Wasserstein Distances, Geodesics and Barycenters of Merge Trees

Mathieu Pont, Jules Vidal, Julie Delon, Julien Tierny

Wasserstein Distances, Geodesics and Barycenters of Merge Trees

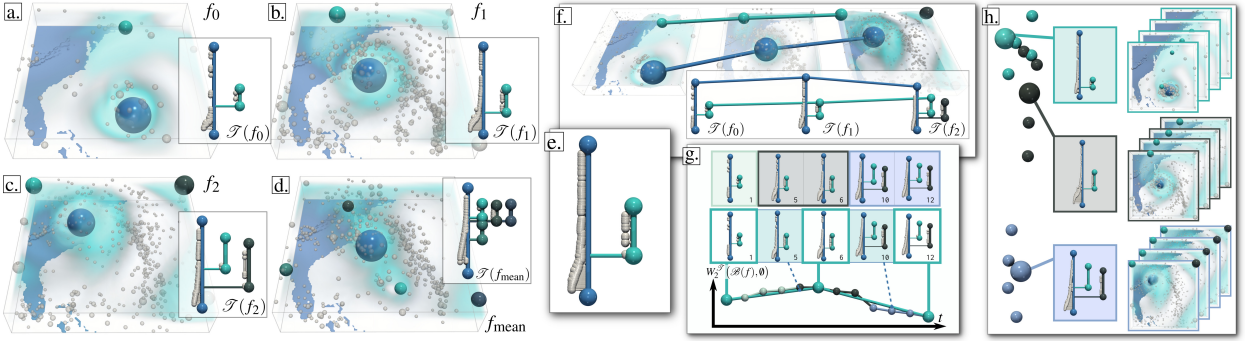Mathieu Pont, Jules Vidal, Julie Delon and Julien Tierny

Fig. 1. The merge trees of three members (a-c) of the Isabel ensemble (wind velocity) concisely and visually encode the number and salience of the features of interest found in the data (eyewall and region of high speed wind, blue and cyan). They also describe how these features are globally connected in the data. In these trees, branches with a low persistence (less than 20% of the function range) are shown with small white arcs. The pointwise mean for the three members (d) exhibits 5 salient maxima (due to distinct eyewall locations, blue, cyan and black) and its merge tree is not representative of the input trees (containing at most 3 large features). In contrast, the Wasserstein barycenter (e) is representative of the input trees, with a number and persistence of large branches that better match the input trees (a-c). Our framework for distances, geodesics and barycenters enables a variety of merge tree based applications, including (f) feature tracking, (g) temporal reduction – *key frames* are automatically identified (white insets) and deleted merge trees (blue insets) are accurately reconstructed with geodesics – and (h) ensemble clustering and summarization – the clusters and centroids automatically computed by our approach provide a visual summary of the main trends of features found in the ensemble.

Abstract— This paper presents a unified computational framework for the estimation of distances, geodesics and barycenters of merge trees. We extend recent work on the edit distance [106] and introduce a new metric, called the *Wasserstein* distance between merge trees, which is purposely designed to enable efficient computations of geodesics and barycenters. Specifically, our new distance is strictly equivalent to the L^2-Wasserstein distance between extremum persistence diagrams, but it is restricted to a smaller solution space, namely, the space of rooted partial isomorphisms between branch decomposition trees. This enables a simple extension of existing optimization frameworks [112] for geodesics and barycenters from persistence diagrams to merge trees. We introduce a task-based algorithm which can be generically applied to distance, geodesic, barycenter or cluster computation. The task-based nature of our approach enables further accelerations with shared-memory parallelism. Extensive experiments on public ensembles and SciVis contest benchmarks demonstrate the efficiency of our approach – with barycenter computations in the orders of minutes for the largest examples – as well as its qualitative ability to generate representative barycenter merge trees, visually summarizing the features of interest found in the ensemble. We show the utility of our contributions with dedicated visualization applications: feature tracking, temporal reduction and ensemble clustering. We provide a lightweight C++ implementation that can be used to reproduce our results.

Index Terms—Topological data analysis, merge trees, scalar data, ensemble data

◆

1 INTRODUCTION

Modern datasets, acquired or simulated, are continuously gaining in geometrical complexity, thanks to the ever-increasing accuracy of acquisition devices or computing power of high performance systems. This geometrical complexity makes interactive exploration and analysis difficult, which challenges the interpretation of the data by the end users. This motivates the definition of expressive data abstractions, capable of capturing the main features of the data into concise representations, which visually convey the most important information to the users.

In that context, Topological Data Analysis (TDA) [32] forms a family of generic, robust, and efficient techniques whose utility has been demonstrated in a number of visualization tasks [55] for revealing the implicit structural patterns present in complex datasets. Examples of popular application fields include turbulent combustion [23, 51, 65], material sciences [40, 53, 54], nuclear energy [71], fluid dynamics [61], bioimaging [4, 20, 26], quantum chemistry [16, 47, 76] or astrophysics [103, 105]. Among the data abstractions developed in TDA (see Sec. 1.1), the merge tree [25], which describes the global structure of the connected components of the sub-level sets of scalar datasets (Fig. 2), is a prominent example in the visualization literature [20, 23, 26].

In practice, in addition to the increasing geometrical complexity of datasets, users are also confronted to the emergence of *ensemble datasets*, where a given phenomenon is not described with only one dataset, but with a collection of datasets, called *ensemble members*. Regarding topological features, a topological data abstraction such as the merge tree can be computed for each ensemble member (possibly insitu [9, 11]). Then, a major challenge for end users is the interpretation of the resulting ensemble of merge trees. To address this, a statistical analysis framework for merge trees is needed, requiring several key building blocks, such as: distances (to compare merge trees), geodesics (to visualize optimum transitions between them), and barycenters (to visualize one merge tree *representative* of a set). These building blocks have been well studied for persistence diagrams [64, 112, 113]. However,

- *M. Pont, J. Vidal, and J. Tierny are with Sorbonne Université and CNRS. E-mail: {mathieu.pont, jules.vidal, julien.tierny}@sorbonne-universite.fr.*
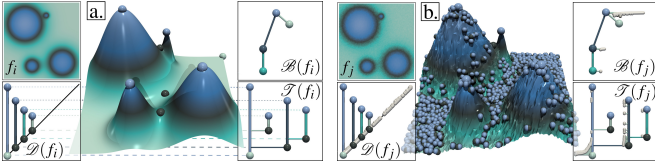- *J. Delon is with University of Paris. E-mail: julie.delon@u-paris.fr*

Fig. 2. Critical points (spheres, white: minima, blue: maxima, other: saddles), persistence diagram (bottom left), merge tree (bottom right) and branch decomposition tree (top right) of a clean (a) and noisy (b) 2D 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. They correspond to noisy features in the data (b).

Fig. 3. The persistence diagram, $\mathscr{D}(f_i)$, and the merge tree, $\mathscr{T}(f_i)$, both visually summarize the number, data range and salience of the features of interest present in the data. However, the persistence diagram represents each individual feature independently, while the merge tree additionally describes how they connect together. This results in a lack of specificity for the persistence diagram which can yield identical data representations for significantly distinct datasets (from left to right, the gaussians with white and cyan spheres have been swapped). In contrast, the merge tree captures this nuance and produces two distinct data representations.

persistence diagrams suffer from a lack of specificity (Fig. 3), which can prevent the identification of distinct feature trends within the ensemble.

This paper addresses this problem by introducing a unified computational framework for the automatic computation of distances, geodesics, barycenters and clusters of merge trees. In particular, we extend recent work on the edit distance [106] and introduce a new metric, called the *Wasserstein* distance between merge trees, which is purposely designed to enable efficient computations of geodesics (i.e. length minimizing morphings) and barycenters. In that regard, our work can be interpreted as an extension of previous work on the edit distance [106], to adapt it to the optimization strategy previously developed for the computation of barycenters of persistence diagrams [112]. We present efficient, task-based algorithms using shared-memory parallelism, resulting in the computation of distances, geodesics and barycenters in practical times for real-life datasets. We illustrate the utility of each of our contributions in dedicated visualization tasks. First, we show that our distance computation algorithm can be used for a merge-tree based tracking of features through time. Second, we show that our framework for computing geodesics between merge trees can be used for the reliable sub-sampling of temporal sequences of merge trees. Third, we illustrate the utility of our barycenters for clustering ensemble members based on their merge trees, while providing cluster centroids which visually summarize the main features of interest present in each cluster.

1.1 Related work

The literature related to our work can be classified into three main 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 modeled and encoded in several ways. In particular, *uncertain* datasets capture variability by modeling each point of the domain as a random variable, whose variability is explicitly modeled by an estimator of an a priori probability density function (PDF). The analysis of uncertain data is a notoriously challenging problem in visualization, described in several surveys [1, 21, 59, 70, 80, 96]. Early techniques focused on estimating the entropy of the random variables [94], their correlations [88] or their gradient variations [86]. The positional uncertainty of level sets has been studied for several interpolation schemes and PDF models [5–7, 87, 90–93, 100]. Similarly, the positional uncertainty of critical points has been studied for Gaussian [66, 78, 79, 85] or uniform distributions [17, 50, 107]. A general limitation of existing methods for uncertain data is their dependence on the specific PDF model for which they have been designed. This reduces their usability for ensemble data, where the PDF estimated from the ensemble members can follow an arbitrary, unknown model. Also, most existing techniques for uncertain data do not consider multi-modal PDF models, which is however necessary when several, distinct trends are present in the ensemble data.

(ii) **Ensemble visualization:** Another way to model and encode variability in data consists in considering ensemble datasets. In this setting, the variability is directly encoded by an ensemble of 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
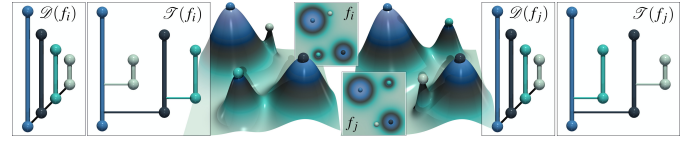
instance, spaghetti plots [30] are a typical example for studying level-set variability, especially for weather data [95, 99]. More specifically, box-plots [115] describe the variability of contours and curves [72]. For flow ensembles, Hummel et al. [58] introduce a Lagrangian framework for classification purposes. Clustering techniques have been investigated, to identify the main trends, and their variability, in ensembles of stream-lines [42] and isocontours [43]. However, only few approaches have applied this overall aggregation strategy to topological objects. Favelier et al. [39] and Athawale et al. [8] introduced approaches for analyzing the variability of critical points and gradient separatrices respectively. Several attempted to generate an aggregated contour tree from an ensemble based on overlap-driven heuristics [63, 117]. Recently, Lohfink et al. [68] introduced an approach for the consistent layout of multiple contour trees, to support effective visual comparisons between the contour trees of the distinct members of an ensemble. Although the above techniques addressed the visualization of ensembles of topological objects, they did not focus explicitly on the computation of a *representative* of multiple topological objects, such as barycenters.

(iii) **Topological methods:** Concepts and algorithms from computational topology [32] have been investigated, adapted and extended by the visualization community for more than twenty years [55, 118]. Popular topological representations include the persistence diagram [32, 36] (Sec. 2.2), which represents the population of features of interest in function of their salience, and which can be computed via matrix reduction [13, 32]. The Reeb graph [18], which describes the connectivity evolution of level sets, has also been widely studied and several efficient algorithms have been documented [31, 81, 83, 111], including parallel algorithms [49]. Efficient algorithms have also been documented for its variants, the merge and contour trees [25, 108] (Sec. 2.3), and parallel algorithms have also been described [2, 27, 48, 69]. The Morse-Smale complex [22, 33, 34], which depicts the global behaviour of integral lines, is another popular topological data abstraction in visualization [29]. Robust and efficient algorithms have been introduced for its computation [52, 97, 102] based on Discrete Morse Theory [44].

Distance metrics, which are necessary ingredients for the computation of barycenters, have been studied for most of the above objects. Inspired by the literature in optimal transport [60, 73], the Wasserstein distance between persistence diagrams [32] (Sec. 2.2) and its variant the Bottleneck distance [36] have been extensively studied. They are based on a bipartite assignment problem, for which exact [75] and approximate [15, 62] implementations are publicly available [110]. Several similarity measures have been introduced for Reeb graphs [57] and their variants [98]. However, since these measures are not distance metrics (the preservation of the triangle inequality is not specifically enforced), they do not seem conducive to barycenter computation. Stable distance metrics between Reeb graphs [12] and merge trees [74] have been studied from a theoretical point of view but their computation, following an exponential time complexity, is not tractable for practical datasets in general, except if reliable correspondence labels between the nodes of the trees are provided on the input [46], which is not practical either for large ensembles. Distances with polynomial time computation algorithms have also been investigated. Similarly to our overall strategy, Beketayev et al. [14] focus on a dual representation,
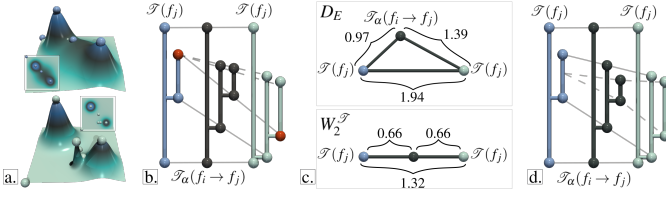
Fig. 4. In this example with two datasets f_i and f_j (a), the optimal matchings (gray) with regard to the edit distance D_E [106] (b) map a maximum to a saddle (red spheres). The resulting linear interpolation $\mathscr{T}_\alpha(f_i \to f_j)$ (b) does not describe a shortest path between the input trees (c, top): $D_E\big(\mathscr{T}(f_i), \mathscr{T}_\alpha(f_i \to f_j)\big) + D_E\big(\mathscr{T}_\alpha(f_i \to f_j), \mathscr{T}(f_j)\big) > D_E\big(\mathscr{T}(f_i), \mathscr{T}(f_j)\big)$. In contrast, our new metric $W_2^{\mathscr{T}}$ enables linear interpolations (d) which exactly coincide with shortest paths (c, bottom). The numbers included in (c) are the actual values for D_E (top) and $W_2^{\mathscr{T}}$ (bottom) for this example.

the *branch decomposition tree* (BDT, Sec. 2.3), but in contrast to our approach, they estimate their distances by iteratively reducing a target mismatch term, in particular, over a search space significantly larger than ours. Sridharamurthy et al. [106] specialize efficient algorithms for computing constrained edit distances between trees [121] to the special case of merge trees (see Appendix 1), resulting in a distance which is computable for real-life datasets and with acceptable practical stability. However, it is not conducive to simple barycenter computations. Indeed, the linear interpolation of the optimal node assignments induced by this metric (Fig. 4) does not result in a shortest path, and hence generates inaccurate midpoints (i.e. inaccurate barycenters given two trees). This further implies that there is no clear or simple strategy for the general computation of barycenters according to that metric.

Regarding the estimation of a *representative* object from a set of topological representations, several approaches emerged recently. A recent line of work [46, 120] introduced a framework for computing a *1-center* of a set of merge trees (i.e. minimizing its maximum distance to the set), according to an interleaving distance. However, as documented by its authors, this approach requires pre-existing, reliable correspondence labels between the nodes of all the input trees, which is not practical with real-life datasets (heuristics need to be considered). Also, since they minimize their *maximum* distance to a set, 1-centers are typically sensitive to outliers, which prevents their usage for estimating trends or supporting clustering tasks (which typically focus on densities rather than maximum distances). This is further evaluated in Sec. 7.2. In contrast, our approach focuses on the estimation of *barycenters* (instead of 1-centers) and computes a tree which minimizes its *average* distance to an ensemble of merge trees (instead of its maximum distance), which is less sensitive to outliers, which better captures trends and which supports clustering tasks. Moreover, the node correspondences between the barycenter and the input trees are automatically estimated by our approach via an assignment optimization present at the core of our distance estimation. Thus our method does not require input correspondences, which makes it readily applicable to real-life ensembles. Several methods [64, 112, 113] have been introduced for the automatic estimation of barycenters (Sec. 2.2) of persistence diagrams (or vectorized variants [3, 24]). However, the persistence diagram can lack specificity in its data characterization (Fig. 3). This limitation is addressed by our work which focuses instead on merge trees.

1.2 Contributions

This paper makes the following new contributions:

1. *A practical distance metric between merge trees:* We extend recent work on the edit distance [106] and introduce a new distance between merge trees, which, in contrast to previous work, is purposely designed to enable efficient computations of geodesics and barycenters. It can be computed efficiently, it has acceptable practical stability and it has a strong connection to established metrics, which eases its interpretation. Specifically, it can be interpreted as a variant of the L^2-Wasserstein distance for persistence diagrams, for which we constrain the underlying search space to account for the additional structural information provided by the merge tree.

2. *A simple approach for computing geodesics between merge trees:* Given our new metric, we present a simple approach for computing geodesics between merge trees. It uses a simple linear interpolation of the assignments resulting from our new metric, enabling the exact computation of geodesics in linear time. This follows from previous work on persistence diagram geodesics [112] and it is made possible thanks to a new, local normalization procedure, guaranteeing the topological consistency of the interpolated trees.

3. *An approach for computing barycenters between merge trees:* Our method for geodesics between merge trees enables a straightforward adaptation of previous optimization strategies for persistence diagram barycenters [112], resulting, to our knowledge, in the first approach for the computation of barycenters of merge trees.

4. *Unified computational framework:* We present a unified computational framework for the estimation of distances, geodesics, barycenters, and clusters of merge trees. In particular, we introduce an efficient, task-based algorithm adapted from previous work on edit distances [106, 121], which is generically applicable to any of the above tasks. Our algorithm supports shared-memory parallelism, allowing for further accelerations in practice.

5. *Applications:* We illustrate the utility of each of our contributions with dedicated visualization tasks, including feature tracking, temporal reduction and ensemble clustering and summarization.

6. *Implementation:* We provide a lightweight C++ implementation of our algorithms that can be used for reproduction purposes.

2 PRELIMINARIES

This section presents the theoretical background of our work. It contains definitions adapted from the Topology ToolKit [110]. We refer the reader to textbooks [32] for an introduction to computational topology.

2.1 Input data

The input data is an ensemble of N piecewise linear (PL) scalar fields $f_i : \mathscr{M} \to \mathbb{R}$, with $i \in \{1, \ldots, N\}$, defined on a PL d-manifold $\mathscr{M}$, with $d \leq 3$ in our applications. The *sub-level set* of f_i, noted $f_{i-\infty}^{-1}(w) = \{p \in \mathscr{M} \mid f_i(p) < w\}$, is defined as the pre-image of $(-\infty, w)$ by f_i. The *super-level set* of f_i is defined symmetrically: $f_{i+\infty}^{-1}(w) = \{p \in \mathscr{M} \mid f_i(p) > w\}$. As w continuously increases, the topology of $f_{i-\infty}^{-1}(w)$ changes at specific vertices of $\mathscr{M}$, called the *critical points* of f_i [10]. In practice, f_i is enforced to contain only isolated, non-degenerate critical points [35, 37]. Critical points are classified by their *index* $\mathscr{I}_i$: 0 for minima, 1 for 1-saddles, $d - 1$ for $(d-1)$-saddles and d for maxima.

2.2 Persistence diagrams

The persistence diagram is a visual summary of the topological features (i.e. connected components, independent cycles, voids) of $f_{i-\infty}^{-1}(w)$. Specifically, 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 [32] states that critical points can be arranged according to this observation in a set of pairs, such that each critical point appears in only one pair (c, c'), with $f_i(c) < f_i(c')$ and $\mathscr{I}_i(c) = \mathscr{I}_i(c') - 1$. 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). Then the persistence diagram, noted $\mathscr{D}(f_i)$, embeds each pair to a single point in 2D at coordinates $\big(f_i(c), f_i(c')\big)$. The *persistence* of a pair is given by its height $f_i(c') - f_i(c)$. Then, the persistence diagram provides a visual overview of the features of interest of a dataset (Fig. 2), where salient features stand out from the diagonal while pairs corresponding to noise are located in the vicinity of the diagonal. Note that, in addition to its interest as a visual summary, the persistence diagram captures all the information about the persistent homology groups of the data [32].

