Wasserstein Dictionaries of Persistence Diagrams

Keanu Sisouk, Julie Delon, Julien Tierny

Wasserstein Dictionaries of Persistence Diagrams

Keanu Sisouk, Julie Delon, Julien Tierny

Abstract—This paper presents a computational framework for the concise encoding of an ensemble of persistence diagrams, in the form of weighted Wasserstein barycenters [100], [102] of a dictionary of *atom diagrams*. We introduce a multi-scale gradient descent approach for the efficient resolution of the corresponding minimization problem, which interleaves the optimization of the barycenter weights with the optimization of the *atom diagrams*. Our approach leverages the analytic expressions for the gradient of both sub-problems to ensure fast iterations and it additionally exploits shared-memory parallelism. Extensive experiments on public ensembles demonstrate the efficiency of our approach, with Wasserstein dictionary computations in the orders of minutes for the largest examples. We show the utility of our contributions in two applications. First, we apply Wassserstein dictionaries to *data reduction* and reliably compress persistence diagrams by concisely representing them with their weights in the dictionary. Second, we present a *dimensionality reduction* framework based on a Wasserstein dictionary defined with a small number of atoms (typically three) and encode the dictionary as a low dimensional simplex embedded in a visual space (typically in 2D). In both applications, quantitative experiments assess the relevance of our framework. Finally, we provide a C++ implementation that can be used to reproduce our results.

Index Terms—Topological data analysis, ensemble data, persistence diagrams.

1 Introduction

As measurement devices and numerical techniques are becoming more and more advanced, datasets are becoming more and more complex geometrically. This geometrical complexity makes interactive exploration and analysis difficult, which challenges the interpretation of the data by the users. This motivates the creation of expressive data abstractions, capable of encapsulating the main features of interest of the data into simple representations, visually conveying the main information to the user.

Topological Data Analysis (TDA) [29] is a family of techniques which precisely addresses this issue. It provides concise topological descriptors of the main structural features hidden in a dataset. The relevance of TDA for analyzing scalar data, its efficiency and robustness have been documented in a number of visualization tasks [47]. Examples of successful applications include turbulent combustion [16], [43], [56], material sciences [34], [45], [46], [94], nuclear energy [61], fluid dynamics [51], [65], bioimaging [3], [13], [20], chemistry [9], [38], [66], [67] or astrophysics [92], [95].

Among the different topological descriptors studied in TDA (such as the merge and contour trees [2], [18], [19], [39], [59], [97], the Reeb graph [11], [28], [40], [72], [73], [99] , or the Morse-Smale complex [15], [26], [30], [31], [37], [44], [87], [91]), the Persistence Diagram (Fig. 1) is a particularly prominent example. As described in Sec. 2.1, it is a concise topological descriptor which captures the main structural features in a dataset and which assesses their individual importance.

In addition to the challenge of increased geometrical complexity (discussed above), a new difficulty has recently emerged in many applications, with the notion of *ensemble dataset*. These representations describe a given phenomenon not only with a single dataset, but with a *collection* of datasets, called *ensemble members*. In that context, the topological analysis of an ensemble dataset consequently results in an ensemble of corresponding topological descriptors (e.g. one persistence diagram per ensemble member).

Then, a major challenge consists in developing practical tools for such an ensemble of topological descriptors, to facilitate its processing, analysis and visualization. Such tools include compression approaches (to facilitate the manipulation of the ensemble of descriptors) or visualization methods (for instance, with planar layouts, where each point encodes a descriptor and the distance between a pair of points encodes the intrinsic differences between the corresponding descriptors).

To enable the above tools, a key research question deals with the definition of a concise, yet informative, encoding of the ensemble of descriptors. A promising research direction consists in defining a *dictionary* (i.e. a set of reference descriptors, or *atoms*), such that the topological descriptors of the ensemble can be concisely encoded by expressing them as a specific *function* of the atoms (e.g. a linear combination). At a technical level, this requires to accurately capture and model the implicit relations (i.e. the possible functions) which link the different descriptors of the ensemble.

A series of recent works started the exploration of this overall direction, in particular with the notion of *average topological representation* [55], [79], [100], [102], [106]. These techniques can produce a topological descriptor which nicely summarizes the ensemble. However, they do not capture the implicit relations between the different topological descriptors.

