of the features whose persistence is strongly correlated with a given direction in the MT-PGA basis. Such features are located on the disk boundary of the correlation view (largest arrows in Figs. 1, 10, 11), and they are the most responsible for the variability in the ensemble. For each of these features, their matching to the origin  $\mathcal{B}^*$  of the MT-PGA basis is encoded with the color map, which enables their direct inspection in the data.

Together, our *Principal Geodesic Surface* (PGS) coupled with our *Persistence Correlation View* (PCV) enable both a global and

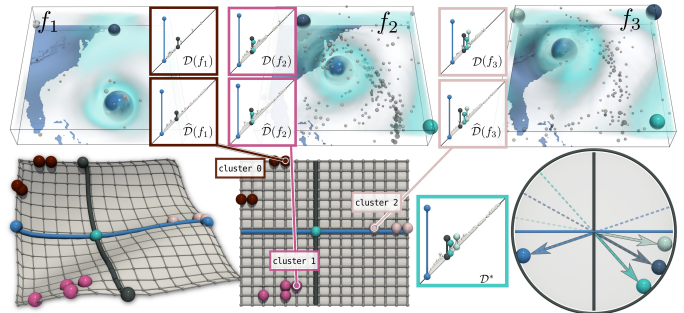


Fig. 11. Feature variability interpretation for the *Isabel* ensemble, with PD-PGA. In the PCV (bottom right), the blue feature (eye of the hurricane) is highly correlated with the direction  $(-\alpha_1, 0)$ , which indicates that its persistence will be stronger for the leftmost points in the PGS (clusters 0 and 1) and weaker for the rightmost points (cluster 2). In other words, the winds in the eye of the hurricane are significantly weaker in the cluster 2 (landfall phase of the hurricane). On the contrary, the other highly correlated features (right half of the PCV) are correlated with the direction  $(\alpha_1, 0)$ : their persistence will be stronger for the rightmost points (cluster 2). This is confirmed visually when inspecting the corresponding features in the data and the input diagrams (top). Similarly to MT-PGA, PD-PGA enables a data reduction (compression factor: 5.49), while guaranteeing visually similar reconstructed diagrams  $(\hat{\mathcal{D}}(f_j))$ .

local inspection of the feature variability in the ensemble. In particular, in the example of Fig. 1, our visualization indicates overall that the global maximum of the seismic wave (largest branch in the merge trees, purple, Fig. 1b) is mostly correlated with the direction  $(-\alpha_1, 0)$  in the PCV (Fig. 1d), and therefore, mostly prominent in the cluster 1 (located on the left of the PGS, Fig. 1f, for the lowest  $\alpha_1$  values). In contrast, the other features (blue, cyan and dark blue branches) are less correlated with this direction (shorter arrows in the PCV Fig. 1d), which indicates that they are slightly less variable through this direction: their persistence decreases less quickly than the global maximum when transitioning from cluster 1 to 2 and 3 (dark red, orange and pink spheres respectively in the PGS Fig. 1f). This indicates that the initial energy of the seismic wave (represented by the global maximum) quickly spreads into multiple wavefronts (spheres of matching colors in the data, Fig. 1a), whose individual energy decreases through time

TABLE 1

Running times (in seconds) of our algorithm for PD-PGA and MT-PGA computation (for  $d_{max} = 2$ , first sequential, then with 20 cores).

Dataset	N	B	PD-PGA			MT-PGA		
			1 c.	20 c.	Speedup	1 c.	20 c.	Speedup
Asteroid Impact (3D)	7	1,295	1,392.17	147.40	9.44	1,180.72	117.97	10.01
Cloud processes (2D)	12	1,209	817.64	61.88	13.21	517.94	38.49	13.46
Viscous fingering (3D)	15	118	86.69	9.17	9.45	42.89	4.71	9.11
Dark matter (3D)	40	2,592	18,388.86	1,366.45	13.46	24,480.42	1,758.04	13.92
Volcanic eruptions (2D)	12	811	460.17	37.99	12.11	1,004.37	81.75	12.29
Ionization front (2D)	16	135	104.74	12.00	8.73	55.73	6.26	8.90
Ionization front (3D)	16	763	3,750.00	300.96	12.46	4,029.71	294.29	13.69
Earthquake (3D)	12	1,203	3,896.52	338.64	11.51	1,973.49	158.12	12.48
Isabel (3D)	12	1,338	1,969.79	164.49	11.98	1,472.54	115.66	12.73
Starting Vortex (2D)	12	124	17.71	2.72	6.51	11.51	1.65	6.98
Sea Surface Height (2D)	48	1,787	12,420.98	670.00	18.54	27,791.00	1,669.52	16.65
Vortex Street (2D)	45	23	18.75	2.69	6.97	35.79	3.93	9.11

more slowly than the global maximum (each cluster represents a different temporal phase in the simulation, from top to bottom in Fig. 1a). Figs. 10 and 11 present a similar analysis and we refer the reader to the detailed captions for specific interpretations. Note that for Fig. 11, the features do not exhibit a clear global structure and the persistence diagram can be used instead of the merge tree.

## 6 RESULTS

This section presents experimental results obtained on a computer with two Xeon CPUs (3.2 GHz, 2x10 cores, 96GB of RAM). The input merge trees were computed with FTM [52] and pre-processed to discard noisy features (persistence simplification threshold: 0.25% of the data range). We implemented our approach in C++ (with the OpenMP task runtime), as modules for TTK [19], [114]. Experiments were performed on the benchmark of public ensembles [93] described in [92], which includes a variety of simulated and acquired 2D and 3D ensembles extracted from previous work and past SciVis contests [80].

### 6.1 Time performance

Barycenters and geodesics are computed with the approach of Pont et al. [92], which implements fine-grain task-based shared-memory parallelism. Specifically, during axis projection,  $N \times N_2$  geodesics need to be computed (Sec. 4.2), each geodesic requiring typically  $\mathcal{O}(|B|^2)$  steps in practice, where  $|B|$  is the size of the input BDTs. In practice, this is the most expensive part of our algorithm. These geodesics are computed concurrently (by submitting each geodesic to the task pool). In comparison, the evaluations of the numerical expressions (Alg. 1, lines 7, 8, 12 and 13) have a nearly negligible cost. Tab. 1 evaluates the time performance of our framework for persistence diagrams (PD-PGA) and merge trees (MT-PGA), for  $d_{max} = 2$ . In sequential mode, the running time is indeed a function of the size of the ensemble ( $N$ ) and the size of trees ( $|B|$ ). It is slightly slower for MT-PGA than for PD-PGA, but timings remain comparable overall. In parallel, speedups are the most important for the largest ensembles. However, the iterative nature of our algorithm has an impact on parallel efficiency (the end of each loop implies a synchronization). Still, our parallelization significantly reduces computation times, with less than 6 minutes on average and at most 30 minutes for the largest ensembles, which we believe is an acceptable pre-processing time, prior to interactive exploration.

### 6.2 Framework quality

Figs. 8 and 9 report compression factors for our application to data reduction (Sec. 5.1). These are ratios between the storage size of the input BDTs and that of the MT-PGA basis (barycenter,

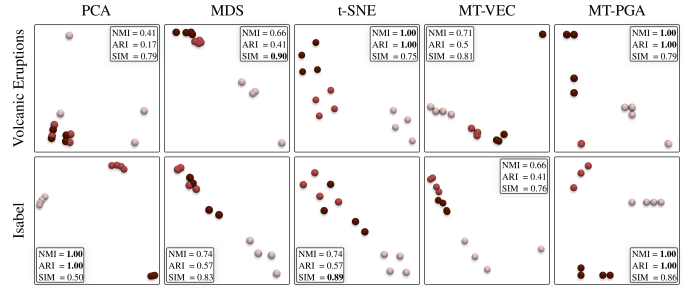


Fig. 12. Comparison of planar layouts for typical dimensionality reduction techniques, on two input ensembles (bold: best value for each quality score). The color encodes the classification ground-truth [93].

TABLE 2

Comparison of layout quality scores, averaged over all ensembles (bold: best values). The layouts induced by MT-PGA better preserve the global structure of the ensemble and still preserve well  $W_2^T$ .

Indicator	PCA	MDS ( $W_2^T$ ) [66]	t-SNE ( $W_2^T$ ) [117]	MT-VEC	MT-PGA
NMI	0.79	0.82	0.88	0.78	<b>0.94</b>
ARI	0.71	0.73	0.84	0.64	<b>0.90</b>
SIM	0.60	<b>0.85</b>	0.75	0.76	0.80

axes and coordinates). This factor is modest for a small example (Fig. 8), with few branches (135) and few members (16). Then, the overhead of the MT-PGA basis is non-negligible. In contrast, for a larger ensemble (Fig. 9), this overhead is negligible and high compression factors (30) can be achieved, while providing reconstructed merge trees which are highly similar visually and which are still viable for the applications.