Given two diagrams $\mathscr{D}(f_i)$ and $\mathscr{D}(f_j)$, a pointwise distance, noted d_q (with $q > 0$), can be introduced in the 2D birth/death space between two points $p_i = (x_i, y_i) \in \mathscr{D}(f_i)$ and $p_j = (x_j, y_j) \in \mathscr{D}(f_j)$:

$$d_q(p_i, p_j) = \big(|x_j - x_i|^q + |y_j - y_i|^q\big)^{1/q} = \|p_i - p_j\|_q. \tag{1}$$
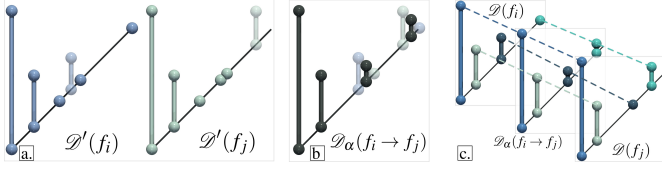
Fig. 5. Distance and geodesic computation between two persistence diagrams $\mathscr{D}(f_i)$ and $\mathscr{D}(f_j)$, given the metric $W_2^{\mathscr{D}}$. The sets $\overline{P_i}$ and $\overline{P_j}$ are shown in transparent (a). The matching induced by the optimal partial assignment ϕ is shown with dashed lines (c). For two diagrams, the Wasserstein barycenter (b) is given, thanks to the d_2 distance in the birth/death space, by the arithmetic mean of the matched points. Then, the linear interpolation of the matchings (c) describes a geodesic [112].

By convention, $d_q(p_i, p_j)$ is set to zero if both p_i and p_j exactly lie on the diagonal ($x_i = y_i$ and $x_j = y_j$). Let P_i be a subset of the off-diagonal points of $\mathscr{D}(f_i)$ and $\overline{P_i}$ its complement (i.e. the other off-diagonal points of $\mathscr{D}(f_i)$ not in P_i). Let $(\phi, \overline{P_i}, \overline{P_j})$ be a *partial assignment* between $\mathscr{D}(f_i)$ and $\mathscr{D}(f_j)$, i.e. a bijective map between P_i and a subset of off-diagonal points P_j of $\mathscr{D}(f_j)$, with complement $\overline{P_j}$ (Fig. 5a). Then, the L^q-Wasserstein distance, noted $W_q^{\mathscr{D}}$, can be introduced as:

$$W_q^{\mathscr{D}}\big(\mathscr{D}(f_i), \mathscr{D}(f_j)\big) = \min_{(\phi, \overline{P_i}, \overline{P_j}) \in \Phi} \left(\sum_{p_i \in P_i} d_q\big(p_i, \phi(p_i)\big)^q \right. \tag{2}$$

$$+ \sum_{p_i \in \overline{P_i}} d_q\big(p_i, \Delta(p_i)\big)^q \tag{3}$$

$$\left. + \sum_{p_j \in \overline{P_j}} d_q\big(\Delta(p_j), p_j\big)^q \right)^{1/q} \tag{4}$$

where Φ is the set of all possible partial assignments mapping each point $p_i \in \mathscr{D}(f_i)$ to a point $\phi(p_i) = p_j \in \mathscr{D}(f_j)$ (line 2), or to its diagonal projection, $\Delta(p_i) = (\frac{x_i + y_i}{2}, \frac{x_i + y_i}{2})$, denoting the removal of the corresponding feature from $\mathscr{D}(f_i)$ or $\mathscr{D}(f_j)$ (lines 3 and 4). Intuitively, the Wasserstein metric optimizes a matching between the two diagrams, and evaluates their distance given the resulting mismatch. In practice, $\mathscr{D}(f_i)$ and $\mathscr{D}(f_j)$ are *augmented* into $\mathscr{D}'(f_i)$ and $\mathscr{D}'(f_j)$ [62], by injecting the diagonal projections of one diagram into the other (Fig. 5a):

$$\mathscr{D}'(f_i) = \mathscr{D}(f_i) \cup \{\Delta(p_j) \mid p_j \in P_j\}$$
$$\mathscr{D}'(f_j) = \mathscr{D}(f_j) \cup \{\Delta(p_i) \mid p_i \in P_i\}.$$

This augmentation (Fig. 5a) preserves the distance, while making the assignment problem balanced, and thus easily solvable with traditional algorithms [15, 75] (with $P_i' = \mathscr{D}'(f_i), P_j' = \mathscr{D}'(f_j)$ and $\overline{P_i'} = \overline{P_j'} = \emptyset$).

Given a set $\mathscr{S}_{\mathscr{D}} = \{\mathscr{D}(f_1), \ldots, \mathscr{D}(f_N)\}$ of persistence diagrams, let $F(\mathscr{D}, \alpha)$ be the Fréchet energy of the set, under the metric $W_2^{\mathscr{D}}$, with the coefficients $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$, such that $\alpha_i \in [0,1]$ and $\sum_i \alpha_i = 1$:

$$F(\mathscr{D}, \alpha) = \sum_{\mathscr{D}(f_i) \in \mathscr{S}_{\mathscr{D}}} \alpha_i W_2^{\mathscr{D}}\big(\mathscr{D}, \mathscr{D}(f_i)\big)^2. \tag{5}$$

Then the diagram $\mathscr{D}^* \in \mathbb{D}$ (where $\mathbb{D}$ is the space of persistence diagrams) which minimizes $F(\mathscr{D}, \alpha)$ is called the *Wasserstein barycenter* of the set $\mathscr{S}_{\mathscr{D}}$ (or its Fréchet mean under the metric $W_2^{\mathscr{D}}$). In practice, the coefficients α_i are all set to the same value ($\alpha_i = 1/N$, $\forall i \in \{1, \ldots, N\}$). When $N = 2$ and $\alpha_1 = \alpha_2 = 0.5$ (Fig. 5b), $\mathscr{D}^*$ becomes a midpoint between $\mathscr{D}(f_i)$ and $\mathscr{D}(f_j)$ and the set of possible values for α_1 and α_2 (Fig. 5c) describes a geodesic in $\mathbb{D}$ (i.e. length minimizing path) with regard to the L^2-Wasserstein metric [112].

2.3 Merge trees

The *join* tree, noted $\mathscr{T}^-(f_i)$, is a visual summary of the connected components of $f_{i-\infty}^{-1}(w)$ [25]. It is a 1-dimensional simplicial complex defined as the quotient space $\mathscr{T}^-(f_i) = \mathscr{M}/\sim$ 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 $\mathscr{T}^+(f_i)$, 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, noted generically $\mathscr{T}(f_i)$ 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. These trees are often visualized according to a persistence-driven *branch decomposition* [82], 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 merge tree, where each persistent branch is represented as a vertical segment. The *branch decomposition tree* (BDT), noted $\mathscr{B}(f_i)$, 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 merge tree. In Fig. 2, the BDTs (top right insets) can be interpreted as the dual of the branch decompositions (bottom right insets, with matching colors): each vertical segment in the branch decomposition (bottom) corresponds to a node in the BDT (top) and each horizontal segment (bottom, denoting an adjacency relation between branches) corresponds to an arc in the BDT. Intuitively, the BDT, like the persistence diagram, describes the population of (extremum-saddle) persistence pairs present in the data. However, unlike the persistence diagram, it additionally captures adjacency relations between them.

3 WASSERSTEIN DISTANCES BETWEEN MERGE TREES

This section introduces our new distance metric between merge trees, which is specifically designed for the subsequent computation of geodesics (Sec. 4) and barycenters (Sec. 5). For this, we bridge the gap between the edit distance between merge trees [106] and existing work addressing the computation of geodesics and barycenters for persistence diagrams according to the L^2-Wasserstein distance [112].

3.1 Overview

The end goal of our work is the computation of barycenters of merge trees. For this, we extend the edit distance D_E [106] (formalized in Appendix 1, additional material), to make it fit the optimization strategy used for barycenters of persistence diagrams [112]. Our key idea consists in transforming D_E such that it becomes strictly equivalent to the L^2-Wasserstein distance of persistence diagrams, but given a *restricted* space of possible assignments, constrained by the structure of the input trees $\mathscr{T}(f_i)$ and $\mathscr{T}(f_j)$, hence its name Wasserstein distance between merge trees. Then, thanks to this compatibility with the L^2-Wasserstein distance, the assignments resulting from our metric can be directly used for interpolation-based geodesic and barycenter computations (Secs. 4 and 5). Overall, our strategy involves four major modifications to the edit distance D_E [106], detailed in the remainder of this section:

1. To consider assignments between persistence pairs instead of merge tree nodes, we consider an edit distance between the BDTs $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$ (Sec. 2.3) instead of the input merges trees $\mathscr{T}(f_i)$ and $\mathscr{T}(f_j)$ (as done with D_E). This is described in Sec. 3.2.

2. We constrain the assignment search space to the space of *rooted partial isomorphisms*. Specifically, similarly to D_E, we enforce the assignment of disjoint subtrees of $\mathscr{B}(f_i)$ to disjoint subtrees of $\mathscr{B}(f_j)$. Moreover, in contrast to D_E, we additionally extend this constraint by enforcing the destruction of entire subtrees upon the destruction of their root. These two constraints together enforce assignments describing isomorphisms between rooted subtrees of $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$. Such isomorphisms pave the way for interpolation-based geodesics. This is described in Secs. 3.2, 3.3 and 4.1.

3. We introduce a cost model based on the Euclidean distance d_2 to enable geodesic computation by linear interpolation of the assignments in the 2D birth/death space. This is described in Secs. 3.2 and 4.1.

4. We finally extend our metric with a local normalization term, which enforces nested birth-death values, along the interpolation of the assignments, for nested branches. This is described in Sec. 4.2.

3.2 Definition and properties

Given two input merge trees, $\mathscr{T}(f_i)$ and $\mathscr{T}(f_j)$, we first consider their BDTs $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$ (Sec. 2.3). Let B_i be a subset of the nodes of $\mathscr{B}(f_i)$ and $\overline{B_i}$ its complement. Note that each node in B_i corresponds to a persistence pair of $\mathscr{D}(f_i)$. Let $(\phi', \overline{B_i}, \overline{B_j})$ be a partial assignment between B_i and a subset B_j of the nodes of $\mathscr{B}(f_j)$ (with complement $\overline{B_j}$). Then we introduce the L^2-Wasserstein distance $W_2^{\mathscr{T}}$ between the BDTs $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$ of the merge trees $\mathscr{T}(f_i)$ and $\mathscr{T}(f_j)$ as:

$$W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big) = \min_{(\phi',\overline{B_i},\overline{B_j})\in\Phi'} \Big(\sum_{b_i\in B_i} \gamma\big(b_i\to\phi'(b_i)\big)^2 \tag{6}$$

$$+ \sum_{b_i\in\overline{B_i}} \gamma(b_i\to\emptyset)^2 \tag{7}$$

$$+ \sum_{b_j\in\overline{B_j}} \gamma(\emptyset\to b_j)^2 \Big)^{1/2} \tag{8}$$

where Φ' is the space of constrained partial assignments mapping disjoints subtrees of $\mathscr{B}(f_i)$ to disjoint subtrees of $\mathscr{B}(f_j)$, *and* mapping entire subtrees to the empty tree $\emptyset$ if their root is itself mapped to $\emptyset$. Then, given the kth direct child of b_i, noted b_i^k, it follows that b_i^k either maps through ϕ' to a direct child of $\phi'(b_i)\in\mathscr{B}(f_j)$ (then $b_i, b_i^k\in B_i$) or to the empty tree $\emptyset$ (then the subtree rooted in b_i^k, noted $\mathscr{B}(f_i, b_i^k)$, belongs to $\overline{B_i}$). This further implies that the rooted subtrees $B_i\subseteq\mathscr{B}(f_i)$ and $B_j = \phi'(B_i)\subseteq\mathscr{B}(f_j)$ are isomorphic and we call $(\phi',\overline{B_i},\overline{B_j})$ a *rooted partial isomorphism*. Unlike D_E (see Appendix 1) but similarly to $W_2^{\mathscr{D}}$ (Eq. 2), the cost of each operation (mapping, line 18, destruction, line 18, and creation, line 19) is squared, and the square root of the sum of the squared costs is considered as the overall distance.

Next, we define the edit costs as follows (we recall that each branch $b_i\in\mathscr{B}(f_i)$ exactly coincides with a persistence pair $p_i\in\mathscr{D}(f_i)$):

$$\begin{aligned}
\gamma\big(b_i\to\phi'(b_i)\big) &= d_2\big(b_i,\phi'(b_i)\big) \\
\gamma(b_i\to\emptyset) &= d_2\big(b_i,\Delta(b_i)\big) \\
\gamma(\emptyset\to b_j) &= d_2\big(\Delta(b_j),b_j\big).
\end{aligned} \tag{9}$$

Note that the expression of the L^2-Wasserstein distance $W_2^{\mathscr{T}}$ between merge trees (Eq. 6) is therefore identical to the expression of the Wasserstein distance between persistence diagrams (Eq. 2) for $q = 2$, at the notable exception of the search space of the partial assignments $\Phi'\subset\Phi$, which is constrained to rooted partial isomorphisms. $W_2^{\mathscr{T}}$ is indeed a distance metric (the proof is included in Appendix 2, supplemental material): it is non-negative and symmetric, it preserves the identity of indiscernibles as well as the triangle inequality. Moreover, since $\Phi'\subset\Phi$, it follows that $W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big)\geq W_2^{\mathscr{D}}\big(\mathscr{D}(f_i),\mathscr{D}(f_j)\big)$, which was one of the main motivations of our work (i.e. to exploit the merge tree to define a more discriminative metric, Fig. 3). Similarly to Sridharamurthy et al. [106], we mitigate saddle swap instabilities in a preprocessing step, by merging adjacent saddles in the input trees if their difference in scalar value is smaller than a threshold $\varepsilon_1\in[0,1]$ (relative to the largest difference between adjacent saddles). Then, when $\varepsilon_1 = 1$, it follows that $W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big) = W_2^{\mathscr{D}}\big(\mathscr{D}(f_i),\mathscr{D}(f_j)\big)$. This simple merging strategy significantly improves the practical stability of $W_2^{\mathscr{T}}$, as empirically studied in Sec. 7.2 (Fig. 14).

3.3 Computation

This section describes our algorithm for the recursive exploration of the search space Φ' (Eq. 6). It is based on the same recursive traversal as Zhang's algorithm [121], which we simplify as our search space is significantly more constrained. Specifically, as detailed in Appendix 3, our distance evaluation between subtrees (Eq. 11) involves fewer solutions and it is restricted to subtrees rooted at identical depth only.

Given the subtree $\mathscr{B}(f_i, b)$ of $\mathscr{B}(f_i)$ (rooted in b) and b^k the kth direct child of b in $\mathscr{B}(f_i, b)$, the distance between the subtree $\mathscr{B}(f_i, b)$ and the empty tree $\emptyset$ is then obtained recursively by:

$$W_2^{\mathscr{T}}\big(\mathscr{B}(f_i,b),\emptyset\big) = \Big(\gamma(b\to\emptyset)^2 + \sum_k W_2^{\mathscr{T}}\big(\mathscr{B}(f_i,b^k),\emptyset\big)^2 \Big)^{1/2}. \tag{10}$$
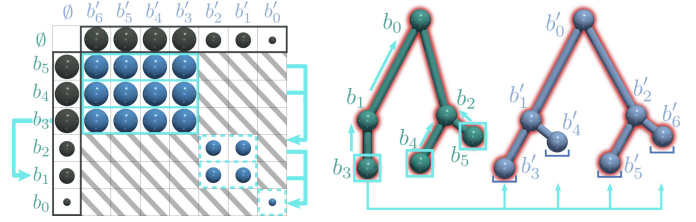

Fig. 6. The exploration of the space Φ' (a candidate is highlighted in red, right) relies on the evaluation of the sparse matrix $\mathbb{T}$ of subtree distances (left). Our task-based algorithm optimizes the parallel computation of independent terms. Spheres of equal radius in $\mathbb{T}$ denote independent terms and arrows between the lines of $\mathbb{T}$ indicate task dependence (equivalently illustrated with arrows in the input BDTs, right).

The first step of our algorithm consists in evaluating $W_2^{\mathscr{T}}\big(\mathscr{B}(f_i,b),\emptyset\big)$ with Eq. 10 for all branches $b\in\mathscr{B}(f_i)$ (and similarly for $\mathscr{B}(f_j)$).

Next, let $\mathscr{F}(f_i,b)$ be the *forest* of b in $\mathscr{B}(f_i)$: $\mathscr{F}(f_i,b)$ is the set of all the subtrees rooted at the k direct children of b: $\mathscr{F}(f_i,b) = \{\mathscr{B}(f_i,b^1),\mathscr{B}(f_i,b^2),\ldots,\mathscr{B}(f_i,b^k)\}$. Then the distance between two subtrees $\mathscr{B}(f_i,b_i)$ and $\mathscr{B}(f_j,b_j)$ is set to $+\infty$ when b_i and b_j have distinct depths (gray crosshatching lines in Fig. 6, left). Otherwise (spheres in Fig. 6, left), it is obtained recursively by:

$$\begin{aligned}
W_2^{\mathscr{T}}\big(\mathscr{B}(f_i,b_i),\mathscr{B}(f_j,b_j)\big) &= \Big(\gamma(b_i\to b_j)^2 \tag{11} \\
&+ W_2^{\mathscr{T}}\big(\mathscr{F}(f_i,b_i),\mathscr{F}(f_j,b_j)\big)^2 \Big)^{1/2}.
\end{aligned}$$

Let F_i be a subset of the forest $\mathscr{F}(f_i,b_i)$ and $\overline{F_i}$ its complement. The distance between two forests is then given recursively by:

$$W_2^{\mathscr{T}}\big(\mathscr{F}(f_i,b_i),\mathscr{F}(f_j,b_j)\big) = \min_{(\phi'',\overline{F_i},\overline{F_j})\in\Phi''} \Big(\sum_{f_i\in F_i} W_2^{\mathscr{T}}\big(f_i,\phi''(f_i)\big)^2$$

$$+ \sum_{f_i\in\overline{F_i}} W_2^{\mathscr{T}}\big(f_i,\emptyset\big)^2$$

$$+ \sum_{f_j\in\overline{F_j}} W_2^{\mathscr{T}}\big(f_j,\emptyset\big)^2 \Big)^{1/2}$$

where $(\phi'',\overline{F_i},\overline{F_j})$ becomes the solution of a local, partial assignment problem between forests, mapping F_i to a subset $F_j\in\mathscr{F}(f_j,b_j)$ (with complement $\overline{F_j}$) or to the empty tree $\emptyset$, and which can be solved with traditional assignment algorithms [15, 75] (see Sec. 2.2). Then, the overall distance $W_2^{\mathscr{T}}$ between the two input merge trees is obtained by estimating Eq. 11 at the roots of $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$, and solving recursively the local assignment problems between forests (the recursion returns are illustrated with arrows in Fig. 6). Note that if $\varepsilon_1 = 1$ (Sec. 3.2), all the branches of $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$ get attached to the roots and the recursive local assignment problems between forests (above) become only one, large, assignment problem between all branches. Thus, when $\varepsilon_1 = 1$, this algorithm indeed becomes strictly equivalent to the resolution of the assignment problem involved in $W_2^{\mathscr{D}}$ (Sec. 2.2).