This paper addresses this issue by introducing a simple and efficient approach for the estimation of linear relations between persistence diagrams on their associated Wasserstein metric space. Inspired by previous work on histograms [90], our approach provides a linear encoding of the input ensemble, where each diagram is represented as a weighted Wasserstein barycenter [100], [102] of a *dictionary* of automatically optimized diagrams called *atom diagrams*. We introduce a novel multi-scale gradient descent algorithm (Sec. 4) for the efficient resolution of the corresponding minimization problem (Sec. 3), for which we interleave the optimization of the barycenter weights (Sec. 3.2) with the optimization of the atom diagrams (Sec. 3.3). Extensive experiments (Sec. 6) on

public ensembles demonstrate the efficiency of our approach, with Wasserstein dictionary computations in the orders of minutes for the largest examples. We illustrate the relevance of our contributions for the visual analysis of ensemble data with two applications, data reduction (Sec. 5.1) and dimensionality reduction (Sec. 5.2).

1.1 Related Work

The literature related to our work can be classified into three main classes: *(i)* uncertainty visualization, *(ii)* ensemble visualization, and *(iii)* topological methods for ensembles.

(i) **Uncertainty visualization:** Data variability can be represented in the form of *uncertain* datasets, by considering the data at each point of the domain as a random variable, associated with an explicit probability density function (PDF). The analysis and visualization of uncertain data has been recognized as a major challenge in the visualization community [1], [14], [49], [60], [71], [85]. Several techniques have been proposed either dealing with the entropy of the random variables [84], or their correlation [77] or gradient variation [75]. The effect of data uncertainty on feature extraction has also been studied (for instance for level set extraction [4], [5], [76], [81]–[83], [89]), for various interpolation schemes and PDF models (e.g. Gaussian [58], [69], [70], [74] or uniform [10], [42], [96] distributions). In general, a central limitation of existing methods for uncertain data is their design dependence on a specific PDF model (Gaussian, uniform, etc). This challenges their usability for ensemble data, where the PDFs estimated from the ensemble can follow an arbitrary, unknown model. Moreover, most of these techniques do not consider multi-modal PDFs, which are however essential when multiple trends appear in the ensemble.

(ii) **Ensemble visualization:** Another approach to model data variability consists in using ensemble datasets. In this context, the variability is encoded by a sequence of empirical observations (*i.e* the members of the ensemble). Established techniques typically compute geometrical objects, such as level sets or streamlines, thereby capturing the main features for each member of the ensemble. From there, a *representative* of the resulting ensemble of geometrical objects can be computed. For this task, a few methods have been introduced. For instance spaghetti plots [27] are used in the case of level-set variability, more particularly for weather data [86], [88], and box-plots [62], [103] for the variability of contours and curves. In the case of trend variability, Hummel et al. [48] conceived a Lagrangian framework for classification purposes in flow ensembles. More specifically, clustering techniques have been used to identify the main trends in ensemble of streamlines [35] and isocontours [36]. However, only few techniques have applied this strategy to topological objects. Favelier et al. [33] and Athawale et al. [6] respectively introduced techniques to analyze the geometrical variability of critical points and gradient separatrices. Overlap-based heuristics have been studied for estimating a representative contour tree from an ensemble [53], [105]. In the context of ensembles of histograms, Schmitz et al. [90] introduced a dictionary encoding approach based on optimal transport [24]. However, this method is not directly applicable to persistence diagrams. It focuses on a fundamentally different object (histograms). Thus, the employed distances, geodesics and barycenters are defined differently (in particular in an entropic form [24], [25]) and the algorithms for their computations are drastically different (based on Sinkhorn matrix scaling [93]). In contrast, our work focuses on *Persistence diagrams* (Sec. 2.1), whose associated metric space is also inspired from optimal transport, but with various formal and computational specificities (Sec. 2.2). Moreover, our approach is based on gradient descent which, from our experience, provides better practical convergence for this kind of problems than quasi-Newton techniques. Finally, we contribute a multi-scale progressive optimization algorithm, which provides improved solutions in comparison to a naive optimization.