Fig. 12 provides a visual comparison of the planar layouts generated by a selection of typical dimensionality reduction techniques. This includes the classical Euclidean PCA (in  $\mathbb{R}^{N_v}$ , where  $N_v$  is the number of vertices in  $\mathcal{M}$ ), MDS [66] and t-SNE [117] (both with  $W_2^T$ ). Certain approaches [5], [69], [101] applied PCA on top of *vectorizations* of topological descriptors (Sec. 1.1). A similar strategy can be considered in our case, by embedding each input BDT  $\mathcal{B}(f_j)$  in  $\mathbb{R}^{2 \times |\mathcal{B}^*|}$ , such that the  $i^{\text{th}}$  entry of this vector corresponds to the birth/death location of the  $i^{\text{th}}$  branch of  $\mathcal{B}(f_j)$  (i.e. the branch of  $\mathcal{B}(f_j)$  mapping to the  $i^{\text{th}}$  branch of  $\mathcal{B}^*$  by the optimal assignment of  $W_2^T$ , Eq. 1). PCA is then computed for this vectorization (column *MT-VEC*). As shown in Fig. 12, the layouts generated with MT-PGA nicely separates the ground-truth classes, while other techniques tend to artificially group them or even merge them. To further quantify this structure preservation, we run  $k$ -means in the 2D layouts and evaluate the quality of the resulting clustering (given the ground-truth [93]) with the normalized mutual information (NMI) and adjusted rand index (ARI). The MT-PGA layout is the only one in Fig. 12 which generates an exact clustering in both cases ( $NMI = ARI = 1$ ). Appendix D extends this analysis to all our test ensembles. Tab. 2 also extends this quantitative comparison to all our input ensembles and confirms the superiority, on average, of MT-PGA for the preservation of the clusters. Tab. 2 also includes a metric similarity indicator, *SIM*, which evaluates the preservation by a layout of the Wasserstein metric  $W_2^T$ , and whose expression is given in Appendix E. As expected, MDS maximizes this score by design, while MT-PGA produces the second best score. Overall, MT-PGA preserves well  $W_2^T$  as well as the global ensemble structure. In comparison to standard techniques (e.g. MDS or



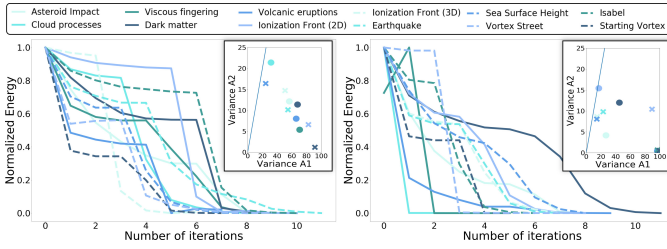


Fig. 13. Evolution of the normalized fitting energy, for PD (left) and MT (right) PGA. The scatter plots show the projected variance ( $\vec{A}_1$  VS  $\vec{A}_2$ ).

t-SNE), it also supports additional features, such as correlation views and interactive reconstructions (Figs. 1, 10 and 11).

Fig. 13 reports the evolution of the normalized fitting energy for PD and MT-PGA computations ( $d_{max} = 2$ ). Sudden energy drops can be observed with clear kinks in the curves, indicating the finalization of the first dimension (Sec. 4.1), and the switch to the second, which immediately improves the overall fit. The algorithm stops when the energy decreases by less than 1%. The curve flat tails indicate that this criterion is reasonable. Fig. 13 also reports scatter plots of the *projected variances*. For a given axis, it is the percentage of the variance of the projected trees (along the axis) over the global variance of the input BDTs (i.e. average of the squared  $W_2^T$  distances to the barycenter  $\mathcal{B}^*$ ). For all examples, the corresponding point is located below the diagonal (blue line): the projected variance is indeed larger for the first axis than for the second. This confirms the ability of our algorithm, similarly to the classical PCA, to identify in practice the most informative directions first. This is confirmed visually in our 2D layouts, where the first axis (blue) is always longer than the second (black). This is also confirmed in Appendix F, which provides a detailed variance analysis, for all test ensembles, in up to 10 dimensions.

### 6.3 Limitations

Our overall strategy (alternation of fitting and constraint enforcement) is similar at a high-level to previous work on the optimal transport of histograms [106]. Thus, it shares the same high-level limitations. The constraint enforcement (Sec. 4.3) induces, by construction, an energy increase. Then, it is possible that, at the next iteration, the fitting optimization (which is itself guaranteed to decrease the energy, Sec. 4.2) does not manage to compensate the above increase. This situation occurs during the first iteration for the *Viscous Fingering* ensemble (dark green line, Fig. 13, right), where the energy increased between two iterations. This temporary energy increase is the only one we observed in all our experiments. Moreover, it did not impact the rest of the algorithm, as it was immediately compensated at the next iteration (Fig. 13). Then, we believe that this theoretical limitation has a very limited impact in practice and that robust implementations can be obtained by stopping the algorithm in case of multiple, consecutive energy increases (which we have not observed). Another limitation involves the sampling of the geodesic axes ( $N_2$ , Sec. 4.2). When  $N_2$  is too low, BDTs which are close in  $\mathbb{B}$  may project to the same points on the geodesic axes, possibly resulting overall in collocated points in the MT-PGA basis. This can be easily resolved by increasing  $N_2$ , at the cost of increased computation times. Finally, similarly to barycenter optimization [92], [116], [118], the overall fitting energy (Eq. 8) is non-convex and can in principle admit multiple

local minimizers. However, our experiments indicate that the axes returned by our approach are relevant, as the projected variance does decrease for increasing dimensions (Sec. 6.2).

## 7 CONCLUSION

In this paper, we presented a computational framework for the Principal Geodesic Analysis of merge trees (MT-PGA), with applications to data reduction and dimensionality reduction. In particular, the visualizations derived from our core contribution (Figs. 1, 10, 11) enable the interactive, visual inspection of the variability in the ensemble, both at a global level (with our two-dimensional layouts) and at a feature level (with our persistence correlation views). Our framework trivially extends to extremum persistence diagrams and our algorithm enables in both cases PGA basis computations within minutes for real-life ensembles.

A natural direction for future work is the extension of our framework to other topological data representations, such as Reeb graphs or Morse-Smale complexes. However, as detailed in Sec. 3, this requires the definition of several low-level geometrical tools, such as routines for geodesic or barycenter computation, which is still an open research problem for the above topological descriptors. By adapting Principal Component Analysis to merge trees, we believe our work is an important practical step towards the definition of a larger statistical framework on the space of merge trees. In the future, we will continue our investigation of the adaptation of classical statistical tools to ensembles of topological objects, as we believe it can become a key solution in the long term for the advanced analysis of large-scale ensembles.

## REFERENCES

- [1] ISO/IEC Guide 98-3:2008 uncertainty of measurement - part 3: Guide to the expression of uncertainty in measurement (GUM). 2008.
- [2] A. Acharya and V. Natarajan. A parallel and memory efficient algorithm for constructing the contour tree. In *IEEE PV*, 2015.
- [3] H. Adams, S. Chepushtanova, T. Emerson, E. Hanson, M. Kirby, F. Motta, R. Neville, C. Peterson, P. Shipman, and L. Ziegelmeier. Persistence Images: A Stable Vector Representation of Persistent Homology. *Journal of Machine Learning Research*, 2017.
- [4] K. Anderson, J. Anderson, S. Palande, and B. Wang. Topological data analysis of functional MRI connectivity in time and space domains. In *MICCAI Workshop on Connectomics in NeuroImaging*, 2018.
- [5] R. Anirudh, V. Venkataraman, K. N. Ramamurthy, and P. K. Turaga. A Riemannian Framework for Statistical Analysis of Topological Persistence Diagrams. In *IEEE CVPR Workshops*, 2016.
- [6] T. Athawale and A. Entezari. Uncertainty quantification in linear interpolation for isosurface extraction. *IEEE TVCG*, 2013.
- [7] T. Athawale and C. R. Johnson. Probabilistic Asymptotic Decider for Topological Ambiguity Resolution in Level-Set Extraction for Uncertain 2D Data. *IEEE TVCG*, 2019.
- [8] T. Athawale, E. Sakhaee, and A. Entezari. Isosurface visualization of data with nonparametric models for uncertainty. *IEEE TVCG*, 2016.
- [9] T. M. Athawale, D. Maljovec, C. R. Johnson, V. Pascucci, and B. Wang. Uncertainty Visualization of 2D Morse Complex Ensembles Using Statistical Summary Maps. *CoRR*, abs/1912.06341, 2019.
- [10] U. Ayachit, A. C. Bauer, B. Geveci, P. O’Leary, K. Moreland, N. Fabian, and J. Mauldin. ParaView Catalyst: Enabling In Situ Data Analysis and Visualization. In *ISAV*, 2015.
- [11] T. F. Banchoff. Critical points and curvature for embedded polyhedral surfaces. *The American Mathematical Monthly*, 1970.
- [12] U. Bauer, X. Ge, and Y. Wang. Measuring distance between Reeb graphs. In *SoCG*, 2014.
- [13] U. Bauer, M. Kerber, and J. Reininghaus. Distributed computation of persistent homology. In *Algorithm Engineering and Experiments*, 2014.
- [14] K. Beketayev, D. Yeliussizov, D. Morozov, G. H. Weber, and B. Hamann. Measuring the distance between merge trees. In *TopoInVis*. 2014.
- [15] D. P. Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 1981.