3.4 Parallelism

Similarly to Zhang [121], Eqs. 10 and 11 can be estimated recursively. To avoid redundant computations, the distances between the forests $\mathscr{F}(f_i,b_i)$ and $\mathscr{F}(f_j,b_j)$ are stored at the entry (b_i,b_j) of a matrix $\mathbb{F}$ (of size $|\mathscr{B}(f_i)|\times|\mathscr{B}(f_j)|$), while the distances between the subtrees $\mathscr{B}(f_i,b_i)$ and $\mathscr{B}(f_j,b_j)$ (used within the assignment problems between higher forests) are stored in a matrix $\mathbb{T}$ (of the same size, see Fig. 6).

In our work, we additionally express this computation in terms of *tasks*, to leverage task-based shared memory parallelism. In particular, we initiate a task for each independent term of Eqs. 10 and 11, which is ready for computation (see Fig. 6), as further detailed in Appendix 4. Then, the number of parallel tasks is initially bounded by the number of leaves in the input BDTs (which is typically much larger than the number of cores) and progressively decreases during the computation.
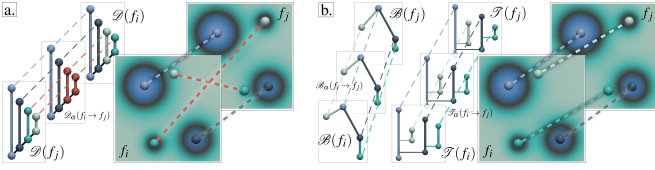
Fig. 7. Our geodesic computation extends interpolation-based geodesics from persistence diagrams (a) to merge trees (b). The interpolated BDT $\mathscr{B}_\alpha(f_i \to f_j)$ is obtained by linear interpolation (with local normalization) of the partial isomorphism ϕ' in the birth/death space. In the data, the feature matching (dashed lines) induced by ϕ' with $W_2^{\mathscr{T}}$ (b) better preserves the global structure of the data than ϕ with $W_2^{\mathscr{D}}$ (a, red crossing).

4 WASSERSTEIN GEODESICS BETWEEN MERGE TREES

This section introduces our approach for the efficient computation of geodesics between merge trees, according to the metric $W_2^{\mathscr{T}}$ (Sec. 2). For this, we leverage the rooted partial isomorphism resulting from the distance computation, as well as linear interpolations of the matchings, as introduced for persistence diagrams [112].

4.1 Definition and properties

Given two input merge trees $\mathscr{T}(f_i)$ and $\mathscr{T}(f_j)$, our approach to geodesic computation (Fig. 7) simply consists in linearly interpolating the rooted partial isomorphism $(\phi', \overline{B_i}, \overline{B_j})$ resulting from the optimization involved in the computation of $W_2^{\mathscr{T}}\big(\mathscr{B}(f_i), \mathscr{B}(f_j)\big)$ (Eq. 6). In particular, given the two BDTs $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$, the interpolated BDT, noted $\mathscr{B}_\alpha(f_i \to f_j)$ with $\alpha \in [0,1]$ such that $\mathscr{B}_0(f_i \to f_j) = \mathscr{B}(f_i)$ and $\mathscr{B}_1(f_i \to f_j) = \mathscr{B}(f_j)$, is obtained by considering the union of:

1. the linear interpolation $B_\alpha \subseteq \mathscr{B}_\alpha(f_i \to f_j)$, between the nodes $B_i \subseteq \mathscr{B}(f_i)$ and these of $B_j \subseteq \mathscr{B}(f_j)$, given the isomorphism ϕ' (the trees B_i, B_j and B_α are then isomorphic, Fig. 7):

$$b(\alpha) = (1-\alpha)b + \alpha\phi'(b) \quad \forall b \in B_i \tag{12}$$

2. the linear interpolation of the destruction of the subtrees $\overline{B_i}$, noted $\overline{B_i}^\alpha \subseteq \mathscr{B}_\alpha(f_i \to f_j)$ ($\overline{B_i}$ and $\overline{B_i}^\alpha$ are also isomorphic):

$$b(\alpha) = (1-\alpha)b + \alpha\Delta(b) \quad \forall b \in \overline{B_i} \tag{13}$$

3. the linear interpolation of the creation of the subtrees $\overline{B_j}$, noted $\overline{B_j}^\alpha \subseteq \mathscr{B}_\alpha(f_i \to f_j)$ ($\overline{B_j}$ and $\overline{B_j}^\alpha$ are also isomorphic):

$$b(\alpha) = (1-\alpha)\Delta(b) + \alpha b \quad \forall b \in \overline{B_j}. \tag{14}$$

Similarly to the distance $W_2^{\mathscr{D}}$ between persistence diagrams, since the edit costs involved in the distance $W_2^{\mathscr{T}}$ are Euclidean distances in the birth/death space (Eq. 9), the interpolated branches $b(\alpha)$ of $\mathscr{B}_\alpha(f_i \to f_j)$ can be efficiently computed with the simple linear interpolations described above. As detailed in Appendix 5, the resulting interpolated tree $\mathscr{B}_\alpha(f_i \to f_j)$ is indeed on a geodesic given $W_2^{\mathscr{T}}$.
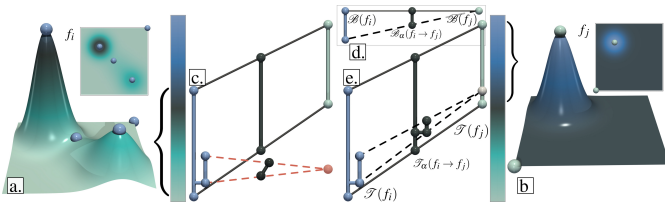

Fig. 8. Given two scalar fields f_i (a) and f_j (b), a simple interpolation of the birth/death values of the branches of their BDTs may result in inconsistencies upon branch destruction (red): the interpolated merge tree (black) in (c) is disconnected, unlike the interpolated BDT (d). Our local normalization (Sec. 4.2) addresses this issue by enforcing nested birth/death values for nested branches. This results in a valid interpolated merge tree (e) whose BDT is indeed equal to the interpolated BDT (e).

4.2 From branch decomposition trees to merge trees

The previous section described the computation of geodesics between BDTs, given $W_2^{\mathscr{T}}$. In this section, given an interpolated BDT $\mathscr{B}_\alpha(f_i \to f_j)$, we describe how to retrieve the corresponding merge tree $\mathscr{T}_\alpha(f_i \to f_j)$ (i.e. a merge tree whose BDT is indeed equal to $\mathscr{B}_\alpha(f_i \to f_j)$).

A requirement for an arbitrary BDT $\mathscr{B}$ to be the valid BDT of a merge tree $\mathscr{T}$ is that subtrees of $\mathscr{B}$ need to respect a *nesting condition* on their birth/death (i.e. x, y values) (to respect the Elder rule, Sec. 2.2). In particular, given a direct child b_α^k of a branch $b_\alpha \in \mathscr{B}_\alpha(f_i \to f_j)$, we need to guarantee that $[x_{b_\alpha^k}, y_{b_\alpha^k}] \subseteq [x_{b_\alpha}, y_{b_\alpha}]$. While this is guaranteed by construction for the subset $B_\alpha \subseteq \mathscr{B}_\alpha(f_i \to f_j)$ (B_α is isomorphic to B_i and B_j), this is not necessarily the case for the subsets of $\mathscr{B}_\alpha(f_i \to f_j)$ involved in subtree creation or destruction ($\overline{B_i}^\alpha$ and $\overline{B_j}^\alpha$, Sec. 4.1). In particular, since the branches involved in destructions map independently to the diagonal (Eq. 13), it is possible that the above nesting condition is not respected along their interpolation. This is shown in Fig. 8c (red interpolation), where the resulting merge tree, $\mathscr{T}_\alpha(f_i \to f_j)$, becomes disconnected and hence invalid (i.e. $\mathscr{B}_\alpha(f_i \to f_j)$, Fig. 8d, *is* connected and not equal to the BDT of $\mathscr{T}_\alpha(f_i \to f_j)$ from Fig. 8c).

In the following, we introduce a pre-processing step for the trees $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$ (together with its inverse post-processing step), which we call *local normalization*, which addresses this issue and guarantees the above nesting condition, even in case of destruction/creation.

Given a direct child b_i^k of a branch $b_i \in \mathscr{B}(f_i)$, we consider the following local, birth/death normalization $\mathscr{N}(b_i^k) = \big(\mathscr{N}_x(b_i^k), \mathscr{N}_y(b_i^k)\big)$:

$$\mathscr{N}_x(b_i^k) = (x_{b_i^k} - x_{b_i})/(y_{b_i} - x_{b_i})$$
$$\mathscr{N}_y(b_i^k) = (y_{b_i^k} - x_{b_i})/(y_{b_i} - x_{b_i}).$$

Once this pre-process is recursively completed, the Wasserstein distance $W_2^{\mathscr{T}}$ between the locally normalized BDTs, noted $\mathscr{N}\big(\mathscr{B}(f_i)\big)$ and $\mathscr{N}\big(\mathscr{B}(f_j)\big)$ is computed as described in Sec. 3.3. Then, the interpolation of the locally normalized BDTs, noted $\mathscr{N}\big(\mathscr{B}_\alpha(f_i \to f_j)\big)$, is evaluated as described in Sec. 4.1. Next, the local normalization is recursively reverted to turn $\mathscr{N}\big(\mathscr{B}_\alpha(f_i \to f_j)\big)$ back into $\mathscr{B}_\alpha(f_i \to f_j)$, by explicitly evaluating $\mathscr{N}^{-1}(b_\alpha^{\mathscr{N}})$ for each branch $b_\alpha^{\mathscr{N}} \in \mathscr{N}\big(\mathscr{B}_\alpha(f_i \to f_j)\big)$. Now, even in case of branch destruction, by construction, the birth/death interval of each interpolated branch $\mathscr{N}(b_\alpha)$, noted $[\mathscr{N}_x(b_\alpha), \mathscr{N}_y(b_\alpha)]$, is included in $[0,1]$ (since $\Delta\big(\mathscr{N}(b_i)\big) \subseteq [\mathscr{N}_x(b_i), \mathscr{N}_y(b_i)] \subseteq [0,1]$). Therefore, after reverting the local normalization, we have the guarantee that $[x_{b_\alpha}^k, y_{b_\alpha}^k] \subseteq [x_{b_\alpha}, y_{b_\alpha}]$ for all the branches b_α of $\mathscr{B}_\alpha(f_i \to f_j)$.

At this stage, $\mathscr{B}_\alpha(f_i \to f_j)$ indeed respects the nesting condition on the birth/death values of all its subtrees. Then, given the dual relation between merge trees and BDTs, the merge tree $\mathscr{T}_\alpha(f_i \to f_j)$ can be simply obtained by creating a vertical branch for each node b_α of $\mathscr{B}_\alpha(f_i \to f_j)$ and connecting them according the arcs of $\mathscr{B}_\alpha(f_i \to f_j)$, as illustrated in Fig. 8 (right). The distance $W_2^{\mathscr{T}}$ between $\mathscr{N}\big(\mathscr{B}(f_i)\big)$ and $\mathscr{N}\big(\mathscr{B}(f_j)\big)$ then still describes a metric between $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$, such that $\mathscr{B}_\alpha(f_i \to f_j)$ is indeed on a geodesic (see Appendix 6).

Note that the local normalization shrinks all the input branches to the interval $[0,1]$, irrespective of their original persistence. To mitigate this effect, we introduce a pre-processing step on the input BDTs,
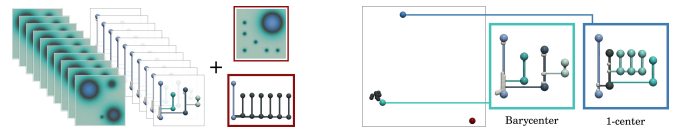

Fig. 9. Visual comparison between our barycenter (Sec. 5) and the 1-center of Yan et al. [119,120]. Left: an ensemble is created with an outlier member f_j (red, 7 persistent branches) and 10 noisy versions of a field f_i (4 persistent branches). Right: planar view of the ensemble computed by multi-dimensional scaling of $W_2^{\mathscr{T}}$. The barycenter computed with our approach (cyan) is more similar to the merge trees of f_i (same number and persistence of large branches) and hence better captures the overall trend of the ensemble, despite the presence of the outlier f_j (red sphere).
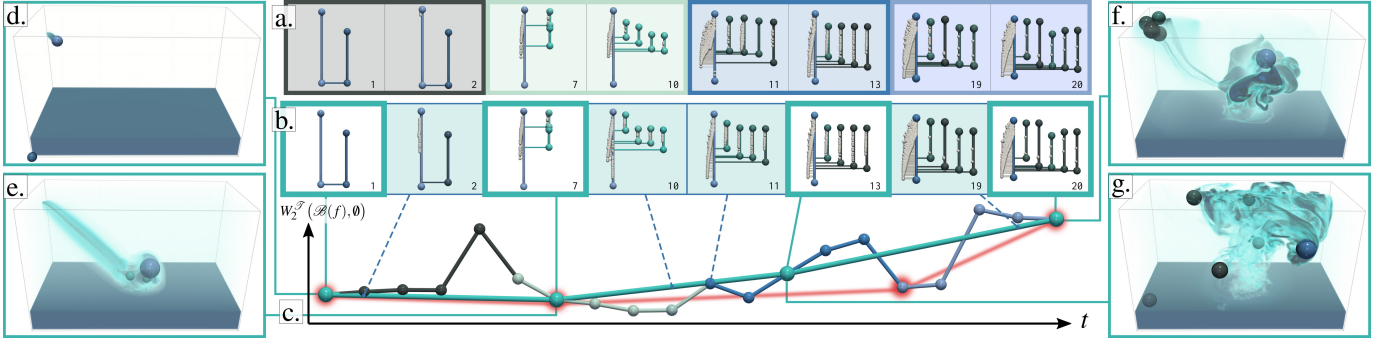
Fig. 10. Geodesic computation for the reduction (b) of temporal sequences of merge trees (a). Our algorithm greedily removes from the sequence the trees that it can accurately estimate by geodesic computation (trees with blue background (b)). This reduction is also visualized with the three curves in (c), plotting the distance $W_2^{\mathcal{T}}(\mathscr{B}(f), \emptyset)$ to the empty tree $\emptyset$ over time (multiple colors: original sequence, cyan: reduction by $W_2^{\mathcal{T}}$, red: reduction by $W_2^{\mathscr{D}}$). This iterated removal of trees highlights *key frames* in the sequence (d-g) corresponding to key phases of an asteroid impact simulation [84]: initial state (black, time steps 1-5), approach (light green, 6-10), impact (blue, 11-15), aftermath (light blue, 16-20). In contrast, a similar greedy optimization based on the distance $W_2^{\mathscr{D}}$ between persistence diagrams (red curve) fails at capturing the *impact* phase (blue) of this sequence.
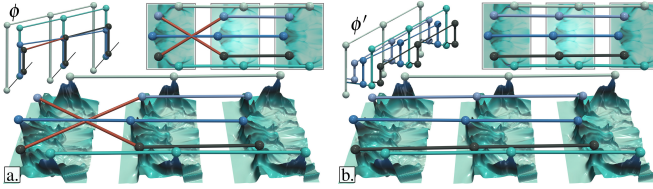

Fig. 11. Tracking features (the five most persistent maxima, spheres) in time-varying 2D data (ion density during universe formation [109]): optimal assignment ϕ (a) of $W_2^{\mathscr{D}}$ (Sec. 2.2), optimal isomorphism ϕ' (b) of $W_2^{\mathcal{T}}$ (Sec. 3.2). Since $W_2^{\mathscr{D}}$ considers persistence pairs individually, it can generate incorrect matchings resulting in a characteristic *crossing* (a, red). Our distance improves this aspect (b) thanks to its more constrained search space, which better preserves the global structure of the data.

which moves, up the trees, subtrees rooted at branches with a relative persistence smaller than ε_3, until their persistence relative to their parent becomes smaller than a threshold ε_2. This has the practical effect of reducing the normalized persistence of small branches corresponding to small features. Overall, ε_1, ε_2 and ε_3 are the only parameters of our approach and we use a unique, default set of values ($\varepsilon_1 = 0.05$, $\varepsilon_2 = 0.95$ and $\varepsilon_3 = 0.9$) in our experiments (Sec. 7). In the remainder, we will consider that all the input BDTs are normalized this way.

5 WASSERSTEIN BARYCENTERS OF MERGE TREES

This section introduces our approach for the computation of barycenters of merge trees, for the metric $W_2^{\mathcal{T}}$ (Sec. 2). The resulting barycenters will serve as core tools for clustering ensembles of merge trees (Sec. 6).

5.1 Definition

Let $\mathscr{S}_{\mathscr{B}} = \{\mathscr{B}(f_1), \mathscr{B}(f_2), \ldots, \mathscr{B}(f_N)\}$ be a set of N BDTs. Similarly to Eq. 5, The Fréchet energy, under the metric $W_2^{\mathcal{T}}$, is given by:

$$F(\mathscr{B}) = \sum_{\mathscr{B}(f_i) \in \mathscr{S}_{\mathscr{B}}} W_2^{\mathcal{T}}\left(\mathscr{B}, \mathscr{B}(f_i)\right)^2.$$

We call a *Wasserstein barycenter* of $\mathscr{S}_{\mathscr{B}}$, a BDT $\mathscr{B}^* \in \mathbb{B}$ (where $\mathbb{B}$ is the space of BDTs) which minimizes $F(\mathscr{B})$. It is a centroid of the set, i.e. a tree which minimizes the sum of its distances to the set.

5.2 Computation

Our distance $W_2^{\mathcal{T}}$ (Sec. 2) is identical to $W_2^{\mathscr{D}}$, but with a smaller search space, restricted to rooted partial isomorphisms. This enabled an extension of interpolation-based geodesics from persistence diagrams to merge trees (Sec. 4). Given these two components, the strategy presented by Turner et al. [112] for minimizing the Fréchet energy over the space of persistence diagrams can be directly extended to our framework. For this, we consider an algorithm that resembles a Lloyd relaxation [67], and which alternates an *(i) assignment* and an *(ii) update* procedure. First, the candidate $\mathscr{B}$ is initialized at an arbitrary tree

of $\mathscr{S}_{\mathscr{B}}$. Then the assignment step *(i)* computes an optimal assignment $(\phi'_i, \overline{B_i}, \overline{B_{\mathscr{B}}})$ between $\mathscr{B}$ and each tree $\mathscr{B}(f_i) \in \mathscr{S}_{\mathscr{B}}$. Next, the update step *(ii)* updates the candidate $\mathscr{B}$ to a position in $\mathbb{B}$ which minimizes $F(\mathscr{B})$ under the current set of assignments $(\phi'_i, \overline{B_i}, \overline{B_{\mathscr{B}}})_{i=1,\ldots,N}$. This is achieved by moving each branch $b \in \mathscr{B}$ (in the birth/death space) to the arithmetic mean of the assignments (by generalizing the interpolation defined in Eqs. 12, 13, and 14, to more than two trees):

$$b \leftarrow \frac{1}{N} \sum_{i=1,\ldots,N} \begin{cases} \phi'_i(b) & \text{if } b \in B_i \\ \Delta(b) & \text{if } b \in \overline{B_i} \\ b & \text{if } b \in \overline{B_{\mathscr{B}}}. \end{cases}$$

This overall assignment/update sequence is then iterated (as discussed in Appendix 7, each iteration of this sequence decreases the Fréchet energy constructively). In our implementation, the algorithm stops and returns the barycenter estimation $\mathscr{B}^*$ when the Fréchet energy decreased by less than 1% between two consecutive iterations. Given $\mathscr{B}^*$, we obtain its dual merge tree $\mathcal{T}^*$ as described in Sec. 4.2. Fig. 9 illustrates a barycenter computed with this strategy for a toy example.

5.3 Parallelism

The N assignment problems (between the candidate $\mathscr{B}$ and the trees of the set $\mathscr{S}_{\mathscr{B}}$, Sec. 5.2) are independent and can be computed in parallel. However, this naive strategy is subject to load imbalance, as the input trees can have different sizes. Hence, each iteration would be bounded by the sequential execution of the largest of the N assignment problems.