(iii) **Topological methods for ensembles:** To analyze the relations between the persistence diagrams of an ensemble, several key low level notions are required, such as the notion of distance and barycenters between diagrams, for which we review the literature here. Inspired by optimal transport [50], [63], the *Wasserstein* distance between persistence diagrams [29] (Sec. 2.2) has been extensively studied [22], [23]. It relies on a bipartite assignment problem, for which exact [64] and approximate [8], [52] implementations are available in open-source [98]. Based on this distance, several approaches have explored the possibility to define a *representative* diagram of an ensemble of persistence diagrams, with the notion of *Wasserstein* barycenter. Turner et al. [100] introduced the first approach for the computation of such a barycenter. Lacombe et al. [55] presented an approach based on entropic transport [24], [25]. However, it requires a pre-vectorization step which is subject to several parameters, and which is not conducive to visualization tasks (features can no longer be individually tracked beyond the pre-vectorization step). In contrast, Vidal et al. [102] introduced a vectorization-free approach which maintains the feature assignments explicitly. It is based on a progressive scheme, which greatly accelerates computation in practice. These concepts have been recently investigated for other topological descriptors, such as merge trees [79], [106]. Recently, several authors have investigated another compact representation of ensembles of topological descriptors, via a basis of representative descriptors. For instance, Li et al. [57] introduce a vectorization for merge trees, which was subsequently used by matrix sketching procedures [104] to create a basis of representative merge trees. In contrast, our work focuses on persistence diagrams (which can encode different features). Also, it directly operates on the Wasserstein metric space of persistence diagrams, thereby avoiding the typical technical difficulties associated with vectorizations (e.g. quantization and/or linearization artifacts, potential stability issues, possible inaccuracies in vectorization reversal, etc.). Pont et al. [80] introduced the notion of principal geodesic analysis of merge trees (and persistence diagrams), with the same overall goal of characterizing the relations between the topological descriptors of an ensemble. In this work, we introduce a different formulation of the problem, which is both simpler (based on the construction of weighted Wasserstein barycenters) and more flexible (our optimization is not subject to complicated constraints such as geodesic orthogonality). This results in a simpler implementation and slightly faster computations (Sec. 6).

1.2 Contributions

This paper makes the following new contributions:

1) *A simple approach for the linear encoding of Persistence Diagrams:* We formulate the linear encoding of an ensemble of persistence diagrams on their associated Wasserstein metric space as a dictionary optimization (Sec. 3), which simply optimizes, simultaneously, *(i)* the barycentric weights (Sec. 3.2) and *(ii)* the atoms of the dictionary (Sec. 3.3).
2) *A multi-scale algorithm for the computation of a Wasserstein dictionary of Persistence Diagrams:* We introduce a novel,

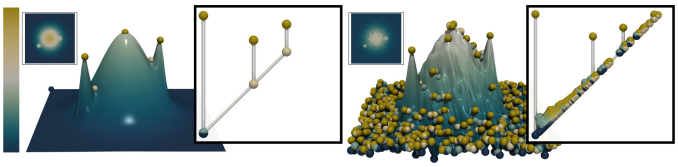


Fig. 1. Persistence diagrams of a clean (left) and noisy (right) terrain (dark blue spheres: minima, dark yellow: maxima, other: saddles). The three main hills are clearly represented with long bars in the persistence diagrams. In the noisy persistence diagram, small bars encode noise.

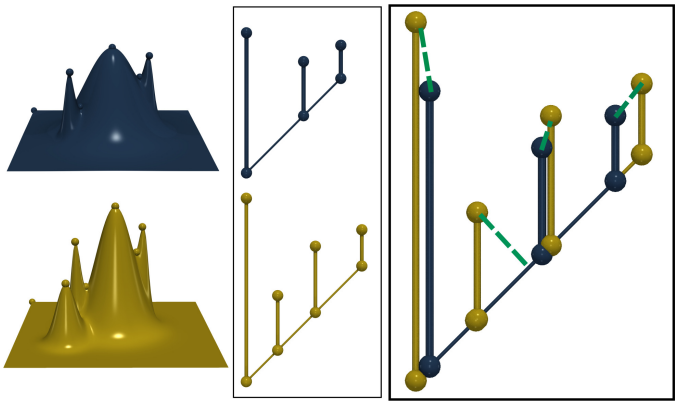


Fig. 2. Optimal matching (green dashes, right) with regard to W between the two persistence diagrams (center) of two terrains (left).

efficient algorithm for the optimization of the above dictionary encoding problem. Our algorithm leverages the analytic expressions of the gradient of both of the above sub-problems, to ensure fast iterations. Moreover, in comparison to a naive optimization, our algorithm reaches solutions of improved energy thanks to a multi-scale strategy. Finally, we leverage shared-memory parallelism to further improve performances.