- [16] H. Bhatia, A. G. Gyulassy, V. Lordi, J. E. Pask, V. Pascucci, and P.-T. Bremer. Topoms: Comprehensive topological exploration for molecular and condensed-matter systems. *J. of Computational Chemistry*, 2018.
- [17] H. Bhatia, S. Jadhav, P. Bremer, G. Chen, J. A. Levine, L. G. Nonato, and V. Pascucci. Flow visualization with quantified spatial and temporal errors using edge maps. *IEEE TVCG*, 2012.
- [18] S. Biasotti, D. Giorgio, M. Spagnuolo, and B. Falcidieno. Reeb graphs for shape analysis and applications. *TCS*, 2008.
- [19] T. Bin Masood, J. Budin, M. Falk, G. Favelier, C. Garth, C. Gueunet, P. Guillou, L. Hofmann, P. Hristov, A. Kamakshidasan, C. Kappe, P. Klacansky, P. Laurin, J. Levine, J. Lukaszczuk, D. Sakurai, M. Soler, P. Steneteg, J. Tierny, W. Usher, J. Vidal, and M. Wozniak. An Overview of the Topology Toolkit. In *TopolnVis*, 2019.
- [20] A. Bock, H. Doraiswamy, A. Summers, and C. T. Silva. TopoAngler: Interactive Topology-Based Extraction of Fishes. *IEEE TVCG*, 2018.
- [21] B. Bollen, E. Chambers, J. A. Levine, and E. Munch. Reeb Graph Metrics from the Ground Up. *CoRR*, 2021.
- [22] G. Bonneau, H. Hege, C. Johnson, M. Oliveira, K. Potter, P. Rheingans, and T. Schultz. Overview and state-of-the-art of uncertainty visualization. *Mathematics and Visualization*, 37:3–27, 2014.
- [23] P. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci. A Multi-Resolution Data Structure for 2-Dimensional Morse Functions. In *Proc. of IEEE VIS*, 2003.
- [24] P. Bremer, G. Weber, J. Tierny, V. Pascucci, M. Day, and J. Bell. Interactive exploration and analysis of large scale simulations using topology-based data segmentation. *IEEE TVCG*, 2011.
- [25] N. Brown, R. Nash, P. Poletti, G. Guzzetta, M. Manica, A. Zardini, M. Flatken, J. Vidal, C. Gueunet, E. Belikov, J. Tierny, A. Podobas, W. D. Chien, S. Markidis, and A. Gerndt. Utilising urgent computing to tackle the spread of mosquito-borne diseases. In *IEEE/ACM UrgentHPC@SC*, 2021.
- [26] P. Bubenik. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 2015.
- [27] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. In *Symp. on Dis. Alg.*, 2000.
- [28] H. Carr, G. Weber, C. Sewell, and J. Ahrens. Parallel peak pruning for scalable SMP contour tree computation. In *IEEE Lдав*, 2016.
- [29] H. A. Carr, J. Snoeyink, and M. van de Panne. Simplifying Flexible Isosurfaces Using Local Geometric Measures. In *IEEE VIS*, 2004.
- [30] E. Cazelles, V. Seguy, J. Bigot, M. Cuturi, and N. Papadakis. Geodesic PCA versus Log-PCA of Histograms in the Wasserstein Space. *SIAM J. Sci. Comput.*, 2018.
- [31] W. Cheney and D. Kincaid. *Linear Algebra: Theory and Applications*. Jones and Bartlett Publishers, 2009.
- [32] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, 2013.
- [33] M. Cuturi and A. Doucet. Fast computation of wasserstein barycenters. In *ICML*, 2014.
- [34] L. De Floriani, U. Fugacci, F. Iuricich, and P. Magillo. Morse complexes for shape segmentation and homological analysis: discrete models and algorithms. *CGF*, 2015.
- [35] P. Diggle, P. Heagerty, K.-Y. Liang, and S. Zeger. *The Analysis of Longitudinal Data*. Oxford University Press, 2002.
- [36] H. Doraiswamy and V. Natarajan. Computing reeb graphs as a union of contour trees. *IEEE TVCG*, 2013.
- [37] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2009.
- [38] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-smale complexes for piecewise linear 3-manifolds. In *SoCG*, 2003.
- [39] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical morse complexes for piecewise linear 2-manifolds. In *SoCG*, 2001.
- [40] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *DCG*, 2002.
- [41] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM ToG*, 1990.
- [42] G. Favelier, N. Faraj, B. Summa, and J. Tierny. Persistence Atlas for Critical Point Variability in Ensembles. *IEEE TVCG*, 2018.
- [43] G. Favelier, C. Gueunet, and J. Tierny. Visualizing ensembles of viscous fingers. In *IEEE SciVis Contest*, 2016.
- [44] F. Ferstl, K. Bürger, and R. Westermann. Streamline variability plots for characterizing the uncertainty in vector field ensembles. *IEEE TVCG*, 2016.
- [45] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. Visual analysis of spatial variability and global correlations in ensembles of iso-contours. *CGF*, 2016.
- [46] P. T. Fletcher, C. Lu, S. M. Pizer, and S. C. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE TMI*, 2004.
- [47] R. Forman. A User's Guide to Discrete Morse Theory. *AM*, 1998.
- [48] E. Gasparovic, E. Munch, S. Oudot, K. Turner, B. Wang, and Y. Wang. Intrinsic interleaving distance for merge trees. *CoRR*, 1908.00063, 2019.
- [49] A. N. Gorban, B. Kegl, D. C. Wunsch, and A. Zinovyev. *Principal Manifolds for Data Visualization and Dimension Reduction*. Springer, 2008.
- [50] A. N. Gorban and A. Y. Zinovyev. Principal Graphs and Manifolds. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques*. 2009.
- [51] D. Guenther, R. Alvarez-Boto, J. Contreras-Garcia, J.-P. Piquemal, and J. Tierny. Characterizing molecular interactions in chemical systems. *IEEE TVCG*, 2014.
- [52] C. Gueunet, P. Fortin, J. Jomier, and J. Tierny. Task-Based Augmented Contour Trees with Fibonacci Heaps. *IEEE TPDS*, 2019.
- [53] C. Gueunet, P. Fortin, J. Jomier, and J. Tierny. Task-based Augmented Reeb Graphs with Dynamic ST-Trees. In *EGPGV*, 2019.
- [54] D. Günther, J. Salmon, and J. Tierny. Mandatory critical points of 2D uncertain scalar fields. *CGF*, 2014.
- [55] A. Gyulassy, P. Bremer, R. Grout, H. Kolla, J. Chen, and V. Pascucci. Stability of dissipation elements: A case study in combustion. *CGF*, 2014.
- [56] A. Gyulassy, P. Bremer, and V. Pascucci. Shared-Memory Parallel Computation of Morse-Smale Complexes with Improved Accuracy. *IEEE TVCG*, 2018.
- [57] A. Gyulassy, M. A. Duchaineau, V. Natarajan, V. Pascucci, E. Bringa, A. Higginbotham, and B. Hamann. Topologically clean distance fields. *IEEE TVCG*, 2007.
- [58] A. Gyulassy, A. Knoll, K. Lau, B. Wang, P. Bremer, M. Papka, L. A. Curtiss, and V. Pascucci. Interstitial and interlayer ion diffusion geometry extraction in graphitic nanosphere battery materials. *IEEE TVCG*, 2015.
- [59] C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. De Floriani, G. Scheuermann, H. Hagen, and C. Garth. A survey of topology-based methods in visualization. *CGF*, 2016.
- [60] M. Hilaga, Y. Shinagawa, T. Komura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *ACM SIGGRAPH*, 2001.
- [61] M. Hummel, H. Obermaier, C. Garth, and K. I. Joy. Comparative visual analysis of lagrangian transport in CFD ensembles. *IEEE TVCG*, 2013.
- [62] C. R. Johnson and A. R. Sanderson. A next step: Visualizing errors and uncertainty. *IEEE Computer Graphics and Applications*, 2003.
- [63] L. Kantorovich. On the translocation of masses. *AS USSR*, 1942.
- [64] J. Kasten, J. Reininghaus, I. Hotz, and H. Hege. Two-dimensional time-dependent vortex regions based on the acceleration magnitude. *IEEE TVCG*, 2011.
- [65] M. Kerber, D. Morozov, and A. Nigmatov. Geometry helps to compare persistence diagrams. *ACM J. of Experimental Algorithmics*, 2016.
- [66] J. B. Kruskal and M. Wish. Multidimensional Scaling. In *SUPS*, 1978.
- [67] T. Lacombe, M. Cuturi, and S. Oudot. Large Scale computation of Means and Clusters for Persistence Diagrams using Optimal Transport. In *NIPS*, 2018.
- [68] D. E. Laney, P. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE TVCG*, 2006.
- [69] M. Li, S. Palande, and B. Wang. Sketching merge trees. *CoRR*, 2021.
- [70] T. Liebmann and G. Scheuermann. Critical points of gaussian-distributed scalar fields on simplicial grids. *CGF*, 2016.
- [71] A. P. Lohfink, F. Wetzels, J. Lukaszczuk, G. H. Weber, and C. Garth. Fuzzy contour trees: Alignment and joint layout of multiple contour trees. *CGF*, 2020.
- [72] S. Maadasamy, H. Doraiswamy, and V. Natarajan. A hybrid parallel algorithm for computing and tracking level set topology. In *HiPC*, 2012.
- [73] A. Maceachren, A. Robinson, S. Hopper, S. Gardner, R. Murray, M. Gahagan, and E. Hetzler. Visualizing geospatial information uncertainty: What we know and what we need to know. *CGIS*, 2005.
- [74] D. Maljovec, B. Wang, P. Rosen, A. Alfonsi, G. Pastore, C. Rabiti, and V. Pascucci. Topology-inspired partition-based sensitivity analysis and visualization of nuclear simulations. In *IEEE PV*, 2016.
- [75] M. Mirzargar, R. Whitaker, and R. Kirby. Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE TVCG*, 2014.
- [76] G. Monge. Mémoire sur la théorie des déblais et des remblais. *Académie Royale des Sciences de Paris*, 1781.
- [77] D. Morozov, K. Beketayev, and G. H. Weber. Interleaving distance between merge trees. In *TopolnVis*. 2014.