We address this issue by leveraging the task-based parallelization of our distance computation algorithm (Sec. 3.4). In particular, we use a single task pool for all of the N assignment problems. Then, the task environment picks up at runtime the tasks to compute irrespective of their tree of origin, and place them on different threads. This fine scheduling granularity has the beneficial effect of triggering the execution of the tasks of a new assignment problem while a first problem is reaching completion (and thus exploiting less threads, Sec. 3.4). This improves thread load imbalance and thus increases the overall parallel efficiency.

6 APPLICATIONS

The section illustrates the utility of our contributions (distances, geodesics, and barycenters) in concrete visualization tasks (Fig. 1).

6.1 Branch matching for feature tracking

Our distance (Sec. 3.3) relies on the optimization of a partial isomorphism between the input BDTs. Then, the resulting matchings can be used to track features in time-varying data, as studied for persistence diagrams [104]. Fig. 11 illustrates this on a temporal sequence (SciVis contest 2008 [109]). Since $W_2^{\mathscr{D}}$ considers persistence pairs individually, it can generate inconsistent matchings with a typical incorrect *crossing* in the feature tracking (already visible on synthetic data, Fig. 7). Our distance $W_2^{\mathcal{T}}$ improves this aspect by better preserving the global structure of the data, thanks to our more constrained, merge-tree driven,
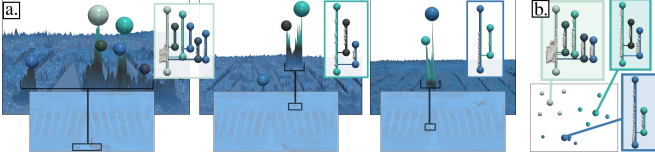
Fig. 12. Three members (a) of an acquired ensemble, corresponding to distinct volcanic eruptions [41]. Our clustering approach correctly assigns the members to each cluster (b, distinct colors in the planar view, generated in a post-process by multi-dimensional scaling of $W_2^{\mathcal{T}}$). Our centroids (larger spheres in the planar view) provide a visual summary of the features of interest (matching colors) for each cluster.
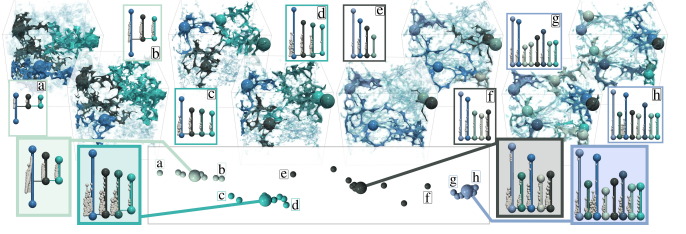

Fig. 13. Eight members (top) of a cosmology ensemble [56], and their merge trees (persistent maxima are displayed with matching colors in the data). Our clustering approach correctly assigns the members to each cluster (distinct colors in the bottom planar view, generated in post-process by multi-dimensional scaling of $W_2^{\mathcal{T}}$). Our centroids (large spheres, bottom) provide a visual summary which is well representative of the trees in the cluster (same number and persistence of large branches, automatically color-coded based on their matching to their centroid).

assignment search space. Overall, our matchings provide visual hints to the users, to help them relate features from distinct time steps.

6.2 Geodesics for temporal reduction

The topological analysis of time-varying data typically requires the computation of a topological representation, for instance a merge tree, for each time step. Although merge trees are usually orders of magnitude smaller than the original data, the resulting sequence of merge trees can still represent considerable amounts of data. To address this issue, we exploit our geodesic computation (Sec. 4) for the reduction of temporal sequences of merge trees. In particular, we greedily remove from the sequence, one by one, the merge trees which can be accurately reconstructed by simple geodesic computation, until the sequence only contains a target number of merge trees (see the detailed algorithm in Appendix 8). This enables the reliable visualization of time-varying sequences of merge trees at greatly reduced storage costs. The remaining merge trees (white background, Fig. 10) correspond to *key frames* of the sequence, i.e. time steps of particular significance in terms of the features of interest. In contrast, a similar strategy based on persistence diagram interpolation (red curve) fails at identifying a key frame in one of the key phases of the sequence (impact, in blue). Also, note that the reduced merge trees (reconstructed with geodesics, blue background) are visually highly similar to the trees from the input sequence.

6.3 Barycenters for topological clustering

To understand the main trends within an ensemble, in terms of features of interest, it may be desirable to cluster the ensemble by grouping members with a similar *topological profile*. For this, we adapt the *k*-means algorithm [28, 38] to the problem of clustering merge trees. In particular, this can be easily achieved by using our merge tree barycenter computation algorithm (Sec. 5) as the centroid estimation routine of *k*-means, and by using $W_2^{\mathcal{T}}$ (Sec. 2) to measure the distance between merge trees. Note that in practice, our entire computational framework is implemented in this single clustering algorithm (with a unique task pool), as the above clustering generalizes the barycenter problem ($k = 1$) as well as the geodesic and distance problems ($N = 2$).

Figs 12 and 13 present clustering examples obtained with this strategy on an acquired [41] and cosmology ensemble [56]. In both cases, our approach correctly assigns the members to each cluster. Moreover, the centroids computed by our algorithm provide a visual summary of the features of interest found in each cluster, enabling global overviews (Figs. 12, right, and 13, bottom) summarizing the topological profile of each of the main trends found in the ensemble. In both figures, the tree branches of the ensemble members are automatically colored with the color of their matched centroid branch. This matching visualization enables users to visually relate the centroid to concrete features in the data (Fig. 12) and to compare matching features across multiple members (i.e. which have been matched to the same centroid branch, Fig. 13). Then, the centroid, in addition to being a visual summary, also acts as a reference point for the visual comparison of ensemble members.

7 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 [48] 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, 110].

Our experiments were performed on a variety of simulated and acquired 2D and 3D ensembles used in previous work [39] (vorticity and sea surface height) or extracted from past SciVis contests: 2004 (wind velocity magnitude [114]), 2006 (wavefront velocity magnitude [77]), 2008 (ion concentration [109]), 2014 (sulfur dioxide concentration [41]), 2015 (dark matter density [56]), 2016 (salt concentration [45]), 2017 (pressure [116]), 2018 (matter density [84]). A detailed specification of these ensembles is provided in Appendix 9.

7.1 Time performance

The time complexity of our algorithm for exploring the search space of $W_2^{\mathcal{T}}$ (Sec. 3.3) is similar to that of the edit distance [106, 121]. It takes $\mathcal{O}(|\mathcal{B}|^2)$ steps in practice, with $|\mathcal{B}|$ the number of nodes in the input BDTs (in our implementation, each local forest assignment problem is solved with the efficient *Auction* approximation [15] with default parameters). Once $W_2^{\mathcal{T}}\left(\mathcal{B}(f_i), \mathcal{B}(f_j)\right)$ is computed, the computation of a point on the geodesic (Sec. 4) between $\mathcal{B}(f_i)$ and $\mathcal{B}(f_j)$ is obtained in $\mathcal{O}(|\mathcal{B}|)$ steps. Regarding our barycenter computation algorithm (Sec. 5), each of its iterations takes $\mathcal{O}(N|\mathcal{B}|^2)$ steps. Tab. 1 evaluates the practical time performance of our computational framework for the barycenter computation (which includes itself distance and geodesic computations). In sequential mode, we observe that the running time is indeed a function of the number of ensemble members (N) and the average size of the trees ($\mathcal{B}$). It is slightly slower for $W_2^{\mathcal{T}}$ than for $W_2^{\mathcal{D}}$, but runtimes remain comparable overall. In parallel, speedups are the most important for the largest examples. However, the iterative nature of our barycenter optimization algorithm seems to limit parallel efficiency globally (the end of each iteration still constitutes a strong synchronization). For the smaller examples, the cost of the task runtime seems to become non-negligible in comparison to the actual computation, resulting in moderate speedups. Still, our parallelization significantly reduces runtimes overall, with less than 3 minutes of computation on average and at most 15 minutes for the largest examples.

7.2 Framework quality

$W_2^{\mathcal{T}}$ is indeed a distance metric (Appendix 2). It is more discriminative than $W_2^{\mathcal{D}}$ (i.e. $W_2^{\mathcal{T}} \geq W_2^{\mathcal{D}}$, Sec. 2, Fig. 3). Fig. 14 evaluates empirically its stability. For this, given a scalar field f_i, a noisy version f_j is created such that $\|f_i - f_j\|_\infty \leq \varepsilon$, for increasing values of ε. Then, we observe the evolution of $W_2^{\mathcal{T}}\left(\mathcal{B}(f_i), \mathcal{B}(f_j)\right)$, as a function of ε (Fig. 14, right), to estimate how $W_2^{\mathcal{T}}$ varies under input perturbations. For $\varepsilon_1 = 1$, we have $W_2^{\mathcal{T}} = W_2^{\mathcal{D}}$ (Sec. 2) and the curve evolves nearly linearly ($W_2^{\mathcal{D}}$ is stable [112]). For other ε_1 values, the curves indicate clear *transition* points (colored dots) before which $W_2^{\mathcal{T}}$ evolves nearly linearly too. This indicates that for reasonable noise levels (smaller than the ε value of each transition point, vertical lines), $W_2^{\mathcal{T}}$ is also stable and that only mild increases of ε_1 result in fast shifts of these transition points (to an accepted noise level of 64% at $\varepsilon_1 = 0.15$). This illustrates overall

| Dataset	N	$|\mathscr{B}|$	$W_2^{\mathcal{D}}$ (1 c.)	$W_2^{\mathcal{T}}$ (1 c.)	$W_2^{\mathcal{T}}$ (20 c.)	Speedup
Asteroid Impact [84] (3D)	7	1,295	514.71	450.91	93.11	4.84
Cloud processes [116] (2D)	12	1,209	54.90	124.99	35.14	3.55
Viscous fingering [45] (3D)	15	118	5.68	5.12	3.89	1.31
Dark matter [56] (3D)	40	2,592	3,172.37	3,083.24	471.45	6.53
Volcanic eruptions [41] (2D)	12	811	171.13	140.02	48.52	2.88
Ionization front [109] (2D)	16	135	10.40	12.10	8.20	1.47
Ionization front [109] (3D)	16	763	682.76	1,277.72	219.61	5.81
Earthquake [77] (3D)	12	1,203	191.54	509.59	117.31	4.34
Isabel [114] (3D)	12	1,338	330.88	284.19	62.70	4.53
Starting Vortex [39] (2D)	12	124	7.72	5.58	6.11	0.91
Sea Surface Height [39] (2D)	48	1,787	4,509.78	10,557.07	881.49	11.97
Vortex Street [39] (2D)	45	23	1.71	1.90	1.44	1.31
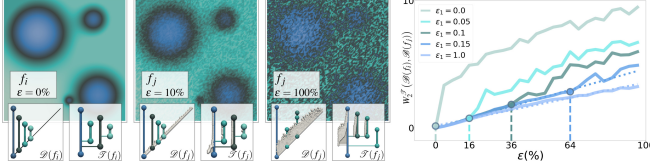
Fig. 14. Empirical stability evaluation. Given an input scalar field f_i, a noisy version f_j is created by inserting a random noise of increasing amplitude ε (left). The evolution of $W_2^{\mathcal{T}}(\mathscr{B}(f_i), \mathscr{B}(f_j))$ with ε (right), for varying values of ε_1 (Sec. 3.2), indicates clear transition points (colored dots) before which $W_2^{\mathcal{T}}$ evolves nearly linearly. Before these transition points (i.e. before these noise levels, vertical lines), $W_2^{\mathcal{T}}$ is stable.

that the stability of $W_2^{\mathcal{T}}$ can indeed be controlled with ε_1 and that small values already lead to stable results for reasonable noise levels. A detailed empirical analysis of the other two parameters of our approach (ε_2, ε_3, Sec. 4.2) is provided in Appendix 10 (supplemental material).

Next, we study the practical relevance of $W_2^{\mathcal{T}}$ by evaluating our clustering performance. For this, each ensemble of Tab. 1 is associated with a ground truth classification (distinct phases of a time-varying phenomenon, distinct input parameters, etc), by following the companion specifications [41, 45, 56, 77, 84, 109, 114, 116]. Clustering performance is evaluated with accepted scores, namely the normalized mutual information and adjusted rand index (NMI, ARI). When using our barycenters (Sec. 5), our clustering approach (Sec. 6.3) achieves a perfect classification for all ensembles ($NMI = ARI = 1$). These scores decrease to $NMI = 0.78$ and $ARI = 0.69$ on average when using, within k-means, a barycenter of persistence diagrams [112] ($\varepsilon_1 = 1$), and to $NMI = 0.73$ and $ARI = 0.56$ when using the 1-center of Yan et al. [120] (obtained with the authors' implementation [119], using leaf labels generated by our distance computation, Sec. 2). This simply confirms experimentally that 1-centers in general are not suited for clustering tasks. A standard clustering approach (multi-dimensional scaling to kD followed by k-means) using the distance D_E [106] achieves lower average scores than our approach, with $NMI = 0.89$ and $ARI = 0.85$ on average. Overall, this confirms that D_E induces more discriminative classifiers than $W_2^{\mathcal{D}}$, and that our metric $W_2^{\mathcal{T}}$ further improves that.

Fig. 15 shows the evolution of the Fréchet energy for our barycenter algorithm (Sec. 5) for various ε_1 values. In practice, the algorithm stops when the Fréchet energy decreases by less than 1% between consecutive iterations, which occurs early in the process.

Fig. 9 provides a visual comparison between our barycenter and the 1-center of Yan et al. [120] (obtained with the authors' implementation [119], using leaf labels generated by our distance computation, Sec. 2). This figure confirms the general sensitivity in practice of 1-centers to outliers, and the ability of barycenters to better capture the main trends in the ensemble. From a qualitative perspective, our framework enables the computation of faithful interpolations of merge trees: the reconstructed trees, blue background (Fig. 10), are visually very similar to the input trees. Moreover, our framework produces barycenters (Figs. 1, 12, and 13) which capture well the main features of the input ensemble: for each cluster, the resulting centroid is visually similar to the input trees of the cluster (same number and persistence of large branches). Then, our clustering framework, coupled with our centroids, provides a faithful visual summary of the features of interest, for each of the main trends (i.e. for each cluster) found in the ensemble.
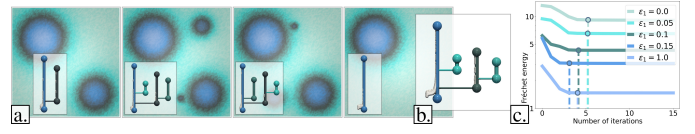

Fig. 15. Evolution of the Fréchet energy for estimating the barycenter (b) for an ensemble of 100 noisy variants of four fields (a). The energy (c) is shown for several ε_1 values (Sec. 3.2). In practice, we stop the algorithm when the energy decreases by less than 1% (vertical lines).

7.3 Limitations

The search space associated with our metric $W_2^{\mathcal{T}}$ is constrained to rooted partial isomorphisms. Then, if a matching exists between two BDTs (i.e. if they are not both destructed when optimizing $W_2^{\mathcal{T}}$), it has to match their roots together. In other words, $W_2^{\mathcal{T}}$ nearly always matches the most persistent branch of the two trees together, which might be too restrictive (in particular for feature tracking applications). Note however, that $W_2^{\mathcal{D}}$ behaves equivalently: the most persistent branch of $\mathscr{B}(f_i)$ corresponds to the component of $f_{i-\infty}^{-1}(w)$ created in the global minimum of f_i, which in principle has infinite persistence and which is typically treated separately when evaluating $W_2^{\mathcal{D}}$. Similarly to Sridharamurthy et al. [106], saddle swap instabilities are handled in our approach by a pre-processing step which merges adjacent saddles (controlled by ε_1). An alternative would consist in exploring the space of all possible branch decompositions (not necessarily persistence-driven), as studied by Beketayev et al. [14]. However, the search space would then become significantly larger. Moreover, the nesting of birth/death values within the BDTs would no longer be guaranteed, which is however a key property which we exploit in our framework (Sec. 4). When computing barycenters of persistence diagrams, Vidal et al. [113] showed that the optimization could be drastically accelerated by introducing persistence pairs progressively along the iterations, while implicitly maintaining previous assignments at each initialization. We leave the study of such a progressive strategy to future work, although the fact that $W_2^{\mathcal{T}}$ handles many small assignment problems (unlike $W_2^{\mathcal{D}}$) indicates that such a strategy may result in only modest gains for merge trees. Fig. 14 provides an empirical evaluation of the stability of $W_2^{\mathcal{T}}$. Similarly to Sridharamurthy et al. [106], we believe that the theoretical investigation of the stability of $W_2^{\mathcal{T}}$ goes beyond the scope of this paper and we leave it for future work.

8 CONCLUSION

In this paper, we presented a computational framework for the estimation of distances, geodesics and barycenters of merge trees, with applications to feature tracking, temporal reduction and ensemble clustering and summarization. Our approach filled the gap between the edit distance [106] and existing optimization frameworks for persistence diagrams [112]. Our work enables faithful interpolations of merge trees (Fig. 10) and the generation of merge trees representative of a set (Figs. 1, 12, and 13). Moreover, our task-based algorithm enables automatic barycenter 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, the question of defining relevant and computable metrics for these objects is still an active research debate. Moreover, as illustrated by this paper, extending existing metrics to make them conducive to efficient geodesic computation further requires additional efforts. 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, based on our framework, we will study the definition of more sophisticated statistical indexes (for instance by investigating a notion of covariance matrix for merge trees), to support even more advanced analyses of large-scale ensemble data.

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] T. Athawale and A. Entezari. Uncertainty quantification in linear interpolation for isosurface extraction. *IEEE TVCG*, 2013.

[6] T. Athawale and C. R. Johnson. Probabilistic asymptotic decider for topological ambiguity resolution in level-set extraction for uncertain 2d data. *IEEE TVCG*, 2019.

[7] T. Athawale, E. Sakhaee, and A. Entezari. Isosurface visualization of data with nonparametric models for uncertainty. *IEEE TVCG*, 2016.

[8] 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.

[9] 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.

[10] T. F. Banchoff. Critical points and curvature for embedded polyhedral surfaces. *The American Mathematical Monthly*, 1970.

[11] A. C. Bauer, H. Abbasi, J. Ahrens, H. Childs, B. Geveci, S. Klasky, K. Moreland, P. O'Leary, V. Vishwanath, B. Whitlock, and E. W. Bethel. In-situ methods, infrastructures, and applications on high performance computing platforms. *CGF*, 2016.

[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. Lukasczyk, D. Sakurai, M. Soler, P. Steneteg, J. Tierny, W. Usher, J. Vidal, and M. Wozniak. An Overview of the Topology ToolKit. In *TopoInVis*, 2019.

[20] A. Bock, H. Doraiswamy, A. Summers, and C. T. Silva. TopoAngler: Interactive Topology-Based Extraction of Fishes. *IEEE TVCG*, 2018.

[21] 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.

[22] 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.

[23] 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.

[24] P. Bubenik. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 2015.

[25] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. In *Symp. on Dis. Alg.*, 2000.

[26] H. A. Carr, J. Snoeyink, and M. van de Panne. Simplifying Flexible Isosurfaces Using Local Geometric Measures. In *IEEE VIS*, 2004.

[27] H. A. Carr, G. H. Weber, C. M. Sewell, and J. P. Ahrens. Parallel peak pruning for scalable SMP contour tree computation. In *LDAV*, 2016.

[28] M. E. Celebi, H. A. Kingravi, and P. A. Vela. A comparative study

[29] L. De Floriani, U. Fugacci, F. Iuricich, and P. Magillo. Morse complexes for shape segmentation and homological analysis: discrete models and algorithms. *CGF*, 2015.

[30] P. Diggle, P. Heagerty, K.-Y. Liang, and S. Zeger. *The Analysis of Longitudinal Data*. Oxford University Press, 2002.