3) *An application to data reduction:* We present an application to data reduction (Sec. 5.1), where the persistence diagrams of the input ensemble are significantly compressed, by solely storing their barycentric weights as well as the atom diagrams.
4) *An application to dimensionality reduction:* We present an application to dimensionality reduction (Sec. 5.2), by embedding each input diagram as a point within a 2D view, based on its weights relative to a Wasserstein dictionary composed of three atoms (thereby defining a 2-simplex).
5) *Implementation:* We provide a C++ implementation of our algorithms that can be used for reproducibility purposes.

2 Preliminaries

This section presents the theoretical elements needed for the formalization of our work. We introduce the topological data representation that we use - the persistence diagram (Sec. 2.1) - and its associated metric (Sec. 2.2). Then we define the notion of Wasserstein barycenter of persistence diagrams (Sec. 2.3), which is a core component of our approach (Sec. 3).

2.1 Persistence diagrams

Each input ensemble member is given in the form of a piecewise linear (PL) scalar field $f : \mathcal{M} \to \mathbb{R}$ defined on a PL $(d_{\mathcal{M}})$-manifold $\mathcal{M}$, with $d_{\mathcal{M}} \leq 3$ for our applications. Given an isovalue $w \in \mathbb{R}$, we denote $f^{-1}_{-\infty}(w) = f^{-1}\big((-\infty, w]\big)$ the sub-level set of f at w. As

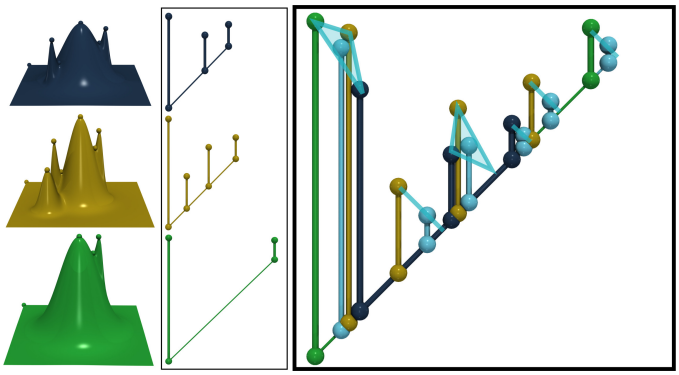


Fig. 3. Wasserstein barycenter (cyan, uniform weights) of 3 persistence diagrams (center) of 3 terrains (left). Each barycenter point (cyan sphere) is the barycenter of its matched points in the inputs (cyan triangle).

w increases, the topology of $f^{-1}_{-\infty}$ changes at specific points of $\mathcal{M}$, called *"critical points"*. Critical points are classified by their index $\mathcal{I}$: 0 for minima, 1 for 1-saddles, $d_{\mathcal{M}} - 1$ for $(d_{\mathcal{M}} - 1)$-saddles, and $d_{\mathcal{M}}$ for maxima (in practice, f is enforced to contain only isolated, non-degenerate critical points [31], [32]). According to the Elder rule [29], each topological feature of $f^{-1}_{-\infty}(w)$ (e.g. a connected component, a cycle, a void) can be associated with a pair of critical points (c, c') (with $f(c) < f(c')$ and $\mathcal{I}_c = \mathcal{I}_{c'} - 1$), corresponding to its *birth* and *death* during the sweep of the data by w (from $-\infty$ to $+\infty$). Such a pair (c, c') is called a *persistence pair*. For instance, when two connected components of $f^{-1}_{-\infty}(w)$ meet at a critical point c', the younger one (created last) *dies*, letting the oldest one (created first) survive. Then, the critical points are represented visually as 2D bar codes where the horizontal axis encodes the *birth* of a feature $\big($noted $b = f(c)\big)$ and where the vertical axis encodes its lifespan and *death* $\big($noted $d = f(c')\big)$. This representation is called the *Persistence Diagram*, noted X. In the diagram, salient features stand out from the diagonal and small-amplitude noise is typically located near the diagonal, as shown in Fig. 1. In the remainder, we enumerate the points of X with indices such that $X = \{x^1, \ldots, x^K\}$ and we note $i_X = \{1, \ldots, K\}$ the set of indices (i.e. the set of all integers going from 1 to K) .