- [78] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 1957.
- [79] M. Olejniczak, A. S. P. Gomes, and J. Tierny. A Topological Data Analysis Perspective on Non-Covalent Interactions in Relativistic Calculations. *International Journal of Quantum Chemistry*, 2019.
- [80] Organizers. The IEEE SciVis Contest. <http://sciviscontest.ieeevis.org/>, 2004.
- [81] M. Otto, T. Germer, H.-C. Hege, and H. Theisel. Uncertain 2D vector field topology. 2010.
- [82] M. Otto, T. Germer, and H. Theisel. Uncertain topology of 3D vector fields. *IEEE PV*, 2011.
- [83] A. T. Pang, C. M. Wittenbrink, and S. K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 1997.
- [84] S. Parsa. A deterministic  $o(m \log m)$  time algorithm for the reeb graph. In *SoCG*, 2012.
- [85] V. Pascucci, K. Cole-McLaughlin, and G. Scorzelli. Multi-resolution computation and presentation of contour trees. In *IASTED*, 2004.
- [86] V. Pascucci, G. Scorzelli, P. T. Bremer, and A. Mascarenhas. Robust on-line computation of Reeb graphs: simplicity and speed. *ACM ToG*, 2007.
- [87] K. Pearson. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, 1901.
- [88] C. Petz, K. Pöthkow, and H.-C. Hege. Probabilistic local features in uncertain vector fields with spatial correlation. *CGF*, 2012.
- [89] T. Pfaffelmoser, M. Mihai, and R. Westermann. Visualizing the variability of gradients in uncertain 2D scalar fields. *IEEE TVCG*, 2013.
- [90] T. Pfaffelmoser, M. Reitingner, and R. Westermann. Visualizing the positional and geometrical variability of isosurfaces in uncertain scalar fields. *CGF*, 2011.
- [91] T. Pfaffelmoser and R. Westermann. Visualization of global correlation structures in uncertain 2D scalar fields. *CGF*, 2012.
- [92] M. Pont, J. Vidal, J. Delon, and J. Tierny. Wasserstein Distances, Geodesics and Barycenters of Merge Trees. *IEEE TVCG*, 2021.
- [93] M. Pont, J. Vidal, J. Delon, and J. Tierny. Wasserstein Distances, Geodesics and Barycenters of Merge Trees – Ensemble Benchmark. <https://github.com/MatPont/WassersteinMergeTreesData>, 2021.
- [94] K. Pöthkow and H.-C. Hege. Positional uncertainty of isocontours: Condition analysis and probabilistic measures. *IEEE TVCG*, 2011.
- [95] K. Pöthkow and H.-C. Hege. Nonparametric models for uncertainty visualization. *CGF*, 2013.
- [96] K. Pöthkow, C. Petz, and H.-C. Hege. Approximate level-crossing probabilities for interactive visualization of uncertain isocontours. *Int. J. Uncert. Quantif.*, 2013.
- [97] K. Pöthkow, B. Weber, and H.-C. Hege. Probabilistic marching cubes. In *CGF*, 2011.
- [98] K. Potter, S. Gerber, and E. W. Anderson. Visualization of uncertainty without a mean. *IEEE Computer Graphics and Applications*, 2013.
- [99] K. Potter, A. Wilson, P. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. R. Johnson. Ensemble-vis: A framework for the statistical visualization of ensemble data. In *2009 IEEE ICDM*, 2009.
- [100] J. C. Potter, K. Rosen P. From quantification to visualization: A taxonomy of uncertainty visualization approaches. *IFIP AICT*, 2012.
- [101] V. Robins and K. Turner. Principal Component Analysis of Persistent Homology Rank Functions with case studies of Spatial Point Patterns, Sphere Packing and Colloids. *Physica D: Nonlinear Phenomena*, 2016.
- [102] V. Robins, P. J. Wood, and A. P. Sheppard. Theory and Algorithms for Constructing Discrete Morse Complexes from Grayscale Digital Images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011.
- [103] H. Saikia, H. Seidel, and T. Weinkauff. Extended branch decomposition graphs: Structural comparison of scalar data. *CGF*, 2014.
- [104] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. Moorhead. Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *IEEE TVCG*, 2010.
- [105] S. Schlegel, N. Korn, and G. Scheuermann. On the interpolation of data with normally distributed uncertainty for visualization. *IEEE TVCG*, 2012.
- [106] V. Seguy and M. Cuturi. Principal Geodesic Analysis for Probability Measures under the Optimal Transport Metric. In *NeurIPS*, 2015.
- [107] N. Shivashankar and V. Natarajan. Parallel Computation of 3D Morse-Smale Complexes. *CGF*, 2012.
- [108] N. Shivashankar, P. Pranav, V. Natarajan, R. van de Weygaert, E. P. Bos, and S. Rieder. Felix: A topology based framework for visual exploration of cosmic filaments. *IEEE TVCG*, 2016.
- [109] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *American Mathematical Monthly*, 1967.
- [110] T. Sousbie. The persistent cosmic web and its filamentary structure: Theory and implementations. *Royal Astronomical Society*, 2011.
- [111] R. Sridharamurthy, T. B. Masood, A. Kamakshidasan, and V. Natarajan. Edit distance between merge trees. *IEEE TVCG*, 2020.
- [112] A. Szymczak. Hierarchy of stable Morse decompositions. *IEEE TVCG*, 2013.
- [113] S. Tarasov and M. Vyalı. Construction of contour trees in 3d in  $o(n \log n)$  steps. In *SoCG*, 1998.
- [114] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. The Topology ToolKit. *IEEE TVCG*, 2017. <https://topology-tool-kit.github.io/>.
- [115] J. Tierny, A. Gyulassy, E. Simon, and V. Pascucci. Loop surgery for volumetric meshes: Reeb graphs reduced to contour trees. *IEEE TVCG*, 2009.
- [116] K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer. Fréchet Means for Distributions of Persistence Diagrams. *DCG*, 2014.
- [117] L. P. van der Maaten and G. Hinton. Visualizing Data Using t-SNE. *JMLR*, 2008.
- [118] J. Vidal, J. Budin, and J. Tierny. Progressive Wasserstein Barycenters of Persistence Diagrams. *IEEE TVCG*, 2019.
- [119] R. T. Whitaker, M. Mirzargar, and R. M. Kirby. Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *IEEE TVCG*, 2013.
- [120] D. P. Woodruff. *Sketching as a Tool for Numerical Linear Algebra*. Now Publishers, 2014.
- [121] L. Yan, T. B. Masood, R. Sridharamurthy, F. Rasheed, V. Natarajan, I. Hotz, and B. Wang. Scalar field comparison with topological descriptors: Properties and applications for scientific visualization. *CGF*, 2021.
- [122] L. Yan, Y. Wang, E. Munch, E. Gasparovic, and B. Wang. A structural average of labeled merge trees for uncertainty visualization. *IEEE TVCG*, 2019.
- [123] K. Zhang. A Constrained Edit Distance Between Unordered Labeled Trees. *Algorithmica*, 1996.