[31] H. Doraiswamy and V. Natarajan. Computing reeb graphs as a union of contour trees. *IEEE TVCG*, 2013.

[32] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2009.

[33] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-smale complexes for piecewise linear 3-manifolds. In *SoCG*, 2003.

[34] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical morse complexes for piecewise linear 2-manifolds. In *SoCG*, 2001.

[35] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manfiolds. *DCG*, 2003.

[36] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *DCG*, 2002.

[37] H. Edelsbrunner and E. P. Mucke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM ToG*, 1990.

[38] C. Elkan. Using the triangle inequality to accelerate k-means. In *ICML*, 2003.

[39] G. Favelier, N. Faraj, B. Summa, and J. Tierny. Persistence Atlas for Critical Point Variability in Ensembles. *IEEE TVCG (IEEE VIS)*, 2018.

[40] G. Favelier, C. Gueunet, and J. Tierny. Visualizing ensembles of viscous fingers. In *IEEE SciVis Contest*, 2016.

[41] D. Feng, B. Hentschel, S. Griessbach, L. Hoffmann, and M. von Hobe. The IEEE SciVis Contest. http://sciviscontest.ieeevis.org/2014/, 2014.

[42] F. Ferstl, K. Bürger, and R. Westermann. Streamline variability plots for characterizing the uncertainty in vector field ensembles. *IEEE TVCG*, 2016.

[43] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. Visual analysis of spatial variability and global correlations in ensembles of iso-contours. *CGF*, 2016.

[44] R. Forman. A User's Guide to Discrete Morse Theory. *AM*, 1998.

[45] C. Garth, B. Geveci, B. Hentschel, J. Kuhnert, I. Michel, T.-M. Rhyne, and S. Schröder. The IEEE SciVis Contest. http://sciviscontest.ieeevis.org/2016/, 2016.

[46] E. Gasparovic, E. Munch, S. Oudot, K. Turner, B. Wang, and Y. Wang. Intrinsic interleaving distance for merge trees. *CoRR*, 1908.00063, 2019.

[47] D. Guenther, R. Alvarez-Boto, J. Contreras-Garcia, J.-P. Piquemal, and J. Tierny. Characterizing molecular interactions in chemical systems. *IEEE TVCG (IEEE VIS)*, 2014.

[48] C. Gueunet, P. Fortin, J. Jomier, and J. Tierny. Task-Based Augmented Contour Trees with Fibonacci Heaps. *IEEE TPDS*, 2019.

[49] C. Gueunet, P. Fortin, J. Jomier, and J. Tierny. Task-based Augmented Reeb Graphs with Dynamic ST-Trees. In *EGPGV*, 2019.

[50] D. Günther, J. Salmon, and J. Tierny. Mandatory critical points of 2D uncertain scalar fields. *CGF*, 2014.

[51] A. Gyulassy, P. Bremer, R. Grout, H. Kolla, J. Chen, and V. Pascucci. Stability of dissipation elements: A case study in combustion. *CGF*, 2014.

[52] A. Gyulassy, P. Bremer, and V. Pascucci. Shared-Memory Parallel Computation of Morse-Smale Complexes with Improved Accuracy. *IEEE TVCG (IEEE VIS)*, 2018.

[53] A. Gyulassy, M. A. Duchaineau, V. Natarajan, V. Pascucci, E. Bringa, A. Higginbotham, and B. Hamann. Topologically clean distance fields. *IEEE TVCG (IEEE VIS)*, 2007.

[54] 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.

[55] 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.

[56] B. Hentschel, B. Geveci, M. Turk, and S. Skillman. The IEEE SciVis Contest. http://sciviscontest.ieeevis.org/2015/, 2015.

[57] M. Hilaga, Y. Shinagawa, T. Komura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *ACM SIGGRAPH*, 2001.

[58] M. Hummel, H. Obermaier, C. Garth, and K. I. Joy. Comparative visual analysis of lagrangian transport in CFD ensembles. *IEEE TVCG*, 2013.

of efficient initialization methods for the k-means clustering algorithm. *Expert Syst. Appl.*, 2013.

[59] C. R. Johnson and A. R. Sanderson. A next step: Visualizing errors and uncertainty. *IEEE Computer Graphics and Applications*, 2003.

[60] L. Kantorovich. On the translocation of masses. *AS URSS*, 1942.

[61] J. Kasten, J. Reininghaus, I. Hotz, and H. Hege. Two-dimensional time-dependent vortex regions based on the acceleration magnitude. *IEEE TVCG*, 2011.

[62] M. Kerber, D. Morozov, and A. Nigmetov. Geometry helps to compare persistence diagrams. *ACM Journal of Experimental Algorithmics*, 2016.

[63] M. Kraus. Visualization of uncertain contour trees. In *IVTA*, 2010.

[64] T. Lacombe, M. Cuturi, and S. Oudot. Large Scale computation of Means and Clusters for Persistence Diagrams using Optimal Transport. In *NIPS*, 2018.

[65] 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 (IEEE VIS)*, 2006.

[66] T. Liebmann and G. Scheuermann. Critical points of gaussian-distributed scalar fields on simplicial grids. *CGF*, 2016.

[67] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 1982.

[68] A. P. Lohfink, F. Wetzels, J. Lukasczyk, G. H. Weber, and C. Garth. Fuzzy contour trees: Alignment and joint layout of multiple contour trees. *CGF*, 2020.

[69] S. Maadasamy, H. Doraiswamy, and V. Natarajan. A hybrid parallel algorithm for computing and tracking level set topology. In *Proc. of HiPC*, 2012.

[70] A. Maceachren, A.Robinson, S. Hopper, S. Gardner, R. Murray, M. Gahegan, and E. Hetzler. Visualizing geospatial information uncertainty: What we know and what we need to know. *CGIS*, 2005.

[71] 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.

[72] M. Mirzargar, R. Whitaker, and R. Kirby. Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE TVCG*, 2014.

[73] G. Monge. Mémoire sur la théorie des déblais et des remblais. *Académie Royale des Sciences de Paris*, 1781.

[74] D. Morozov, K. Beketayev, and G. H. Weber. Interleaving distance between merge trees. In *TopoInVis*. 2014.

[75] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 1957.

[76] 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.

[77] K. Olsen, S. Day, B. Minster, R. Moore, Y. Cui, A. Chourasia, M. Thiebaux, H. Francoeur, P. Maechling, S. Cutchin, and K. Nunes. The IEEE SciVis Contest. http://sciviscontest.ieeevis.org/2006/, 2006.

[78] M. Otto, T. Germer, H.-C. Hege, and H. Theisel. Uncertain 2D vector field topology. 2010.

[79] M. Otto, T. Germer, and H. Theisel. Uncertain topology of 3D vector fields. *IEEE PV*, 2011.

[80] A. T. Pang, C. M. Wittenbrink, and S. K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 1997.

[81] S. Parsa. A deterministic *o(m log m)* time algorithm for the reeb graph. In *SoCG*, 2012.

[82] V. Pascucci, K. Cole-McLaughlin, and G. Scorzelli. Multi-resolution computation and presentation of contour trees. In *IASTED*, 2004.

[83] V. Pascucci, G. Scorzelli, P. T. Bremer, and A. Mascarenhas. Robust on-line computation of Reeb graphs: simplicity and speed. *ACM ToG*, 2007.

[84] J. Patchett and G. R. Gisler. The IEEE SciVis Contest. http://sciviscontest.ieeevis.org/2018/, 2018.

[85] C. Petz, K. Pöthkow, and H.-C. Hege. Probabilistic local features in uncertain vector fields with spatial correlation. *CGF*, 2012.

[86] T. Pfaffelmoser, M. Mihai, and R. Westermann. Visualizing the variability of gradients in uncertain 2D scalar fields. *IEEE TVCG*, 2013.

[87] T. Pfaffelmoser, M. Reitinger, and R. Westermann. Visualizing the positional and geometrical variability of isosurfaces in uncertain scalar fields. *CGF*, 2011.

[88] T. Pfaffelmoser and R. Westermann. Visualization of global correlation structures in uncertain 2D scalar fields. *CGF*, 2012.

[89] S. Popinet. Gerris Flow Solver. http://gfs.sourceforge.net/wiki/index.php/Main_Page, 2006.

[90] K. Pöthkow and H.-C. Hege. Positional uncertainty of isocontours:

[91] K. Pöthkow and H.-C. Hege. Nonparametric models for uncertainty visualization. *CGF*, 2013.

[92] K. Pöthkow, C. Petz, and H.-C. Hege. Approximate level-crossing probabilities for interactive visualization of uncertain isocontours. *Int. J. Uncert. Quantif.*, 2013.

[93] K. Pöthkow, B. Weber, and H.-C. Hege. Probabilistic marching cubes. In *CGF*, 2011.

[94] K. Potter, S. Gerber, and E. W. Anderson. Visualization of uncertainty without a mean. *IEEE Computer Graphics and Applications*, 2013.

[95] 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.

[96] J. C. Potter K, Rosen P. From quantification to visualization: A taxonomy of uncertainty visualization approaches. *IFIP AICT*, 2012.

[97] 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.

[98] H. Saikia, H. Seidel, and T. Weinkauf. Extended branch decomposition graphs: Structural comparison of scalar data. *CGF*, 2014.

[99] 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.

[100] S. Schlegel, N. Korn, and G. Scheuermann. On the interpolation of data with normally distributed uncertainty for visualization. *IEEE TVCG (IEEE VIS)*, 2012.

[101] D. Shepard. A two-dimensioanl interpolation function for irregularly-spaced data. In *ACM National Conference*, 1968.

[102] N. Shivashankar and V. Natarajan. Parallel Computation of 3D Morse-Smale Complexes. *CGF*, 2012.

[103] 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.

[104] M. Soler, M. Plainchault, B. Conche, and J. Tierny. Lifted Wasserstein matcher for fast and robust topology tracking. In *IEEE LDAV*, 2018.

[105] T. Sousbie. The persistent cosmic web and its filamentary structure: Theory and implementations. *Royal Astronomical Society*, 2011.

[106] R. Sridharamurthy, T. B. Masood, A. Kamakshidasan, and V. Natarajan. Edit distance between merge trees. *IEEE TVCG*, 2020.

[107] A. Szymczak. Hierarchy of stable Morse decompositions. *IEEE TVCG*, 2013.

[108] S. Tarasov and M. Vyali. Construction of contour trees in 3d in o(n log n) steps. In *SoCG*, 1998.

[109] R. Taylor, A. Chourasia, D. Whalen, and M. L. Norman. The IEEE SciVis Contest. http://sciviscontest.ieeevis.org/2008/, 2008.

[110] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. The Topology ToolKit. *IEEE TVCG (IEEE VIS)*, 2017. https://topology-tool-kit.github.io/.

[111] J. Tierny, A. Gyulassy, E. Simon, and V. Pascucci. Loop surgery for volumetric meshes: Reeb graphs reduced to contour trees. *IEEE TVCG (IEEE VIS)*, 2009.

[112] K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer. Fréchet Means for Distributions of Persistence Diagrams. *DCG*, 2014.

[113] J. Vidal, J. Budin, and J. Tierny. Progressive Wasserstein Barycenters of Persistence Diagrams. *IEEE TVCG (IEEE VIS)*, 2019.

[114] W. Wang, C. Bruyere, B. Kuo, and T. Scheitlin. The IEEE SciVis Contest. http://sciviscontest.ieeevis.org/2004/, 2004.

[115] 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.

[116] T. Wischgoll, A. Chourasia, K. Gorges, M. Bruck, and N. Rober. The IEEE SciVis Contest. http://sciviscontest.ieeevis.org/2017/, 2017.

[117] K. Wu and S. Zhang. A contour tree based visualization for exploring data with uncertainty. *IJUQ*, 2013.

[118] 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.

[119] L. Yan, Y. Wang, E. Munch, E. Gasparovic, and B. Wang. Source Code for a Structural Average of Labeled Merge Trees for Uncertainty Visualization. https://github.com/tdavislab/amt, 2019.

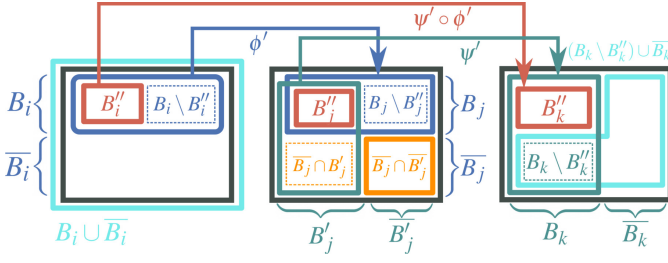[120] L. Yan, Y. Wang, E. Munch, E. Gasparovic, and B. Wang. A structural

Fig. 16. The composition $\psi' \circ \phi'$ of two rooted isomorphisms ϕ' (blue) and ψ' (green) is itself a rooted isomorphism (red). In this schematic view, the involved subtrees are represented as squares.

average of labeled merge trees for uncertainty visualization. *IEEE TVCG (IEEE VIS)*, 2019.

[121] K. Zhang. A Constrained Edit Distance Between Unordered Labeled Trees. *Algorithmica*, 1996.

APPENDIX

———————————————— ◆ ————————————————

1 THE EDIT DISTANCE BETWEEN MERGE TREES [106]

This section formalizes the edit distance introduced by Sridharamurthy et al. [106] (Sec. 1.1) and discusses some of its technical aspects which make it not conducive to interpolation-based geodesics (Sec. 1.2).

1.1 Definition

The edit distance between two merge trees $\mathscr{T}(f_i)$ and $\mathscr{T}(f_j)$, noted $D_E\big(\mathscr{T}(f_i),\mathscr{T}(f_j)\big)$, is defined as follows. Let N_i be a subset of the nodes of $\mathscr{T}(f_i)$ and $\overline{N_i}$ its complement. Let ϕ''' be a *partial* assignment between N_i and a subset N_j of the nodes of $\mathscr{T}(f_j)$ (with complement $\overline{N_j}$). Then $D_E\big(\mathscr{T}(f_i),\mathscr{T}(f_j)\big)$ is given by:

$$D_E\big(\mathscr{T}(f_i),\mathscr{T}(f_j)\big) = \min_{(\phi''',\overline{N_i},\overline{N_j})\in\Phi'''}\Big(\sum_{n_i\in N_i}\gamma\big(n_i\to\phi'''(n_i)\big) \tag{15}$$

$$+ \sum_{n_i\in\overline{N_i}}\gamma(n_i\to\emptyset) \tag{16}$$

$$+ \sum_{n_j\in\overline{N_j}}\gamma(\emptyset\to n_j)\Big) \tag{17}$$

where Φ''' is the space of *constrained* partial assignments (i.e. ϕ''' maps disjoint subtrees of $\mathscr{T}(f_i)$ to disjoint subtrees of $\mathscr{T}(f_j)$) and where γ refers to the cost for: *(i)* mapping a node $n_i \in \mathscr{T}(f_i)$ to a node $\phi''(n_i) = n_j \in \mathscr{T}(f_j)$ (line 15), *(ii)* deleting a node $n_i \in \mathscr{T}(f_i)$ (line 16) and *(iii)* creating a node $n_j \in \mathscr{T}(f_j)$ (line 17), $\emptyset$ being the empty tree.

Zhang [121] introduced a polynomial time algorithm for computing a constrained sequence of edit operations with minimal edit distance (Eq. 15), and showed that the resulting distance is indeed a metric if each cost γ for the above three edit operations is itself a metric (non-negativity, identity, symmetry, triangle inequality). Sridharamurthy et al. [106] exploited this property to introduce their metric, by defining the following distance-based cost model, where p_i and p_j stand for the persistence pairs *containing* the nodes $n_i \in \mathscr{T}(f_i)$ and $n_j \in \mathscr{T}(f_j)$:

$$\gamma(n_i \to n_j) = \min\big(d_\infty(p_i,p_j),\gamma(n_i\to\emptyset)+\gamma(\emptyset\to n_j)\big)$$
$$\gamma(n_i \to \emptyset) = d_\infty\big(p_i,\Delta(p_i)\big)$$
$$\gamma(\emptyset \to n_j) = d_\infty\big(\Delta(p_j),p_j\big).$$

In our work, we introduce an alternative edit distance which further adheres to the L^2-Wasserstein distance between persistence diagrams.

1.2 Interpolation

As shown in the main manuscript (Fig. 4), the linear interpolation of D_E's matchings does not describe a shortest path (i.e. it generates inaccurate midpoints). A key technical reason for this is that D_E involves assignments between *nodes* (of the input merge trees) and not *persistence pairs*. This has several consequences. First, given two input trees $\mathscr{T}(f_i)$ and $\mathscr{T}(f_j)$, D_E's matchings may assign a saddle node in $\mathscr{T}(f_i)$ to an extremum node in $\mathscr{T}(f_j)$, resulting in inconsistent interpolations in the data (from a valley to a peak). Second, D_E's matchings can possibly assign two nodes in $\mathscr{T}(f_i)$ belonging to a *single* persistence pair of f_i to nodes in $\mathscr{T}(f_j)$ belonging to *distinct* persistence pairs in f_j. This second phenomenon further challenges interpolation-based geodesics.

2 $W_2^{\mathscr{T}}$ IS A METRIC

As further described in the main manuscript, given two merge trees $\mathscr{T}(f_i)$ and $\mathscr{T}(f_j)$ and their branch decomposition trees (BDTs) $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$, the dissimilarity measure $W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big)$ is given by:

$$W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big) = \min_{(\phi',\overline{B_i},\overline{B_j})\in\Phi'}\Big(\sum_{b_i\in B_i}\gamma\big(b_i\to\phi'(b_i)\big)^2$$

$$+ \sum_{b_i\in\overline{B_i}}\gamma(b_i\to\emptyset)^2 \tag{18}$$

$$+ \sum_{b_j\in\overline{B_j}}\gamma(\emptyset\to b_j)^2\Big)^{1/2} \tag{19}$$

where ϕ' is an isomorphism between $B_i \subseteq \mathscr{B}(f_i)$ and $B_j \subseteq \mathscr{B}(f_j)$.

In this section, we argue that $W_2^{\mathscr{T}}$ is a metric.
$W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big)$ is always non-negative (the costs γ are squared).
$W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big)$ is symmetric (destruction and creation costs are symmetric, lines 18 and 19).
$W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big) = 0$ if and only if all costs $\gamma = 0$, which only happens if $\mathscr{B}(f_i) = \mathscr{B}(f_j)$ (the identity is included in Φ').
We now argue that $W_2^{\mathscr{T}}$ preserves the triangle inequality, given three trees $\mathscr{B}(f_i)$, $\mathscr{B}(f_j)$ and $\mathscr{B}(f_k)$. For this, we follow a classical approach which we detail here for the sake of completeness. First, we argue that a composition of (optimal) partial rooted isomorphisms (from $\mathscr{B}(f_i)$ to $\mathscr{B}(f_j)$, then from $\mathscr{B}(f_j)$ to $\mathscr{B}(f_k)$) is itself a valid partial rooted isomorphism (and hence belong to our solution space Φ') and that its associated cost consequently bounds by above $W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_k)\big)$ (Eq. 20). Second, we argue that this associated cost is itself bounded by above by $W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big)+W_2^{\mathscr{T}}\big(\mathscr{B}(f_j),\mathscr{B}(f_k)\big)$.

Let $(\phi',\overline{B_i},\overline{B_j})$ be the optimal solution of the partial assignment problem between $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$. ϕ' is a *rooted* isomorphism (i.e. an isomorphism between rooted subtrees) between a subtree B_i of $\mathscr{B}(f_i)$ and a subtree B_j of $\mathscr{B}(f_j)$ (blue, Fig. 16). Equivalently, ϕ' can also be interpreted as a bijection between the *arcs* of B_i and those of B_j.