2.2 Wasserstein distance

To evaluate the distance between two diagrams, a typical pre-processing step consists in augmenting each diagram, by including the diagonal projection of all the off-diagonal points of the other diagram. To illustrate that, let us consider $X_1 = \{x_1^1, \ldots, x_1^{K_1}\}$ and $X_2 = \{x_2^1, \ldots, x_2^{K_2}\}$. Given an off-diagonal point x (i.e. $b < d$), let Δx be its diagonal projection, specifically: $\Delta x = (\frac{b+d}{2}, \frac{b+d}{2})$. Let P_1 and P_2 be the sets of the diagonal projections of the points of X_1 and X_2 respectively. Then, X_1 and X_2 are augmented into ${X_1}'$ and ${X_2}'$ by considering ${X_1}' = X_1 \cup P_2$ and ${X_2}' = X_2 \cup P_1$. This ensures that $|X_1'| = |X_2'| = K$ (which eases distance evaluation). Specifically, we consider in the remainder that the notations X_1 and X_2 refer to *augmented diagrams* (i.e. $|X_1| = |X_2| = K$).

Then, given two persistence diagrams X_1 and X_2 the L^2-Wasserstein distance between them is defined as:

$$W(X_1, X_2) = \min_{\psi : i_X \xrightarrow{bij} i_X} \sqrt{\sum_{j=1}^{K} c(x_1^j, x_2^{\psi(j)})}, \qquad (1)$$

where ψ, the matching, is a bijection of the index set i_X towards itself (i.e. ψ is a permutation of i_X). This bijection matches one

persistence pair x_1 of X_1 (i.e. a pair of critical points of index $\mathscr{I}$ and $\mathscr{I}+1$ respectively) to one persistence pair x_2 in X_2 (another pair of critical points, with the *same* indices $\mathscr{I}$ and $\mathscr{I}+1$) whether x_1 and x_2 are on the diagonal or not (Fig. 2). Given the cost c in the definition of W, ψ is the optimal way to transport X_1 onto X_2. In our work, we consider the cost $c(x,y)=0$ when x and y are both diagonal points, and $c(x,y)=\|x-y\|^2$ otherwise ($\|x-y\|$ denotes the Euclidean distance between x and y in the birth/death space).

2.3 Wasserstein barycenter

Given a set of persistence diagrams $\mathscr{D}=\{a_1,\ldots,a_m\}$ (which we will call in the remainder *dictionary*), a Wasserstein barycenter (Fig. 3) – or Fréchet mean – of the dictionary $\mathscr{D}$ with barycentric weights $\boldsymbol{\lambda}=(\lambda_1,\ldots,\lambda_m)$ is a diagram, which we note $Y(\boldsymbol{\lambda},\mathscr{D})$ in the following, which minimizes the Fréchet energy $E_F(B)$:

$$E_F(B)=\sum_{i=1}^{m}\lambda_i W^2(a_i,B).$$

$\boldsymbol{\lambda}$ is such that $\lambda_i\geq 0$ and $\sum_{i=1}^{m}\lambda_i=1$. We denote Σ_m the simplex of such vectors. Intuitively, $Y(\boldsymbol{\lambda},\mathscr{D})$ is a diagram which minimizes the above linear combination, given $\boldsymbol{\lambda}$, of its squared Wasserstein distances to the diagrams of the dictionary $\mathscr{D}$.