**Mathieu Pont** is a Ph.D. student at Sorbonne Université. He received a M.S. degree in Computer Science from Paris Descartes University in 2020. He is an active contributor to the Topology ToolKit (TTK), an open source library for topological data analysis. His notable contributions to TTK include distances, geodesics and barycenters of merge trees, for feature tracking and ensemble clustering.



**Jules Vidal** is a post-doctoral researcher, currently at Sorbonne Université, from where he received the Ph.D. degree in 2021. He received the engineering degree in 2018 from ENSTA Paris. He is an active contributor to the Topology ToolKit (TTK), an open source library for topological data analysis. His notable contributions to TTK include the efficient and progressive approximation of distances, barycenters and clusterings of persistence diagrams.



**Julien Tierny** received the Ph.D. degree in Computer Science from the University of Lille in 2008. He is currently a CNRS research director, affiliated with Sorbonne University. Prior to his CNRS tenure, he held a Fulbright fellowship (U.S. Department of State) and was a post-doctoral researcher at the Scientific Computing and Imaging Institute at the University of Utah. His research expertise lies in topological methods for data analysis and visualization. He is the founder and lead developer of the Topology ToolKit (TTK), an open source library for topological data analysis.



## APPENDIX

### APPENDIX A

#### GRADIENT OF THE FITTING ENERGY

Given the Wasserstein barycenter  $\mathcal{B}^*$  of the input set of BDTs  $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$  as well as an initialization of the  $d'$ -th geodesic axis of  $B_{\mathbb{B}}$ , noted  $\overrightarrow{\mathcal{A}}_{d'}$ , we want to update  $\overrightarrow{\mathcal{A}}_{d'}$  in order to decrease the fitting energy associated to  $B_{\mathbb{B}}$  (Sec. 3, main manuscript):

$$E_{W_2^T}(B_{\mathbb{B}}) = \sum_{j=1}^N W_2^T \left( \mathcal{B}(f_j), \mathcal{B}^* + \sum_{i=1}^{d'} \overrightarrow{\mathcal{A}}_i(\widehat{\mathcal{B}}_i(f_j)) \right)^2. \quad (10)$$

Similarly to previous optimization strategies for the Fréchet energy [92], [116], [118], our approach consists in alternating phases of (i) *Assignments* (between the input BDTs and their projections along translations of  $\overrightarrow{\mathcal{A}}_{d'}$ , cf. Eq. 10) and (ii) *Updates* (of the axis  $\overrightarrow{\mathcal{A}}_{d'}$ ).

Thus, at the *Update* phase (ii), the assignment  $\phi_{d'}^j$  between each input BDT  $\mathcal{B}(f_j)$  and its projection  $\widehat{\mathcal{B}}_{d'}(f_j)$  on the translated axis  $\overrightarrow{\mathcal{A}}_{d'}(\widehat{\mathcal{B}}_{d'-1}(f_j))$  is constant. It follows that each branch  $b^* \in \mathcal{B}^*$  can be considered independently for the minimization of Eq. 10. In particular, let  $(b^* + \sum_{i=1}^{d'} (1 - \alpha_i^j) \overrightarrow{g}_i + \alpha_i^j \overrightarrow{g}'_i)$  be the branch assigned to  $b^*$  in  $\widehat{\mathcal{B}}_{d'}(f_j)$ , with  $\overrightarrow{g}_i$  and  $\overrightarrow{g}'_i$  being two 2D vectors in the birth/death space (i.e. the entries of  $\overrightarrow{\mathcal{G}}_i$  and  $\overrightarrow{\mathcal{G}}'_i$  corresponding to the branch  $b^*$  of  $\mathcal{B}^*$ ). Then, the individual fitting energy associated to  $b^*$ , i.e. its contribution to Eq. 10, can be expressed as:

$$E_{b^*}(\overrightarrow{g}_{d'}, \overrightarrow{g}'_{d'}) = \sum_{j=1}^N d_2 \left( b_j, b^* + \sum_{i=1}^{d'} (1 - \alpha_i^j) \overrightarrow{g}_i + \alpha_i^j \overrightarrow{g}'_i \right)^2, \quad (11)$$

where  $b_j$  stands for the branch in  $\mathcal{B}(f_j)$  assigned to  $b^*$  and  $d_2$  stands for the  $L_2$  norm in the 2D birth/death space (i.e. the ground distance involved in  $W_2^T$ , see Sec. 2.4, main manuscript). Note that the usage of the  $L_2$  norm implies that  $E_{b^*}$  is convex. Our goal at this stage is to find the two vectors  $\overrightarrow{g}_{d'}$  and  $\overrightarrow{g}'_{d'}$  which minimize Eq. 11.

Eq. 11 can be further detailed as follows, where  $g_{ix}$  and  $g_{iy}$  stand for the X-Y coordinates of the vector  $\overrightarrow{g}_i$  in the birth/death plane:

$$E_{b^*}(\overrightarrow{g}_{d'}, \overrightarrow{g}'_{d'}) = \sum_{j=1}^N \left( b_{jx} - (b_x^* + \sum_{i=1}^{d'} (1 - \alpha_i^j) \times g_{ix} + \alpha_i^j \times g'_{ix}) \right)^2 + \left( b_{jy} - (b_y^* + \sum_{i=1}^{d'} (1 - \alpha_i^j) \times g_{iy} + \alpha_i^j \times g'_{iy}) \right)^2. \quad (12)$$

In the following, we derive the gradient of the above convex energy. In particular, we detail the derivation for the X coordinate only (the derivation along the Y coordinate being identical):

$$\frac{\partial E_{b^*}(\overrightarrow{g}_{d'}, \overrightarrow{g}'_{d'})}{\partial g_{d'x}} = 2 \sum_{j=1}^N (1 - \alpha_{d'}^j) \left( b_{jx} - (b_x^* + \sum_{i=1}^{d'} (1 - \alpha_i^j) \times g_{ix} + \alpha_i^j \times g'_{ix}) \right) + \sum_{i=1}^{d'} (1 - \alpha_i^j) \times g_{ix} + \alpha_i^j \times g'_{ix}$$

$$\frac{\partial E_{b^*}(\overrightarrow{g}_{d'}, \overrightarrow{g}'_{d'})}{\partial g'_{d'x}} = 2 \sum_{j=1}^N \alpha_{d'}^j \left( b_{jx} - (b_x^* + \sum_{i=1}^{d'} (1 - \alpha_i^j) \times g_{ix} + \alpha_i^j \times g'_{ix}) \right) + \sum_{i=1}^{d'} \alpha_i^j \times g'_{ix}$$

To minimize Eq. 12, we aim to find the values of  $g_{d'x}$  and  $g'_{d'x}$  for which the above partial derivatives equal zero. This yields the following linear system of two equations (with two unknowns,  $g_{d'x}$  and  $g'_{d'x}$ ):

$$\begin{cases} \sum_{j=1}^N (1 - \alpha_{d'}^j) \left( b_{jx} - (b_x^* + \sum_{i=1}^{d'} (1 - \alpha_i^j) \times g_{ix} + \alpha_i^j \times g'_{ix}) \right) = 0 \\ \sum_{j=1}^N \alpha_{d'}^j \left( b_{jx} - (b_x^* + \sum_{i=1}^{d'} (1 - \alpha_i^j) \times g_{ix} + \alpha_i^j \times g'_{ix}) \right) = 0. \end{cases} \quad (13)$$

Given the above system, we aim next at expressing  $g_{d'x}$  as a function of  $g'_{d'x}$ . To simplify notations, we introduce the term  $b_{d'x}^*$  as follows:

$$b_{d'x}^* := b_x^* + \sum_{i=1}^{d'-1} (1 - \alpha_i^j) \times g_{ix} + \alpha_i^j \times g'_{ix}.$$

Then, the first line of Eq. 13 can be re-written as:

$$\sum_{j=1}^N (1 - \alpha_{d'}^j) \left( b_{jx} - (b_{d'x}^* + (1 - \alpha_{d'}^j) \times g_{d'x} + \alpha_{d'}^j \times g'_{d'x}) \right) = 0.$$

Then, it follows that:

$$g_{d'x} = \frac{\sum_{j=1}^N (1 - \alpha_{d'}^j) \left( b_{jx} - (b_{d'x}^* + \alpha_{d'}^j \times g'_{d'x}) \right)}{\sum_{j=1}^N (1 - \alpha_{d'}^j)^2}. \quad (14)$$

Now, we apply the same reasoning with the second line of Eq. 13, yielding the following expression of  $g'_{d'x}$ :

$$g'_{d'x} = \frac{\sum_{j=1}^N \alpha_{d'}^j \left( b_{jx} - (b_{d'x}^* + (1 - \alpha_{d'}^j) \times g_{d'x}) \right)}{\sum_{j=1}^N (\alpha_{d'}^j)^2}. \quad (15)$$