Let $(\psi',\overline{B_j'},\overline{B_k})$ be the optimal solution of the partial assignment problem between $\mathscr{B}(f_j)$ and $\mathscr{B}(f_k)$. ψ' is a rooted isomorphism between a subtree B_j' of $\mathscr{B}(f_j)$ and a subtree B_k of $\mathscr{B}(f_k)$ (green, Fig. 16).

Let B_j'' be the set of nodes of $\mathscr{B}(f_j)$ involved in *both* ϕ' and ψ' ($B_j'' = B_j \cap B_j'$, in red in Fig. 16, center). Let B_i'' be their pre-image by ϕ' ($B_i'' = \phi'^{-1}(B_j'')$, in red in Fig. 16, left) and B_k'' their image by ψ' ($B_k'' = \psi'(B_j'')$, in red in Fig. 16, right).

Since both ϕ' and ψ' are rooted isomorphisms, their composition $\psi' \circ \phi'$ is also a (rooted) isomorphism between the subtrees B_i'' of $\mathscr{B}(f_i)$ and B_k'' of $\mathscr{B}(f_k)$ (equivalently, it is a bijection between the arcs of B_i'' and the arcs of B_k''). Then $(\psi' \circ \phi',\overline{B_i''},\overline{B_j''})$ is itself a rooted partial isomorphism and belongs to Φ'.

Then, it follows that:

$$W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_k)\big) \leq \Big(\sum_{b\in B_i''}\gamma\big(b\to\psi'\circ\phi'(b)\big)^2 \tag{20}$$

$$+ \sum_{b\in B_i''}\gamma(b\to\emptyset)^2$$

$$+ \sum_{b\in B_k''}\gamma(\emptyset\to b)^2\Big)^{1/2}.$$

Now, let U, V, W be scalar functions on the nodes of the set $B_{ik} = B_i \cup \overline{B_i} \cup (B_k\setminus B_k'')\cup\overline{B_k}$ (cyan subset, Fig. 16) such that:

$$U(b) = \begin{cases} \gamma\big(b\to\psi'\circ\phi'(b)\big) & \text{for } b\in B_i''=\phi'^{-1}(B_j'') \\ \gamma(b\to\emptyset) & \text{for } b\in B_i\setminus B_i'' \\ \gamma(b\to\emptyset) & \text{for } b\in\overline{B_i} \\ \gamma(\emptyset\to b) & \text{for } b\in\overline{B_k} \\ \gamma(\emptyset\to b) & \text{for } b\in B_k\setminus B_k''=\psi'(\overline{B_j}\cap B_j'). \end{cases} \tag{21}$$

U describes all the possible individual costs involved in the composition $\psi' \circ \phi'$. In particular, we can re-write Eq. 20 as:

$$W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_k)\big) \leq \|U\|_2 = \big(\sum_{b\in B_{ik}}U(b)^2\big)^{1/2}. \tag{22}$$

$$V(b) = \begin{cases} \gamma\big(b\to\phi'(b)\big) & \text{for } b\in B_i''=\phi'^{-1}(B_j'') \\ \gamma\big(b\to\phi'(b)\big) & \text{for } b\in B_i\setminus B_i'' \\ \gamma(b\to\emptyset) & \text{for } b\in\overline{B_i} \\ 0 & \text{for } b\in\overline{B_k} \\ \gamma\big(\psi'^{-1}(b)\to\emptyset\big) & \text{for } b\in B_k\setminus B_k''=\psi'(\overline{B_j}\cap B_j') \end{cases} \tag{23}$$

V describes a *subset* of the individual costs involved in the optimal rooted partial isomorphism ϕ'. In particular, only the costs involving $\overline{B_j} \cap \overline{B'_j}$ (orange square, Fig. 16, middle) are excluded. Thus, we have:

$$W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big) \geq \quad \|V\|_2 \quad = \big(\sum_{b \in B_{ik}} V(b)^2\big)^{1/2}. \tag{24}$$

$$W(b) = \begin{cases} \gamma\big(\phi'(b) \to \psi' \circ \phi'(b)\big) & \text{for } b \in B''_i = \phi'^{-1}(B_j'') \\ \gamma\big(\phi'(b) \to \emptyset\big) & \text{for } b \in B_i \setminus B''_i \\ 0 & \text{for } b \in \overline{B_i} \\ \gamma(b \to \emptyset) & \text{for } b \in \overline{B_k} \\ \gamma\big(\psi'^{-1}(b) \to b\big) & \text{for } b \in B_k \setminus B''_k = \psi'(\overline{B_j} \cap B'_j) \end{cases} \tag{25}$$

Similarly to V, W describes a *subset* of the individual costs involved in the optimal rooted partial isomorphism ψ'. In particular, only the costs involving B''_k (red square, Fig. 16, right) are excluded. Thus:

$$W_2^{\mathscr{T}}\big(\mathscr{B}(f_j),\mathscr{B}(f_k)\big) \geq \quad \|W\|_2 \quad = \big(\sum_{b \in B_{ik}} W(b)^2\big)^{1/2}. \tag{26}$$

Now, since γ is defined by the Euclidean distance (Equation 9 of the main manuscript), we have for each node $b \in B_{ik}$:

$$0 \leq U(b) \leq V(b) + W(b).$$

This can be verified by comparing the ith line of Eq. 21 to the sum of the ith lines of Eq. 23 and Eq. 25. Then, we have:

$$\|U\|_2 \leq \|V+W\|_2. \tag{27}$$

Now, since the L^2 norm between vectors respects itself the triangle inequality, we have the following inequality:

$$\|V+W\|_2 \leq \|V\|_2 + \|W\|_2. \tag{28}$$

Then, by combining equations 22, 27, 28, 24, and 26, it follows that:

$$\begin{aligned} W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_k)\big) &\leq \|U\|_2 \leq \|V+W\|_2 \leq \|V\|_2 + \|W\|_2 \\ &\leq W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big) + W_2^{\mathscr{T}}\big(\mathscr{B}(f_j),\mathscr{B}(f_k)\big) \end{aligned}$$

which concludes the proof.

3 COMPARISON TO THE EDIT DISTANCE ALGORITHM [121]

In addition to considering squared costs in our edit distance (equations of the section 3.3 of the main manuscript), our algorithm for the exploration of the search space indeed simplifies the approach by Zhang [121] (used by Sridharamurthy et al. [106]), as our search space is significantly more constrained.

First, since our solution space only considers partial isomorphisms between rooted subtrees, this implies that the destruction of a node (a branch) $b_j \in \mathscr{B}(f_j)$ necessarily implies the destruction of its subtrees, i.e. of its forest $\mathscr{F}(f_j, b_j)$. Thus, the admissible solutions in [106, 121] consisting in deleting b_j and mapping a subtree $\mathscr{B}(f_i, b_i)$ to one of the subtrees of b_j in the forest $\mathscr{F}(f_j, b_j)$ are no longer admissible given our overall solution space Φ'. The removal of such solutions drastically simplifies the evaluation of the distance between subtrees (being the minimum of three solutions in [106], Eq. 12) to the Equation 11 of our main manuscript (containing only one expression to evaluate).

Second, our solution space (rooted partial isomorphisms) also implies that the nodes of $\mathscr{B}(f_i)$ can only be assigned to nodes with the same *depth* in $\mathscr{B}(f_j)$. This further implies that the distance between subtrees (Equation 11 of the main manuscript) only needs to be evaluated for subtrees rooted at nodes of identical depth (see Fig. 6 of the main manuscript).

Together, these two simplifications (*(i)* simpler subtree distance and *(ii)* distance evaluation restricted to subtrees of identical depth from the root) are the key adaptations of Zhang's algorithm [121] that are required for the exploration of our (more constrained) solution space.

4 PARALLEL COMPUTATION OF $W_2^{\mathscr{T}}$

In our work, we express the computation of $W_2^{\mathscr{T}}$ in terms of *tasks*, to leverage task-based shared memory parallelism. First, the Equation 10 of the main manuscript is evaluated. For this, we initiate a task at each leaf of $\mathscr{B}(f_i)$. If a task is the last one to compute among all the direct children of a node $b \in \mathscr{B}(f_i)$, it is then authorized to continue and estimate Equation 10 in b. Atomic counters in b are implemented (and atomically incremented by the task of each child) to determine which child task is the last one to complete, which enables an efficient lightweight synchronization (Fig. 6 of the main manuscript). Overall, Equation 10 is completely estimated with this strategy in a bottom-up fashion. Second, Equation 11 (main manuscript) is evaluated similarly, by initiating a task at each leaf b_i

of $\mathscr{B}(f_i)$. In particular, this task will evaluate Equation 11 given b_i against all subtrees of $\mathscr{B}(f_j)$ of identical depth (again using independent tasks initiated at the leaves of $\mathscr{B}(f_j)$, see Fig. 6). Similarly to Equation 10, we employ the same lightweight synchronization mechanism based on atomic counters to continue a task over to its parent only when it is the last child task reaching it. Thus, in both cases (Eqs. 10 and 11), the number of parallel tasks is initially bounded by the number of leaves in $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$ (which is typically much larger than the number of cores) and progressively decreases during the computation.

5 $\mathscr{B}_\alpha(f_i \to f_j)$ IS A GEODESIC FOR $0 \leq \alpha \leq 1$

We now argue that $W_2^{\mathscr{T}}$ defines a geodesic space. For this, for any two BDTs $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$, we describe the existence of a path between them whose length is equal to $W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big)$ (and thus minimal).

Let $P = \big(\mathscr{B}_t\big)_{t \in [0,1]}$ be a path of BDTs parameterized by t.

We recall that the length $\mathscr{L}(P)$ of P is given by:

$$\mathscr{L}(P) = \sup_{n;0 \leq t_0 \leq t_1 \leq \cdots \leq t_n = 1} \sum_{k=0}^{n-1} W_2^{\mathscr{T}}\big(\mathscr{B}_{t_k},\mathscr{B}_{t_{k+1}}\big).$$

Now, let P_α be the path corresponding to the interpolation between $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$, as defined in section 4.1 of the main manuscript. We now argue that $\mathscr{L}(P_\alpha) = W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big)$.

Let $(\phi', \overline{B_i}, \overline{B_j})$ be the optimal rooted partial isomorphism between $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$. Moreover, let $\mathscr{B}_s(f_i \to f_j)$ and $\mathscr{B}_t(f_i \to f_j)$ be two interpolated trees obtained respectively with $\alpha = s$ and $\alpha = t$, given $0 \leq s \leq t \leq 1$. We will note ϕ'_s the application on $B_i \cup \overline{B_i} \cup \overline{B_j}$ defined by interpolation (section 4.1 of the main manuscript):

$$\phi'_s(b) = \begin{cases} (1-s)b + s\phi'(b) & \text{for } b \in B_i \\ (1-s)b + s\Delta(b) & \text{for } b \in \overline{B_i} \\ sb + (1-s)\Delta(b) & \text{for } b \in \overline{B_j}. \end{cases} \tag{29}$$

ϕ'_t is defined similarly for t. Then, as discussed in Sec. 2 of this appendix, since the composition of partial rooted isomorphisms is itself a partial rooted isomorphism, the composition $\phi'_t \circ \phi'^{-1}_s$ (which goes from $\mathscr{B}_s(f_i \to f_j)$ to $\mathscr{B}(f_i)$ and then from $\mathscr{B}(f_i)$ to $\mathscr{B}_t(f_i \to f_j)$) does define a valid partial rooted isomorphism between $\mathscr{B}_s(f_i \to f_j)$ and $\mathscr{B}_t(f_i \to f_j)$ and we have:

$$\begin{aligned} W_2^{\mathscr{T}}\big(\mathscr{B}_s(f_i \to f_j),\mathscr{B}_t(f_i \to f_j)\big) &\leq \\ &\Big(\sum_{b \in B_i \cup \overline{B_i} \cup \overline{B_j}} \gamma\big(\phi'_s(b) \to \phi'_t(b)\big)^2\Big)^{1/2} \end{aligned}$$

and we also have by definition of ϕ'_s and ϕ'_t (Eq. 29):

$$\Big(\sum_{b \in B_i \cup \overline{B_i} \cup \overline{B_j}} \gamma\big(\phi'_s(b) \to \phi'_t(b)\big)^2\Big)^{1/2} = (t-s)W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big).$$

Now, given the triangle inequality on the path P_α, we have:

$$\begin{aligned} W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big) &\leq \\ & W_2^{\mathscr{T}}\big(\mathscr{B}_0(f_i \to f_j),\mathscr{B}_s(f_i \to f_j)\big) \\ &+ W_2^{\mathscr{T}}\big(\mathscr{B}_s(f_i \to f_j),\mathscr{B}_t(f_i \to f_j)\big) \\ &+ W_2^{\mathscr{T}}\big(\mathscr{B}_t(f_i \to f_j),\mathscr{B}_1(f_i \to f_j)\big) \\ &\leq \big(s + (t-s) + (1-t)\big)W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big) \\ &= W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big). \end{aligned}$$

If follows that the above inequalities are in fact equalities and we have:

$$W_2^{\mathscr{T}}\big(\mathscr{B}_s(f_i \to f_j),\mathscr{B}_t(f_i \to f_j)\big) = (t-s)W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big).$$

Then, for *any* subdivision $0 = t_0 \leq t_1 \leq \cdots \leq t_n = 1$ of P_α, we have:

$$\begin{aligned} \sum_{k=0}^{n-1} W_2^{\mathscr{T}}\big(\mathscr{B}_{t_k}(f_i \to f_j),\mathscr{B}_{t_{k+1}}(f_i \to f_j)\big) \\ = \sum_{k=0}^{n-1} (t_{k+1} - t_k)W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big) \\ = (t_n - t_0)W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big) \\ = W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big). \end{aligned}$$

Thus $\mathscr{L}(P_\alpha) = W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big)$.

Hence the space of merge trees equipped with $W_2^{\mathscr{T}}$ is a geodesic space, and $\mathscr{B}_\alpha(f_i \to f_j)$ constructs paths of minimal length on it.

6 $W_2^{\mathscr{T}}$ WITH NORMALIZED COSTS DEFINES A GEODESIC SPACE

Let $W_2^{\mathscr{N}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big)$ be a similarity measure between $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$, defined as:

$$W_2^{\mathscr{N}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big) = W_2^{\mathscr{T}}\big(\mathscr{N}\big(\mathscr{B}(f_i)\big),\mathscr{N}\big(\mathscr{B}(f_j)\big)\big)$$

where $\mathscr{N}$ is the local normalization described in Section 4.2 of the main manuscript. Since $\mathscr{N}$ is invertible, $W_2^{\mathscr{N}}$ inherits all the properties of $W_2^{\mathscr{T}}$ and is also a distance metric.

The *normalized* interpolation $\mathscr{B}_s(f_i \to f_j)$, $s \in [0,1]$ between $\mathscr{B}(f_i)$ and $\mathscr{B}(f_j)$ is defined as the image by $\mathscr{N}^{-1}$ of the interpolation between the normalized trees $\mathscr{N}\big(\mathscr{B}(f_i)\big)$ and $\mathscr{N}\big(\mathscr{B}(f_j)\big)$. Then, given s and t such that $0 \le s \le t \le 1$, it follows that:

$$W_2^{\mathscr{N}}\big(\mathscr{B}_s(f_i \to f_j),\mathscr{B}_t(f_i \to f_j)\big) = (t-s)W_2^{\mathscr{T}}\big(\mathscr{N}\big(\mathscr{B}(f_i)\big),\mathscr{N}\big(\mathscr{B}(f_j)\big)\big)$$
$$= (t-s)W_2^{\mathscr{N}}\big(\mathscr{B}(f_i),\mathscr{B}(f_j)\big)$$

which proves that the space of merge trees equipped with $W_2^{\mathscr{N}}$ is a geodesic space, and that the above normalized interpolation constructs paths of minimal length on it.

7 MINIMIZING THE FRÉCHET ENERGY

The optimization algorithm described in Section 5.2 of the main manuscript constructively decreases the Fréchet energy at each iteration. In particular, once a local minimizer of the Fréchet energy is obtained for a fixed assignment with the *update* step *(ii)*, the subsequent *assignment* step *(i)* does further improve the assignments hence iteratively decreasing the Fréchet energy constructively.

Let F' be a function of an arbitrary BDT $\mathscr{B}$ and of an arbitrary (i.e. not necessarily optimal) set of N rooted partial isomorphisms $(\phi'_i, \overline{B_i}, \overline{B_\mathscr{B}})_{i=1,\dots,N}$ between $\mathscr{B}$ and the N BDTs of $\mathscr{S}_\mathscr{B}$:

$$F'\big(\mathscr{B},(\phi'_i,\overline{B_i},\overline{B_\mathscr{B}})_{i=1,\dots,N}\big) := \sum_{\mathscr{B}(f_i)\in\mathscr{S}_\mathscr{B}}\Big(\sum_{b_i\in B_i}\gamma(b_i\to\phi'_i(b_i))^2 \\ +\sum_{b_i\in\overline{B_i}}\gamma(b_i\to\emptyset)^2 \\ +\sum_{b_\mathscr{B}\in\overline{B_\mathscr{B}}}\gamma(\emptyset\to b_\mathscr{B})^2\Big)^{1/2}.$$

Now, let $\mathscr{B}_k$ be the candidate barycenter at the iteration k of the algorithm and let $(\phi'^k_i,\overline{B_i^k},\overline{B_\mathscr{B}^k})$ be the optimal rooted partial isomorphism between $\mathscr{B}_k$ and $\mathscr{B}(f_i)$, computed by the *assignment* step of the iteration. Then, we have:

$$F'\big(\mathscr{B}_k,(\phi'^k_i,\overline{B_i^k},\overline{B_\mathscr{B}^k})_{i=1,\dots,N}\big) = \sum_{\mathscr{B}(f_i)\in\mathscr{S}_\mathscr{B}}W_2^{\mathscr{T}}\big(\mathscr{B}_k,\mathscr{B}(f_i)\big)^2.$$

Next, the *update* step of the iteration k consists in moving $\mathscr{B}_k$ to $\mathscr{B}_{k+1}$ by placing (in the 2D birth/death space) each branch $b\in\mathscr{B}_k$ at the arithmetic mean of the assignments. Since the arithmetic mean generally minimizes sums of Euclidean distances, we have:

$$F'\big(\mathscr{B}_{k+1},(\phi'^k_i,\overline{B_i^k},\overline{B_\mathscr{B}^k})_{i=1,\dots,N}\big) \le F'\big(\mathscr{B}_k,(\phi'^k_i,\overline{B_i^k},\overline{B_\mathscr{B}^k})_{i=1,\dots,N}\big).$$

Now, observe that since the previous rooted partial isomorphisms are not optimal anymore for $\mathscr{B}_{k+1}$, we also have:

$$\sum_{\mathscr{B}(f_i)\in\mathscr{S}_\mathscr{B}}W_2^{\mathscr{T}}\big(\mathscr{B}_{k+1},\mathscr{B}(f_i)\big)^2 \le F'\big(\mathscr{B}_{k+1},(\phi'^k_i,\overline{B_i^k},\overline{B_\mathscr{B}^k})_{i=1,\dots,N}\big).$$

Once $\mathscr{B}_{k+1}$ is fixed, all the rooted partial isomorphisms are then optimized again with the *assignment* step of the iteration $k+1$ to attain:

$$F'\big(\mathscr{B}_{k+1},(\phi'^{k+1}_i,\overline{B_i^{k+1}},\overline{B_\mathscr{B}^{k+1}})_{i=1,\dots,N}\big) \\ = \sum_{\mathscr{B}(f_i)\in\mathscr{S}_\mathscr{B}}W_2^{\mathscr{T}}\big(\mathscr{B}_{k+1},\mathscr{B}(f_i)\big)^2.$$

The result of these two steps is that:

$$F'\big(\mathscr{B}_{k+1},(\phi'^{k+1}_i,\overline{B_i^{k+1}},\overline{B_\mathscr{B}^{k+1}})_{i=1,\dots,N}\big) \\ \le F'\big(\mathscr{B}_k,(\phi'^k_i,\overline{B_i^k},\overline{B_\mathscr{B}^k})_{i=1,\dots,N}\big) \\ \sum_{\mathscr{B}(f_i)\in\mathscr{S}_\mathscr{B}}W_2^{\mathscr{T}}\big(\mathscr{B}_{k+1},\mathscr{B}(f_i)\big)^2 \le \sum_{\mathscr{B}(f_i)\in\mathscr{S}_\mathscr{B}}W_2^{\mathscr{T}}\big(\mathscr{B}_k,\mathscr{B}(f_i)\big)^2.$$

Then, each iteration of our algorithm indeed decreases the Fréchet energy. Since there is a finite number of combinations of rooted partial isomorphisms between the barycenter and the N input trees $\mathscr{B}(f_i)$, it follows that the algorithm converges, in a finite number of steps, to a local minimum $\mathscr{B}^*$ of the Fréchet energy (if multiple, equally valued, optimal sets of assignments exist between $\mathscr{B}^*$ and $\mathscr{S}_\mathscr{B}$, each one needs to be explored with the update step of our algorithm). In practice, as described in the manuscript, we stop our algorithm when the Fréchet energy has decreased by less than 1% between consecutive iterations.
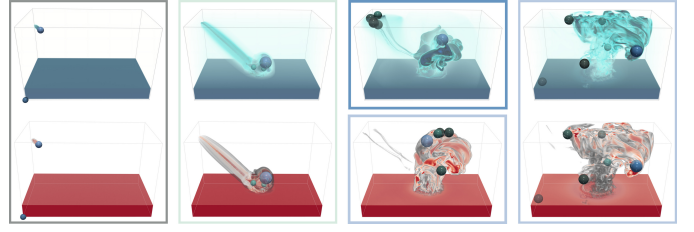

Fig. 17. Comparison between the *key frames* identified by our temporal reduction algorithm, with regard to $W_2^{\mathscr{T}}$ (blue, top) and with regard to $W_2^{\mathscr{D}}$ (red, bottom). By construction, the reduction algorithm identifies as key frames the first and last time steps, irrespective of the employed metric.

8 TEMPORAL REDUCTION ALGORITHM

Let $\mathscr{S} = \{\mathscr{B}(f_1),\mathscr{B}(f_2),\dots,\mathscr{B}(f_N)\}$ be the input temporal sequence of BDTs (we assume a regular temporal sampling). Let $\mathscr{K} \subseteq \mathscr{S}$ be a set of key frames. Let $\mathscr{S}' = \{\mathscr{B}'(f_1),\mathscr{B}'(f_2),\dots,\mathscr{B}'(f_N)\}$ be a *reduced* temporal sequence, where:

$$\mathscr{B}'(f_i) = (1-\alpha_i)\mathscr{B}(f_j) + \alpha_i\mathscr{B}(f_k) \tag{30}$$

where $\mathscr{B}(f_j)$ and $\mathscr{B}(f_k)$ are two consecutive trees in $\mathscr{K}$, such that $j \le i \le k$ and $\alpha_i = (i-j)/(k-j)$. $\mathscr{B}'(f_i)$ is then on a geodesic between $\mathscr{B}(f_j)$ and $\mathscr{B}(f_k)$. We introduce the following distance between the temporal sequences $\mathscr{S}$ and $\mathscr{S}'$:

$$d_S(\mathscr{S},\mathscr{S}') = \Big(\sum_{i=0}^N W_2^{\mathscr{T}}\big(\mathscr{B}(f_i),\mathscr{B}'(f_i)\big)^2\Big)^{1/2}. \tag{31}$$

d_S is indeed a metric since it is a composition of metrics (being the L^2 norm between vectors of BDTs under the metric $W_2^{\mathscr{T}}$).

Our algorithm for temporal reduction consists in initializing $\mathscr{K}$ with the entire input sequence ($\mathscr{K} \leftarrow \mathscr{S}$) and then removing greedily, at each iteration, the tree $\mathscr{B}^*$ from $\mathscr{K}$ ($\mathscr{K} \leftarrow \mathscr{K} - \{\mathscr{B}^*\}$) which minimizes $d_S(\mathscr{S},\mathscr{S}')$, and which, hence, better preserves the input sequence, until $\mathscr{K}$ reaches a target size.

Fig. 17 shows the temporal reduction performed by this algorithm on the *Asteroid impact* sequence (see Section 9.1). This figure illustrates *key frames*, which correspond to time steps for which $\mathscr{B}(f_i) = \mathscr{B}'(f_i)$: these are the time steps which have *not* been removed from the sequence through the reduction. In particular, this figure compares the usage of two metrics in the reduction algorithm: $W_2^{\mathscr{T}}$ (blue, top) and $W_2^{\mathscr{D}}$ (red, bottom, obtained with $\varepsilon_1 = 1$). By construction, since our reduction algorithm is based on interpolation only, the first and last BDTs in the sequence $\mathscr{S}$ are always kept in the reduced sequence $\mathscr{S}'$. In other words, the first (leftmost, Fig. 17) and last (rightmost, Fig. 17) time steps are always identified as key frames, irrespective of the employed metric. For this specific example, the second key frame (second from left, Fig. 17) also happens to be identical for both metrics. In contrast to the sequence extremities, the common identification of this time step as a key frame by $W_2^{\mathscr{T}}$ and $W_2^{\mathscr{D}}$ is not obtained by construction: the reduction algorithm did select this key frame in both configurations. Then, only the third key frame (third from left, Fig. 17) is different in this example. In particular, when using $W_2^{\mathscr{T}}$, the reduction algorithm identifies one key frame per key phase of the simulation (see Section 9.1, each key phase is represented in Fig. 17 with a frame of distinct color). In contrast, when using $W_2^{\mathscr{D}}$, the third key frame belongs to the same key phase as the last key frame (*"Aftermath"*, light blue frame). Then the reduction driven by $W_2^{\mathscr{D}}$ fails at identifying a key frame for the third key phase (*"Impact"*, dark blue frame). This is confirmed visually in Fig. 17, as the third key frame identified with $W_2^{\mathscr{T}}$ (in blue) seems to represent an intermediate step in the simulation between the second and fourth key frames. In contrast, the third key frame identified by the reduction with $W_2^{\mathscr{D}}$ (in red) is more visually similar to the fourth key frame, and hence possibly more redundant.

9 DATA SPECIFICATION

This section provides a complete specification of the ensemble datasets used in the paper. In particular, we document the data provenance, its representation, its pre-processing when applicable, and we specify the associated ground-truth classification.

All of these ensemble datasets were extracted from public repositories. We additionally provide a set of scripts which automatically download all of these datasets (at the exception of *Asteroid impact* and *Cloud processes*, for which the dataset providers need to be contacted personally), pre-process them with TTK and output them in VTK file format, with the ground-truth classification attached to the files as meta-data (i.e. *"Field Data"* in the VTK terminology). For convenience, we also provide an archive containing the entire curated ensemble datasets (in VTK file format). All of this new material (scripts and curated data) is located at the following address: https://github.com/MatPont/WassersteinMergeTreesData.

Moreover, we also provide in additional material all the ensembles of merge trees computed from these datasets (in the code archive containing the implementation of our method).

9.1 Asteroid impact

This ensemble is composed of 7 members, given as 3D regular grids (sampled at $300 \times 300 \times 300$, implicitly triangulated by TTK). It has been made available in the context of the SciVis contest 2018 [84]. Each member corresponds to the last time step of the simulation of the impact of an asteroid with the sea at the surface of the Earth, for two configurations of asteroid diameter. The considered scalar field is the matter density, which is one of the variables of the simulation which discriminates well the asteroid from the water and the ambient air. This ensemble corresponds to a parameter study (in this case, studying the effect of the asteroid's diameter on the resulting wave), which is a typical task in numerical simulation. In this application, salient maxima capture well the asteroid and large water splashes. Thus, each member is represented by the split tree (capturing maxima). The associated ground-truth classification assigns members computed with similar asteroid diameters to the same class. Thus, the corresponding classification task consists in identifying, for a given member, its correct asteroid diameter class. The ground-truth classification is as follows:

- **Class 1** (3 members): yA11, yB11, yC11.

- **Class 2** (4 members): yA31, yA32, yB31, yC31

Another selection of the original data has been used for the evaluation of our temporal reduction framework (Fig. 10 of the main manuscript). For this experiment, we used the asteroid diameter configuration *"yA31"* and considered the following time steps, organized in 4 phases (according to the SciVis contest companion documentation [84]):

- **Phase 1, initial state** (5 time steps): 01141, 03429, 05700, 07920, 09782

- **Phase 2, approach** (5 time steps): 13306, 16317, 18124, 19599, 21255

- **Phase 3, impact** (5 time steps): 28649, 31737, 34654, 37273, 39476

- **Phase 4, aftermath** (5 time steps): 44229, 45793, 47190, 48557, 49978

9.2 Cloud processes

This ensemble is composed of 12 members, given as 2D regular grids (sampled at 1430×1557, implicitly triangulated by TTK). Each member corresponds to a time step of the simulation of cloud formations [116]. For this application, large clouds are well captured by the maxima of the pressure variable (pre-processed with 10 iterations of smoothing). Thus, split trees (capturing maxima) are considered for this ensemble. The associated ground-truth classification assigns each time step to one of the three key phases of the simulation. The corresponding classification task therefore consists in identifying, for each time step, to which phase it belongs. The ground-truth classification is as follows:

- **Class 1** (4 members): 0, 5, 10, 15

- **Class 2** (4 members): 500, 505, 510, 515

- **Class 3** (4 members): 1000, 1005, 1010, 1015

9.3 Viscous fingering

This ensemble is composed of 15 members, given as 3-dimensional point clouds (representing a particle-based flow simulation). Each point cloud is turned into a Eulerian representation of the variables by using the *"Gaussian Resampling"* filter of ParaView, effectively transforming, via interpolation [101], each ensemble member into a 3D regular grid (sampled at $50 \times 50 \times 50$, implicitly triangulated by TTK). The original data has been made available in the context of the SciVis contest 2016 [45]. Each member corresponds to the last time step of the simulation of a viscous fingering phenomenon, occurring when dissolving salt in water. The considered scalar field is the salt concentration, whose salient maxima capture well the most prominent fingers. Thus, each member is represented by the split tree (capturing maxima). Given the studied physical phenomenon, the simulation approach is not deterministic, resulting in distinct outputs for identical initial configurations. In this application, three distinct solver resolutions have been considered, corresponding to three distinct numbers of particles (resolution code 20: 194k particles, resolution code 30: 544k particles, resolution code 44: 1.7M particles). Thus, this ensemble corresponds to a parameter study (in this case, studying the effect of the input resolution on the output fingering), which is a typical task in numerical simulation. The associated ground-truth classification assigns members with the same input resolution to the same class. Thus, the corresponding classification task consists in identifying, for a given ensemble member, its corresponding particle count. The ground-truth classification is as follows:

- **Class 1, resolution 20** (5 members): 20run1, 20run3, 20run4, 20run5, 20run6.

- **Class 2, resolution 30** (5 members): 30run1, 30run2, 30run3, 30run4, 30run5

- **Class 3, resolution 44** (5 members): 30run1, 30run2, 30run3, 30run4, 30run5

9.4 Dark matter

This ensemble is composed of 40 members, given as 3-dimensional point clouds (representing a particle-based simulation). Each point cloud is turned into a Eulerian representation of the variables by using the *"Gaussian Resampling"* filter of ParaView, effectively transforming, via interpolation [101], each ensemble member into a 3D regular grid (sampled at $100 \times 100 \times 100$, implicitly triangulated by TTK). The original data has been made available in the context of the SciVis contest 2015 [56]. Each member corresponds to a time step of a simulation of the universe formation, where regions of high concentration of dark matter form a filament structure known as the *cosmic web*. The considered scalar field is therefore dark matter density, whose salient maxima capture well large clusters of galaxies. Thus, each member is represented by the split tree (capturing maxima). The associated ground-truth classification assigns each time step to one of the four key phases of the simulation. The corresponding classification task therefore consists in identifying, for each time step, to which phase it belongs. The ground-truth classification is as follows:

- **Class 1** (10 members): 0.0200, 0.0300, 0.0400, 0.0500, 0.0600, 0.0700, 0.0800, 0.0900, 0.1000, 0.1100

- **Class 2** (10 members): 0.2700, 0.2800, 0.2900, 0.3000, 0.3100, 0.3200, 0.3300, 0.3400, 0.3500, 0.3600

- **Class 3** (10 members): 0.5900, 0.6000, 0.6100, 0.6200, 0.6300, 0.6400, 0.6500, 0.6600, 0.6700, 0.6800

- **Class 4** (10 members): 0.9100, 0.9200, 0.9300, 0.9400, 0.9500, 0.9600, 0.9700, 0.9800, 0.9900, 1.0000

9.5 Volcanic eruptions

This ensemble is composed of 12 members, given as 2D regular grids (sampled at 500×500, implicitly triangulated by TTK). Each member corresponds to an observation of a volcanic eruption, obtained by satellite imaging (as this data exhibits a bit of noise, it has been pre-simplified by removing all saddle-maxima pairs with a persistence lower than 0.5% of the data range). The original data has been made available in the context of the SciVis contest 2014 [41]. The considered scalar field is the sulfur dioxide concentration, for which salient maxima correspond to volcanic eruptions. Thus, each observation is represented by the split tree (capturing maxima). Each member corresponds to a specific acquisition period, itself corresponding to the eruption of one particular volcano at the surface of the Earth. The associated ground-truth classification assigns observations acquired in the same period of time to the same class. The corresponding classification task therefore consists in identifying, for each observation (taken at a specified date), the erupting volcano it corresponds to. The ground-truth classification is as follows:

- **Class 1** (4 members): 150_am, 150_pm, 151_am, 151_pm

- **Class 2** (4 members): 156_am, 156_pm, 157_am, 157_pm

- **Class 3** (4 members): 164_am, 164_pm, 165_am, 165_pm

9.6 Ionization front (3D)

This ensemble is composed of 16 members, given as 3D regular grids (sampled at $300 \times 124 \times 124$, implicitly triangulated by TTK). Each member corresponds to a time step of a simulation of ionization front propagation [109]. For this application, large ionization flares are well captured by salient maxima of the ion concentration. Thus, split trees (capturing maxima) are considered for this ensemble. The associated ground-truth classification assigns each time step to one of the four key phases of the simulation. The corresponding classification task therefore consists in identifying, for each time step, to which phase it belongs. The ground-truth classification is as follows:

- **Class 1** (4 members): 0025, 0026, 0027, 0028

- **Class 2** (4 members): 0075, 0076, 0077, 0078,

- **Class 3** (4 members): 0125, 0126, 0127, 0128

- **Class 4** (4 members): 0175, 0176, 0177, 0178,

9.7 Ionization front (2D)

This ensemble is a 2D version of the above ensemble, where the dataset providers have selected a 2D slice in the center of the volume (sampled at 600×248). The associated classification task is therefore identical.

9.8 Earthquake

This ensemble is composed of 12 members, given as 3D regular grids (sampled at $375 \times 188 \times 50$, implicitly triangulated by TTK). Each member corresponds to a time step of the simulation of an earthquake at the San Andreas fault [77]. For this application, the shock wave can be tracked with the local maxima of the wave front velocity magnitude (this scalar field is pre-processed to pre-simplify all saddle-maxima pairs with a persistence smaller than 0.05% of the data range). Thus, split trees (capturing maxima) are considered for this ensemble. The associated ground-truth classification assigns each time step to one of the three key phases of the simulation. The corresponding classification task therefore consists in identifying, for each time step, to which phase it belongs. The ground-truth classification is as follows:

- **Class 1** (4 members): 002700, 002900, 003100, 003300

- **Class 2** (4 members): 007700, 007900, 008100, 008300

- **Class 3** (4 members): 011700, 011900, 012100, 012300

9.9 Isabel

This ensemble is composed of 12 members, given as 3D regular grids (sampled at $250 \times 250 \times 50$, implicitly triangulated by TTK). Each member corresponds to a time step of the simulation of the Isabel hurricane [114]. This ensemble has been used in previous work [39, 113] and the corresponding VTK files are available at the following address: `https://github.com/julesvidal/wasserstein-pd-barycenter`. In this application, the eyewall of the hurricane is typically characterized by high wind velocities, well captured by the the maxima of the flow velocity. Thus, split trees (capturing maxima) are considered for this ensemble. The associated ground-truth classification assigns each time step to one of the three key phases (formation, drift, landfall) of the hurricane simulation. The corresponding classification task therefore consists in identifying, for each member, to which key phase it belongs. The ground-truth classification is as follows:

- **Class 1** (4 members): 2, 3, 4, 5

- **Class 2** (4 members): 30, 31, 32, 33

- **Class 3** (4 members): 45, 46, 47, 48

9.10 Starting vortex

This ensemble is composed of 12 members, given as 2D regular grids (sampled at 1500×1000, implicitly triangulated by TTK). It has been generated with the Gerris flow solver [89] and was provided in previous work [39, 113]. It is available at the following address: `https://github.com/julesvidal/wasserstein-pd-barycenter`. The data models flow turbulence behind a wing, for two ranges of wing inclination angles. The considered scalar field is the orthogonal component of the curl of the flow velocity. This ensemble corresponds to a parameter study (in this case, studying the effect of the wing configuration on turbulence), which is a typical task in numerical simulation. In this application, salient extrema are typically considered as reliable estimations of the center of vortices. Thus, each run is represented by two merge trees (the join tree – capturing minima, and the split tree, capturing maxima), which are processed independently by our algorithms. The associated ground-truth classification assigns members computed with similar inclination angles to the same class. The corresponding classification task therefore consists in identifying, for a given ensemble member, its correct wing configuration class. The ground-truth classification is as follows:

- **Class 1** (6 members): Angle=2, Angle=3, Angle=4, Angle=5, Angle=6, Angle=8

- **Class 2** (6 members): Angle=38, Angle=39, Angle=40, Angle=41, Angle=42, Angle=43

9.11 Sea surface height

This ensemble is composed of 48 members, given as 2D regular grids (sampled at 1440×720, implicitly triangulated by TTK). Each member corresponds to an observation of the sea surface height at the surface of the Earth, taken in January, April, July and October 2012. The original data can be found at the following address: `https://ecco.jpl.nasa.gov/products/all/`. This ensemble has been used in previous work [39, 113] and the corresponding VTK files are available at the following address: `https://github.com/julesvidal/wasserstein-pd-barycenter`. In this application, the features of interest are the center of eddies, which can be reliably estimated with height extrema. Thus, each observation is represented by two merge trees (the join tree – capturing minima, and the split tree, capturing maxima), which are processed independently by our algorithms. The associated ground-truth classification assigns observations acquired in the same month to the same class. The corresponding classification task therefore consists in identifying, for each observation (taken at a specified date), the season in which it has been acquired. The ground-truth classification is as follows:

- **Class 1** (12 members): 20120111, 20120115, 20120116, 20120117, 20120118, 20120119, 20120120, 20120121, 20120123, 20120128, 20120129

- **Class 2** (12 members): 20120419, 20120420, 20120421, 20120422, 20120423, 20120424, 20120425, 20120426, 20120427, 20120428, 20120429, 20120430

- **Class 3** (12 members): 20120711, 20120712, 20120713, 20120714, 20120715, 20120716, 20120717, 20120718, 20120719, 20120720, 20120721, 20120722

- **Class 4** (12 members): 20121008, 20121009, 20121010, 20121011, 20121012, 20121016, 20121017, 20121018, 20121019, 20121020, 20121022, 20121023

9.12 Vortex street

This ensemble is composed of 45 members, given as 2D regular grids (sampled at 300×100, implicitly triangulated by TTK). It has been generated with the Gerris flow solver [89] and was provided in previous work [39, 113]. It is available at the following address: `https://github.com/julesvidal/wasserstein-pd-barycenter`. The data models flow turbulence behind an obstacle. The considered scalar field is the orthogonal component of the curl of the flow velocity, for 5 fluids of different viscosity. This ensemble corresponds to a parameter study (in this case, studying the effect of viscosity on turbulence), which is a typical task in numerical simulation. In this application, salient extrema are typically considered as reliable estimations of the center of vortices. Thus, each run is represented by two merge trees (the join tree – capturing minima, and the split tree, capturing maxima), which are processed independently by our algorithms. The associated ground-truth classification assigns members computed with similar viscosities to the same class. The corresponding classification task therefore consists in identifying, for a given ensemble member, its correct viscosity class. The ground-truth classification is as follows:

- **Class 1** (9 members): Viscosity=100.0, Viscosity=100.1, Viscosity=100.2, Viscosity=100.3, Viscosity=100.4, Viscosity=100.5, Viscosity=100.6, Viscosity=100.7, Viscosity=100.9

- **Class 2** (9 members): Viscosity=160.0, Viscosity=160.1, Viscosity=160.2, Viscosity=160.3, Viscosity=160.4, Viscosity=160.5, Viscosity=160.6, Viscosity=160.7, Viscosity=160.8

- **Class 3** (9 members): Viscosity=200.0, Viscosity=200.1, Viscosity=200.2, Viscosity=200.3, Viscosity=200.4, Viscosity=200.5, Viscosity=200.6, Viscosity=200.7, Viscosity=200.8

- **Class 4** (9 members): Viscosity=50.0, Viscosity=50.1, Viscosity=50.2, Viscosity=50.3, Viscosity=50.5, Viscosity=50.6, Viscosity=50.7, Viscosity=50.8, Viscosity=50.9

- **Class 5** (9 members): Viscosity=60.1, Viscosity=60.2, Viscosity=60.3, Viscosity=60.4, Viscosity=60.5, Viscosity=60.6, Viscosity=60.7, Viscosity=60.8, Viscosity=60.9

10 PARAMETER ANALYSIS

In this section, we study the practical effect of the parameters of our approach. In particular, we extend the empirical stability evaluation of our metric with regard to all the parameters of our approach and we illustrate their effect on geodesic computation.