At this stage, one can notice that the expression of  $g'_{d'x}$  is itself a function of  $g_{d'x}$ . Thus, we insert the expression of  $g'_{d'x}$  (Eq. 15) into that of  $g_{d'x}$  (Eq. 14), which results eventually in the following expression (we omit the detailed, intermediate steps):

$$g_{d'x} = \frac{\sum_{j=1}^N (1 - \alpha_{d'}^j) \left( b_{jx} - b_{d'x}^* - \alpha_{d'}^j \frac{\sum_{k=1}^N \alpha_{d'}^k (b_{kx} - b_{d'x}^*)}{\sum_{k=1}^N (\alpha_{d'}^k)^2} \right)}{\sum_{j=1}^N (1 - \alpha_{d'}^j)^2 + \sum_{j=1}^N (1 - \alpha_{d'}^j) \alpha_{d'}^j \frac{\sum_{k=1}^N \alpha_{d'}^k (- (1 - \alpha_{d'}^k))}{\sum_{k=1}^N (\alpha_{d'}^k)^2}}. \quad (16)$$

Finally, we can insert the expression of  $g_{d'x}$  (Eq. 14) into the original expression of  $g'_{d'x}$  (Eq. 15), resulting in the following expression (again, we omit the detailed, intermediate steps):

$$g'_{d'x} = \frac{\sum_{j=1}^N \alpha_{d'}^j \left( b_{jx} - b_{d'x}^* - (1 - \alpha_{d'}^j) \frac{\sum_{k=1}^N (1 - \alpha_{d'}^k) (b_{kx} - b_{d'x}^*)}{\sum_{k=1}^N (1 - \alpha_{d'}^k)^2} \right)}{\sum_{j=1}^N (\alpha_{d'}^j)^2 + \sum_{j=1}^N \alpha_{d'}^j (1 - \alpha_{d'}^j) \frac{\sum_{k=1}^N (1 - \alpha_{d'}^k) (- \alpha_{d'}^k)}{\sum_{k=1}^N (1 - \alpha_{d'}^k)^2}}. \quad (17)$$

Overall, Eq. 16 and Eq. 17 provide the expression of the X-coordinate of the vectors  $\overrightarrow{g}_{d'}$  and  $\overrightarrow{g}'_{d'}$  which minimize the individual fitting energy (Eq. 12). The same reasoning (not detailed here) can be applied identically to retrieve the Y-coordinate of  $\overrightarrow{g}_{d'}$  and  $\overrightarrow{g}'_{d'}$ . Then, the entire vectors  $\overrightarrow{\mathcal{G}}_{d'}$  and  $\overrightarrow{\mathcal{G}}'_{d'}$  (defining  $\overrightarrow{\mathcal{A}}_{d'}(\widehat{\mathcal{B}}_{d'}(f_j))$ ) which minimize Eq. 10 under the current assignments can be updated similarly, by iterating the above computation for all the branches  $b^*$  of  $\mathcal{B}^*$ .

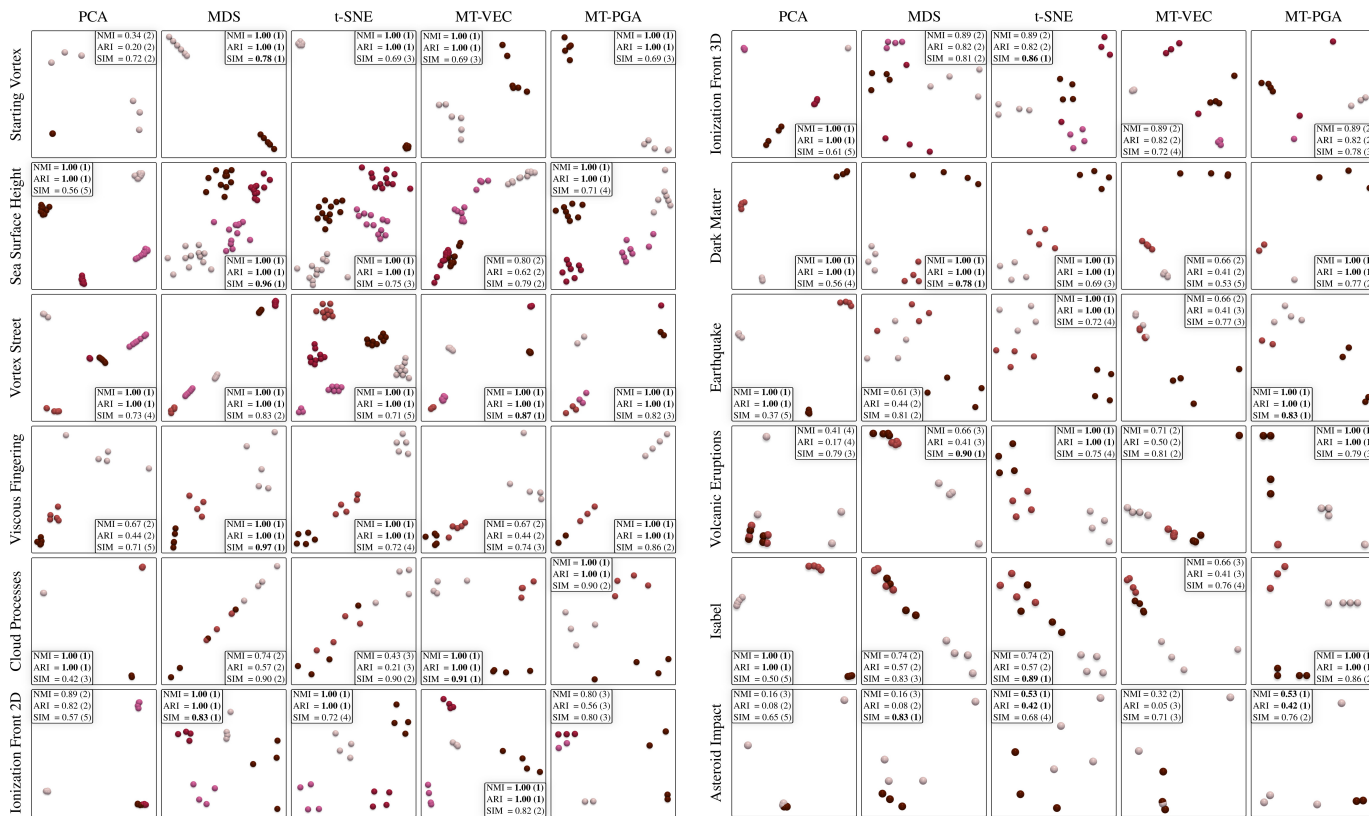


Fig. 14. Comparison of planar layouts for typical dimensionality reduction techniques on all our test ensembles. The color encodes the classification ground-truth [92]. For each quality score, the best value appears bold and the rank of the score among all methods is in parenthesis.

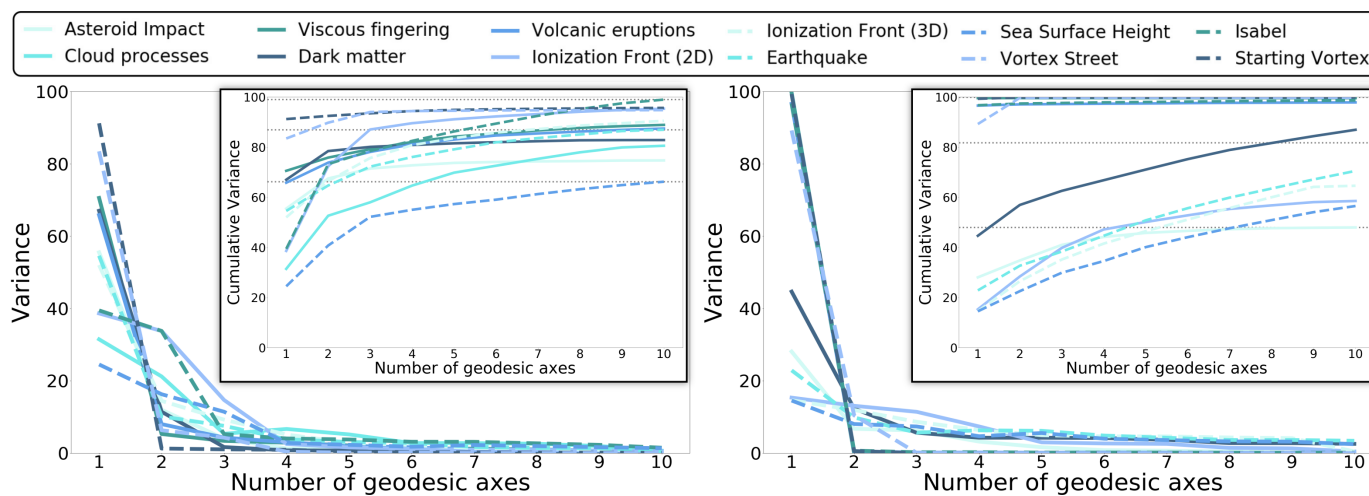


Fig. 15. Evolution of the projected variance (and cumulative variance, inset) with the number of geodesic axes for PD-PGA (left) and MT-PGA (right).

TABLE 3

Compression factor and average relative reconstruction error of our algorithm for PD-PGA and MT-PGA (for  $d_{max} = 3$  and  $N_1 = 20$ ).

Dataset	$N$	$ \mathcal{B} $	PD-PGA		MT-PGA	
			Factor	Error	Factor	Error
Asteroid Impact (3D)	7	1,295	2.97	0.07	4.84	0.22
Cloud processes (2D)	12	1,209	5.94	0.19	7.39	0.01
Viscous fingering (3D)	15	118	2.23	0.13	4.71	0.02
Dark matter (3D)	40	2,592	10.00	0.04	19.27	0.04
Volcanic eruptions (2D)	12	811	9.99	0.12	4.83	0.04
Ionization front (2D)	16	135	2.56	0.14	5.12	0.40
Ionization front (3D)	16	763	3.27	0.17	4.85	0.46
Earthquake (3D)	12	1,203	1.42	0.18	2.19	0.33
Isabel (3D)	12	1,338	5.49	0.27	9.25	0.05
Starting Vortex (2D)	12	124	1.76	0.07	4.42	0.01
Sea Surface Height (2D)	48	1,787	19.59	0.18	9.48	0.48
Vortex Street (2D)	45	23	1.86	0.04	11.84	0.02

## APPENDIX B DATA REDUCTION

Tab. 3 reports the compression factors and average relative reconstruction error for all our test ensembles. In particular, for each input BDT  $\mathcal{B}(f_i)$ , we compute its reconstruction error via the  $W_2^T$  distance to its reconstruction  $\hat{\mathcal{B}}(f_i)$ . In order to become comparable across ensembles, this distance is then divided by the maximum  $W_2^T$  distance observed among two input BDTs in the ensemble. Finally, this relative reconstruction error is averaged over all the BDTs of the input ensemble.

This table shows that our framework results in significant compression factors (9 or above) for the largest ensembles, i.e. the ensembles counting the most members and for which the BDTs are the largest (i.e. Dark matter, Sea Surface Height). On the contrary, modest compression factors tend to be obtained for the smallest ensembles, counting few members and few features. The average relative reconstruction error seems acceptable overall: 0.13 and 0.17 on average for PD-PGA and MT-PGA respectively.

Finally, note that for each ensemble, the merge tree based clustering [92] computed from the input BDTs is strictly identical to the clustering computed from the reconstructed BDTs. This confirms the viability of our reconstructed BDTs, and their usability for typical visualization and analysis tasks.

## APPENDIX C PERSISTENCE CORRELATION

This section details the computation of the correlation between the persistence of the  $i^{\text{th}}$  feature of an input BDT  $\mathcal{B}(f_j)$  and its coordinate  $\alpha_k^j$  along the geodesic axis  $\mathcal{A}_k$ . Let  $\mathbb{P}$  be an  $(N_1 \times N)$ -matrix, such that the entry  $\mathbb{P}(i, j)$  denotes the topological persistence, in the  $j^{\text{th}}$  input BDT, of the branch  $b_j \in \mathcal{B}(f_j)$  mapped to the  $i^{\text{th}}$  most persistent branch of  $\mathcal{B}^*$  given the optimal assignment induced by  $W_2^T$  (Eq. 1 of the main manuscript). Next, let  $\mathbb{A}$  be a  $(d_{max} \times N)$ -matrix, such that then entry  $\mathbb{A}(k, j)$  denotes the coordinate  $\alpha_k^j$  of the BDT  $\mathcal{B}(f_j)$  along the axis  $\mathcal{A}_k$ .

In the following, we aim at assessing how much the persistence of the branches in  $\mathcal{B}(f_j)$  is correlated with the coordinate  $\alpha_k^j$ . Let  $\rho_{p_i, \alpha_k}$  be the correlation between the persistence  $p_i$  ( $i^{\text{th}}$  line of  $\mathbb{P}$ ) and the coordinate  $\alpha_k$  ( $k^{\text{th}}$  line of  $\mathbb{A}$ ). It is given by the following expression, where  $\bar{p}_i$  and  $\bar{\alpha}_k$  are the average values for the  $i^{\text{th}}$  line of  $\mathbb{P}$  and the  $k^{\text{th}}$  line of  $\mathbb{A}$ , and where  $\sigma_{p_i}$  and  $\sigma_{\alpha_k}$  stand for their standard deviation:

$$\rho_{p_i, \alpha_k} = \frac{\sum_{j=1}^N (\mathbb{P}(i, j) - \bar{p}_i) \times (\mathbb{A}(k, j) - \bar{\alpha}_k)}{N \times \sigma_{p_i} \times \sigma_{\alpha_k}}.$$

## APPENDIX D DIMENSIONALITY REDUCTION

Fig. 14 extends Figure 12 (main manuscript) to all our test ensembles and it confirms visually the conclusions of the table of aggregated scores (Table 2 of the main manuscript).

In particular, it confirms that MT-PGA provides a *trade-off* between the respective advantages of standard techniques such as MDS [66] and t-SNE [117]. Specifically, MDS is known to preserve the input metric well, while t-SNE tends to better preserve the global structure of the data (i.e. the ground-truth classification), at the expense of metric violation.

MT-PGA provides a *balance* between these two behaviors: (i) it improves structure preservation over MDS (it provides equivalent or better NMI/ARI scores for 11 out of 12 ensembles) and (ii) it improves metric preservation over t-SNE (it provides an equivalent or better SIM score for 9 out of 12 ensembles). Visually, this means that MT-PGA groups together the members belonging to the same ground-truth class, while providing a layout which is more faithful than t-SNE's regarding the distances between the corresponding merge trees.

As can be expected, the straightforward PCA preserves the  $W_2^T$  metric poorly (as it is based on the  $L_2$  norm). Since it relies on a rough approximation of  $\mathbb{B}$ , a PCA derived from a vectorization of the BDTs (MT-VEC) preserves poorly the global structure of the ensemble (MT-PGA provides equivalent or better NMI/ARI scores for 11 out of 12 ensembles).

## APPENDIX E METRIC DISTORTION

The section details the computation of the metric distortion indicator *SIM*, which evaluates the preservation of the Wasserstein metric in dimensionality reduction tasks.

Specifically, given two points  $x$  and  $y$  in a planar layout (with BDTs  $\mathcal{B}(f_x)$  and  $\mathcal{B}(f_y)$ ), we first measure their pairwise distortion:

$$\delta(x, y) = \left( \|x - y\|_2 - W_2^T(\mathcal{B}(f_x), \mathcal{B}(f_y)) \right)^2.$$

This measure is then normalized into:

$$\delta'(x, y) = \frac{\delta(x, y)}{\max_{x \neq y} (\delta(x, y))}.$$

Finally, we evaluate the global indicator  $SIM := 1 - \bar{\delta}'$ , where  $\bar{\delta}'$  stands for the average of  $\delta'(x, y)$  for all pairs  $(x, y)$  in the ensemble. *SIM* values lie within the interval  $[0, 1]$  and are optimal near 1.

## APPENDIX F PROJECTED VARIANCE

Fig. 15 extends to 10 dimensions the scatter plots of projected variance reported in Figure 13 (main manuscript), which were computed for only 2 dimensions. This figure confirms the conclusions of Figure 13 (main manuscript). Except for a very mild oscillation for a specific dataset ("*Cloud processes*", PD-PGA, with a local maximum of low amplitude at 4 dimensions), overall, the projected variance is indeed monotonically decreasing in practice for an increasing number of dimensions (i.e. the algorithm does tend to identify the most informative directions first).

## APPENDIX G

### METRIC PARAMETER ANALYSIS

The Wasserstein metric between merge trees (introduced in previous work [92]) is subject to three parameters ( $\epsilon_1$ ,  $\epsilon_2$ , and  $\epsilon_3$ ), in order to adjust the stability/discriminateness balance of the metric. Pont et al. [92] provided a detailed, experimental parameter analysis of their metric, in particular for the specific task of geodesic computation (Appendix 10.3 of [92]).

In this section, we replicate the same experimental protocol, but this time for the specific task of MT-PGA computation. Specifically, Figures 16, 17 and 18 respectively illustrate the effect of the parameters  $\epsilon_1$ ,  $\epsilon_2$ , and  $\epsilon_3$  on the MT-PGA basis.

As described by Pont et al. [92], in the data, moving a branch up the BDT corresponds to only slight modifications, which consists in reconnecting maxima to distinct saddles. For each parameter ( $\epsilon_1$ ,  $\epsilon_2$ , and  $\epsilon_3$ ), the resulting pre-processing addresses cases where nearby saddles have close function values, which impacts the stability of the metric. As discussed in the main manuscript (section 2.3), similarly to Sridaharamurthy et al. [111], Pont et al. mitigate this effect with  $\epsilon_1$ , but they also introduce  $\epsilon_2$  and  $\epsilon_3$  to specifically limit the importance in the metric of branches with a persistence close to that of their parents [92].