10.1 Interpretation

The first parameter of our approach is $\varepsilon_1 \in [0, 1]$. It dictates the merge of saddles in the input trees, to mitigate saddle swap instabilities, as previously documented by Sridharamurthy et al. [106]. Adjacent saddles in the input trees are merged if their relative difference in scalar value (relative to the largest function difference between adjacent saddles) is *smaller* than ε_1. For $\varepsilon_1 = 0$, no saddle merge is performed whereas for $\varepsilon_1 = 1$, all saddles are merged and $W_2^{\mathcal{T}}$ becomes equivalent to the L^2 Wasserstein distance between persistence diagrams, noted $W_2^{\mathcal{D}}$.

The local normalization step of our framework (Section 4.2 of the main manuscript) guarantees the topological consistency of the interpolated merge trees (Fig. 8 of the main manuscript). However, this normalization shrinks the birth/death values of all the input branches to the interval $[0, 1]$, irrespective of their original persistence. To mitigate this effect, the input BDTs are pre-processed, so that branches with small initial persistence (i.e. *small branches*) are not given too much importance in the metric. In particular, small branches are moved up the input BDT if their persistence relative to their parent is *larger* than $\varepsilon_2 \in [0, 1]$. When $\varepsilon_2 = 0$, all branches are moved up to the root of the BDT and again, $W_2^{\mathcal{T}}$ becomes equivalent to $W_2^{\mathcal{D}}$. When $\varepsilon_2 = 1$, no branch is moved up the BDT and ε_2 has no effect on the outcome (i.e. the input BDT is left unchanged). In practice, we recommend the default value $\varepsilon_2 = 0.95$: if a branch b has a nearly identical persistence to that of its parent b', it is moved higher in the BDT, so that its normalized persistence becomes nearly identical to that of its parent b' (instead of being artificially larger due to the local normalization).

The parameter $\varepsilon_3 \in [0, 1]$ further restricts the application of the above BDT pre-processing, by only considering (for displacement up the BDT) the branches with a relative persistence (with respect to the overall data range) smaller than ε_3. When $\varepsilon_3 = 1$, all branches are subject to the above pre-processing and ε_2 fully dictates the BDT pre-processing. When $\varepsilon_3 = 0$, no branch is moved up the BDT and the two parameters ε_2 and ε_3 have no effect on the outcome. In practice, we recommend the default value $\varepsilon_3 = 0.9$, which prevents the most persistent branches from moving up the BDT.

Overall, when the parameters $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$ are set to the values $(0, 1, 1)$, the input trees are *not* pre-processed by the above procedures (i.e. they are left unchanged) and their structure has a strong impact on $W_2^{\mathcal{T}}$. When $\varepsilon_1 = 1$ or when $\varepsilon_2 = 0$, $W_2^{\mathcal{T}}$ becomes equivalent to $W_2^{\mathcal{D}}$ and the structure of the input trees has no impact anymore on the metric. In-between values balance the importance of the structure of the trees on the metric. We recommend the default values $(0.05, 0.95, 0.9)$, which provides an acceptable stability with regard to saddle swaps (mitigated by ε_1) and which gives a reasonable importance to small branches in the metric (controlled by ε_2 and ε_3, which are dependent parameters).
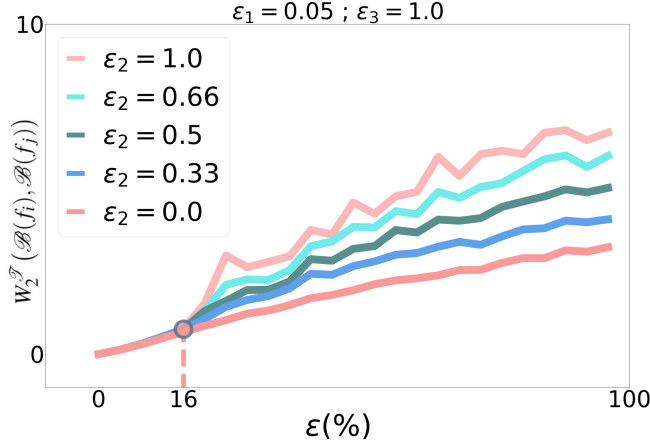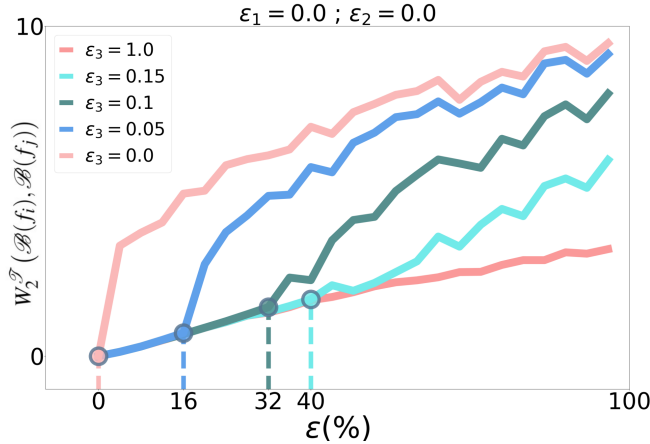
Fig. 18. Empirical stability evaluation with regard to ε_2. Given an input scalar field f_i, a noisy version f_j is created by inserting a random noise of increasing amplitude ε (cf. Figure 14 of the main manuscript). The evolution of $W_2^{\mathscr{T}}\left(\mathscr{B}(f_i),\mathscr{B}(f_j)\right)$ with ε is reported for varying values of ε_2.

10.2 Metric stability

Figure 14 of the main manuscript provides an empirical stability evaluation of our new metric $W_2^{\mathscr{T}}$, as a function of an input perturbation, modeled by a random noise of amplitude ε. In particular, this experiment is achieved for several values of ε_1. The conclusion of this experiment is that $W_2^{\mathscr{T}}$ is not stable when $\varepsilon_1 = 0$ (sudden increase in $W_2^{\mathscr{T}}$ for small values of ε) and that it is stable when $\varepsilon_1 = 1$ (as anticipated [112]). For in-between values, $W_2^{\mathscr{T}}$ is stable until a transition point (colored dots in Fig. 14 of the main manuscript), located at increasing noise levels (ε) for increasing values of ε_1. In particular, for the recommended default value $\varepsilon_1 = 0.05$, $W_2^{\mathscr{T}}$ is stable up to a perturbation noise of amplitude 16% (of the overall data range).

In the following, we perform the same study for the other parameters of our approach, ε_2 and ε_3. Fig. 18 studies the practical stability of $W_2^{\mathscr{T}}$, for several values of ε_2. For this experiment, ε_3 has been set to 1 (then, only ε_2 has an impact on the BDT pre-processing described in the previous section). Moreover, ε_1 has been set to its recommended value, 0.05. Several curves are reported, one per ε_2 values. For $\varepsilon_2 = 0$, all branches are moved up the BDT (irrespective of ε_1) and $W_2^{\mathscr{T}}$ becomes equivalent to $W_2^{\mathscr{D}}$ and the corresponding curve (red) exactly coincides with the light blue curve of the Figure 14 of the main manuscript ($\varepsilon_1 = 1$). For $\varepsilon_2 = 1$, the input BDT is not pre-processed at all and the corresponding curve (pink) exactly coincides with the cyan curve of Figure 14 of the main manuscript (obtained for the default value $\varepsilon_1 = 0.05$). In-between values of ε_2 result in continuous transitions between these two extreme cases (blue, green and cyan curves).

Fig. 19 studies the practical stability of $W_2^{\mathscr{T}}$, for several values of ε_3. For this experiment, we set $\varepsilon_1 = 0$ and $\varepsilon_2 = 0$, to better isolate the effect of ε_3. When $\varepsilon_3 = 1$, all the branches of the input BDTs are subject to the BDT pre-processing. Since $\varepsilon_2 = 0$, all branches are moved up to the root and $W_2^{\mathscr{T}}$ becomes equivalent to $W_2^{\mathscr{D}}$ and the corresponding curve (red) exactly coincides with the light blue curve of the Figure 14 of the main manuscript ($\varepsilon_1 = 1$). When $\varepsilon_3 = 0$, no branch is moved up in the input BDTs and the corresponding curve (pink) exactly coincides with the grey curve of the Figure 14 of the main manuscript ($\varepsilon_1 = 0$). In-between values of ε_3 result in transitions between these two extreme cases (blue, green and cyan curves), with transition points (similar to the Figure 14 of the main manuscript), before which $W_2^{\mathscr{T}}$ is stable. Note however, that since it is dependent on ε_2 (default value: 0.95), ε_3 has only a very mild practical impact on the metric.

10.3 Geodesic analysis

Figures 20, 21 and 22 respectively illustrate the effect of the parameters ε_1, ε_2 and ε_3 on the geodesics between merge trees. In particular, each figure shows, on the left, the geodesic obtained with a disabling value of the parameter (no effect on the computation). In contrast, the right side of each figure shows the geodesic obtained with the recommended default value of the parameter, to clearly visualize its impact.

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 the trees on the metric, but also improving its stability (as discussed in Section 10.2). In the data, moving a branch up the BDT corresponds to only slight modifications, which consist in reconnecting maxima to distinct saddles. For each parameter, the resulting pre-processing addresses cases where nearby saddles have very close function values, which impacts the stability of the metric. Similarly to Sridharamurthy et al. [106], we mitigate this effect with ε_1, but we also introduce ε_2 and ε_3 to specifically limit the importance in the metric of branches with persistence close to that of their parent.
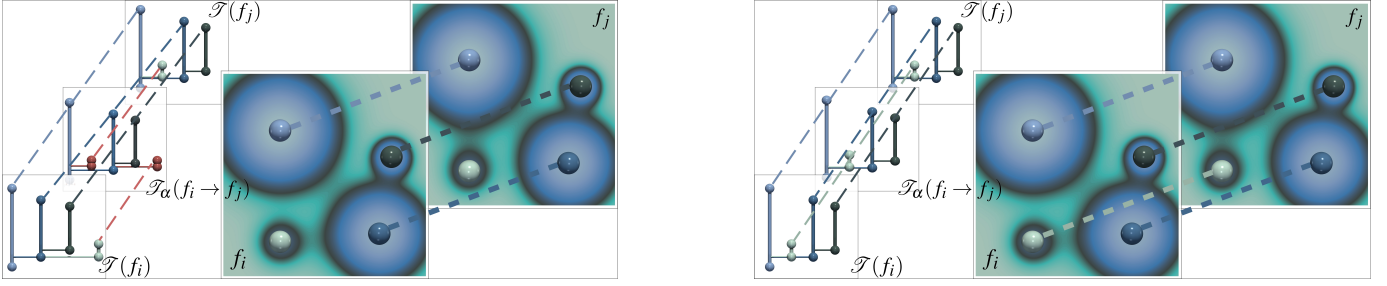

Fig. 19. Empirical stability evaluation with regard to ε_3. Given an input scalar field f_i, a noisy version f_j is created by inserting a random noise of increasing amplitude ε (cf. Figure 14 of the main manuscript). The evolution of $W_2^{\mathscr{T}}\left(\mathscr{B}(f_i),\mathscr{B}(f_j)\right)$ with ε is reported for varying values of ε_3.

Fig. 20. Impact of the parameter ε_1 on geodesic computation (left: $\varepsilon_1 = 0$, right: $\varepsilon_1 = 0.05$). In this example (left), the white branch in $\mathscr{T}(f_i)$ is not matched to the white branch in $\mathscr{T}(f_j)$ as they have distinct depths in the corresponding BDTs (2 versus 1). However, these features are visually similar in the data (Gaussians with the white maximum in f_i and f_j, bottom left corner of the domain). With $\varepsilon_1 = 0.05$ (right), the saddle of the white branch in $\mathscr{T}(f_i)$ gets merged with its ancestor saddle (whose f_i value was less than ε_1 away). Consequently, the white branch gets moved up the BDT (the white branch is attached to the main light blue branch in $\mathscr{T}(f_i)$, right). Since they now have identical depths in the corresponding BDTs, the white branches of $\mathscr{T}(f_i)$ and $\mathscr{T}(f_j)$ can now be matched together (right), which results in an overall matching (and geodesic) between these two trees which better conveys the resemblance between the two scalar fields f_i and f_j. Equivalently, one can interpret this procedure of saddle merge in the input trees as a modification of the input scalar field, turning f_i into f_j. In particular, this field modification disconnects the Gaussian with the white maximum from the Gaussian with the dark blue maximum (f_i) and reconnects it to the Gaussian with the light blue maximum (f_j).
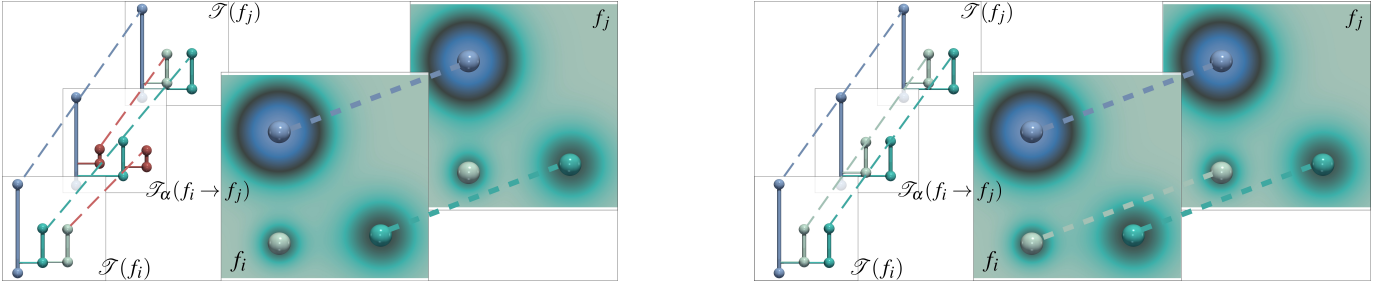

Fig. 21. Impact of the parameter ε_2 on geodesic computation (left: $\varepsilon_2 = 1$, right: $\varepsilon_2 = 0.95$). In this example (left), the white branch in $\mathscr{T}(f_i)$ is not matched to the white branch in $\mathscr{T}(f_j)$ as they have distinct depths in the corresponding BDTs (2 versus 1). Moreover, given the function difference between the white branch's saddle and its ancestor, that branch cannot be moved up the BDT under the effect of the ε_1 procedure (above). The white branch in $\mathscr{T}(f_i)$ 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 ε_2 (recommended default value: 0.95). In this example (right), the white branch in $\mathscr{T}(f_i)$ moves up the BDT and becomes adjacent to the main light blue branch in $\mathscr{T}(f_i)$. Since they now have identical depths in the corresponding BDTs, the white branches of $\mathscr{T}(f_i)$ and $\mathscr{T}(f_j)$ can now be matched together (right), which better conveys the resemblance between the two scalar fields f_i and f_j. Equivalently, one can interpret this procedure of BDT pre-processing as a modification of the input scalar field, turning f_i into f_j. In particular, this field modification disconnects the Gaussian with the white maximum from the Gaussian with the cyan maximum (f_i) and reconnects it to the Gaussian with the light blue maximum (f_j).
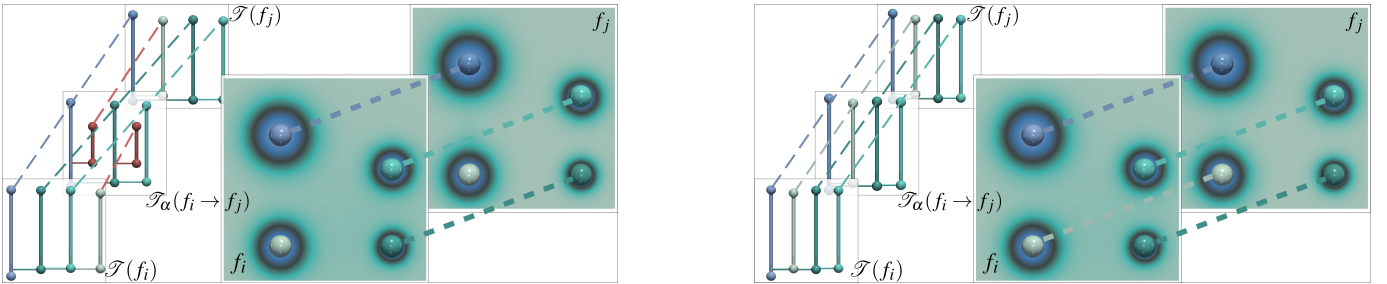

Fig. 22. Effect of the parameter ε_3 on geodesic computation (left: $\varepsilon_3 = 0$, right: $\varepsilon_3 = 0.9$). In this example (left), the white branch in $\mathscr{T}(f_i)$ is not matched to the white branch in $\mathscr{T}(f_j)$ as they have distinct depths in the corresponding BDTs (3 versus 1). Applying the above BDT pre-processing (ε_2) to all branches would move the cyan branch in $\mathscr{T}(f_i)$ up the BDT, which would prevent it to match to the cyan branch in $\mathscr{T}(f_j)$. The parameter ε_3 restricts the application of the above BDT pre-processing and prevents the movement of the most persistent branches (relative persistence larger than ε_3, default: 0.9). In this example (right), the white branch in $\mathscr{T}(f_i)$ moves up the BDT and becomes adjacent to the main light blue branch in $\mathscr{T}(f_i)$. Since they now have identical depths in the corresponding BDTs, the white branches of $\mathscr{T}(f_i)$ and $\mathscr{T}(f_j)$ can now be matched together (right), which results in an overall matching (and geodesic) between these two trees which better conveys the resemblance between the two scalar fields f_i and f_j. Equivalently, one can interpret this procedure on the BDTs as a modification of the input scalar field, turning f_i into f_j. In particular, this field modification disconnects the Gaussian with the white maximum from the Gaussian with the dark green maximum (f_i) and reconnects it to the Gaussian with the light blue maximum (f_j).