For each figure, we study an ensemble consisting of two main clusters:  $A$  (pink spheres, left of Figures 16, 17 and 18) and  $B$  (dark red spheres, right of Figures 16, 17 and 18). These clusters have been synthesized by considering Gaussian mixtures (such that  $B$  has one more prominent feature than  $A$ ) and by generating the other members of the ensemble with variants of these two patterns, by inserting a random additive noise. Specifically, the cluster  $B$  is synthesized out of 2 slightly distinct Gaussian mixtures, yielding two *artificial* sub-clusters  $B'$  and  $B''$ , such that, in each case, the red branch directly connects to a different branch in the sub-clusters  $B'$  and  $B''$ . Then each figure visualizes the impact of each parameter  $\epsilon_1$ ,  $\epsilon_2$ , and  $\epsilon_3$  on the displacement of the red branch, and hence on the resulting MT-PGA basis.

Overall, as discussed in the detailed captions, these three parameters have the effect of moving branches up the input BDTs, hence reducing the structural impact of these branches on the metric, but also improving its stability. In Figures 16, 17 and 18, by moving the red branch up, a larger section of the BDTs of the sub-clusters  $B'$  and  $B''$  become isomorphic, and thus, the two sub-patterns of the cluster  $B$  become closer to each other in  $\mathbb{B}$  and the clusters  $A$  and  $B$  become better differentiated in the MT-PGA basis (cluster  $A$  on the left of the planar layout, pink spheres, cluster  $B$  on the right of the planar layout, dark red spheres).

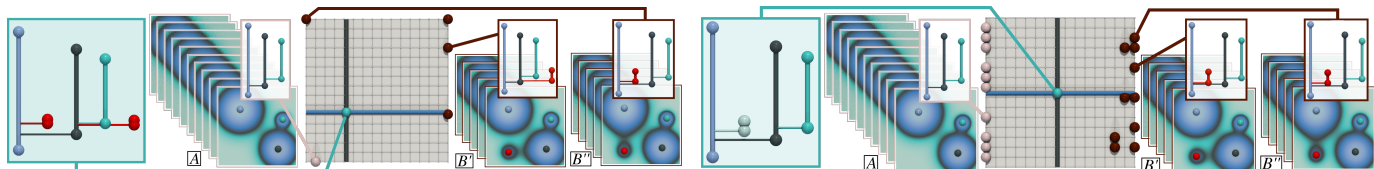


Fig. 16. Impact of the parameter  $\varepsilon_1$  on MT-PGA computation (in this example, left:  $\varepsilon_1 = 0$ , right:  $\varepsilon_1 = 0.3$ ). Initially (left), the red branch in the cluster  $B'$  is not matched to the red branch in the cluster  $B''$  as they have distinct depths in the corresponding BDTs (2 versus 1). However, these features are visually similar in the data (Gaussians with the red maximum in  $B'$  and  $B''$ , bottom left corner of the domain). After the  $\varepsilon_1$  pre-processing (right), the saddle of the red branch in  $B''$  gets merged with its ancestor saddle (whose scalar value was less than  $\varepsilon_1$  away). Consequently, the red branch gets moved up the BDT (i.e. the red branch is attached to the main light blue branch in  $B''$ , right). Since they now have identical depths in the corresponding BDTs, the red branches of the sub-clusters  $B'$  and  $B''$  can now be matched together (right), which results in an overall matching between these two trees which better conveys the resemblance between the two sub-clusters  $B'$  and  $B''$ . Equivalently, one can interpret this procedure of saddle merge in the input trees as a modification of the input scalar field, turning the Gaussian mixture  $B'$  into  $B''$ . In particular, this field modification disconnects the Gaussian with the red maximum from the Gaussian with the black maximum ( $B'$ ) and reconnects it to the Gaussian with the light blue maximum ( $B''$ ). Overall, after the  $\varepsilon_1$  pre-processing (right), the MT-PGA better distinguishes the cluster  $A$  (pink spheres, left of the planar layout) from the cluster  $B$  (dark red spheres, right of the planar layout).

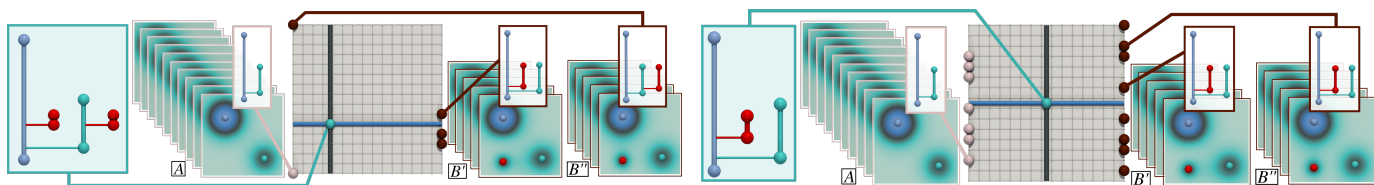


Fig. 17. Impact of the parameter  $\varepsilon_2$  on MT-PGA computation (in this example, left:  $\varepsilon_2 = 1$ , right:  $\varepsilon_2 = 0.8$ ). Initially (left), the red branch in  $B'$  is not matched to the red branch in  $B''$  as they have distinct depths in the corresponding BDTs (1 versus 2). The red branch in  $B''$  has a persistence nearly identical to its parent (cyan). Thus, after local normalization (necessary to guarantee the topological consistency of the interpolated trees), its normalized persistence would become artificially high, which can have an undesirable effect on the metric. The BDT pre-processing addresses this issue and moves up the BDT branches with a relative persistence to their parent larger than  $\varepsilon_2$ . After the  $\varepsilon_2$  pre-processing (right), the red branch in  $B''$  moves up the BDT and becomes adjacent to the main light blue branch. Since they now have identical depths in the corresponding BDTs, the red branches of  $B'$  and  $B''$  can now be matched together (right), which better conveys the resemblance between the two sub-clusters  $B'$  and  $B''$ . Equivalently, one can interpret this procedure of BDT pre-processing as a modification of the input scalar field, turning the Gaussian mixture  $B'$  into  $B''$ . In particular, this field modification disconnects the Gaussian with the red maximum from the Gaussian with the cyan maximum ( $B''$ ) and reconnects it to the Gaussian with the light blue maximum ( $B'$ ). Overall, after the  $\varepsilon_2$  pre-processing (right), the MT-PGA better distinguishes the cluster  $A$  (pink spheres, left of the planar layout) from the cluster  $B$  (dark red spheres, right of the planar layout).

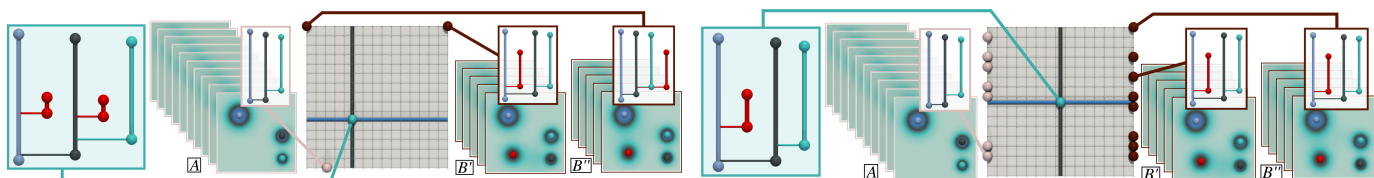


Fig. 18. Impact of the parameter  $\varepsilon_3$  on MT-PGA computation (in this example, left:  $\varepsilon_3 = 0$ , right:  $\varepsilon_3 = 0.7$ ). Initially (left), the red branch in  $B''$  is not matched to the red branch in  $B'$  as they have distinct depths in the corresponding BDTs (1 versus 3). The parameter  $\varepsilon_3$  restricts the application of the above BDT pre-processing ( $\varepsilon_2$ ) and prevents the movement of the most persistent branches (relative persistence larger than  $\varepsilon_3$ ). After the  $\varepsilon_3$  pre-processing (right), the red branch in  $B''$  moves up the BDT and becomes adjacent to the main light blue branch. Since they now have identical depths in the corresponding BDTs, the red branches of  $B'$  and  $B''$  can now be matched together (right), which results in an overall matching between these two trees which better conveys the resemblance between the two sub-clusters  $B'$  and  $B''$ . Equivalently, one can interpret this procedure on the BDTs as a modification of the input scalar field, turning the Gaussian mixture  $B'$  into  $B''$ . In particular, this field modification disconnects the Gaussian with the red maximum from the Gaussian with the cyan maximum ( $B''$ ) and reconnects it to the Gaussian with the light blue maximum ( $B'$ ). Overall, after the  $\varepsilon_3$  pre-processing (right), the MT-PGA better distinguishes the cluster  $A$  (pink spheres, left of the planar layout) from the cluster  $B$  (dark red spheres, right of the planar layout).