The computation of the barycenter $Y(\boldsymbol{\lambda},\mathscr{D})$ requires generalizing the pairwise augmentation described in Sec. 2.2. Specifically, each non-diagonal point of each dictionary diagram a_i is projected to the diagonal of all the other dictionary diagrams a_j (with $i\neq j$). After this first augmentation, each dictionary diagram a_i contains $K=\sum_{i=1}^{m}|a_i|$ points (where $|a_i|$ is the number of non-diagonal points in a_i). Then $Y(\boldsymbol{\lambda},\mathscr{D})$ is typically initialized on the dictionary diagram a_* which initially minimizes the Fréchet energy E_F. Let $|Y(\boldsymbol{\lambda},\mathscr{D})|=|a_*|$ be the number of non-diagonal points of a_*. Then, all the non-diagonal points of all the atoms are projected on the diagonal of $Y(\boldsymbol{\lambda},\mathscr{D})$, and reciprocally, all the non-diagonal points of $Y(\boldsymbol{\lambda},\mathscr{D})$ are projected on the diagonal of each atom. Thus, at this stage, after this second augmentation, each dictionary diagram a_i and the candidate barycenter $Y(\boldsymbol{\lambda},\mathscr{D})$ contains $K=\sum_{i=1}^{m}|a_i|+|Y(\boldsymbol{\lambda},\mathscr{D})|$ points (mostly on the diagonal).

Next, we optimize $Y(\boldsymbol{\lambda},\mathscr{D})$ in practice with the approach by Vidal et al. [102], which provides a time-efficient approximation of the original algorithm by Turner et al. [100]. Similar to Turner et al., it is based on an iterative optimization, where each iteration includes an *Assignment* step, followed by an *Update* step. Specifically, the *Assignment* step computes the optimal assignments ψ_i between the candidate $Y(\boldsymbol{\lambda},\mathscr{D})$ and each dictionary diagram a_i. Next, the *Update* step minimizes the Fréchet energy E_F under the current assignments ψ_i. Since the L^2-Wasserstein distance considers the Euclidean distance as a cost function (Sec. 2.2), this minimization is achieved by simply placing each point of $Y(\boldsymbol{\lambda},\mathscr{D})$ at the arithmetic mean in the birth/death space of its assigned points in the dictionary diagrams. This can be done since the arithmetic mean minimizes the Fréchet energy defined respectively to Euclidean distances (more sophisticated *Update* procedures, e.g. based on an optimization routine, would need to be derived for other distances in the birth/death space). After this *Update* step, the subsequent *Assignment* further improves the assignments ψ_i, hence decreasing the Fréchet energy constructively at each iteration.

The approach by Vidal et al. [102] revisits this framework by integrating tailored approximations throughout the computation. Specifically, it approximates the optimal assignments ψ_i with the fast *Auction* optimization [8] (instead of the traditional, yet prohibitive, *Munkres* algorithm [64]). Further, it improves performance with a mechanism called *price memorization*, which enables the initialization of the *Auction* optimization with the assignments ψ_i computed in the previous *Assignment* step. This allows the barycenter optimization to resume the assignment optimization instead of re-computing it from scratch at each iteration. This approach also includes a strategy for the adaptive increase of the *accuracy* parameter of the *Auction* optimization, allowing for fast assignments in the early iterations of the barycenter algorithm, and slower but more accurate assignments towards its convergence.

3 Wasserstein Dictionary Encoding

This section formalizes our approach for the Wasserstein dictionary encoding of an ensemble of persistence diagrams. Sec. 3.1 provides an overview of our approach, which interleaves barycentric weight optimization ($\boldsymbol{\lambda}$) with atom optimization ($\mathscr{D}$). Finally, Secs. 3.2 and 3.3 detail the gradient estimation for both sub-problems.

3.1 Overview

Let $\{X_1,\ldots,X_N\}$ be the input ensemble of N persistence diagrams. The goal of our approach is to jointly optimize two sub-problems:

- Optimize a set $\mathscr{D}$ of m *reference* persistence diagrams, called the *atoms* of the *Wasserstein dictionary* $\mathscr{D}$;
- Optimize for each input diagram X_n a vector of m barycentric weights $\boldsymbol{\lambda}_n\in\Sigma_m$, in order to accurately approximate X_n with a Wasserstein barycenter $Y(\boldsymbol{\lambda}_n,\mathscr{D})$ (Sec. 2.3).

This can be formalized as a joint optimization, where one wishes to find the optimal barycentric weights $\Lambda^*=\boldsymbol{\lambda}_1^*,\ldots,\boldsymbol{\lambda}_N^*$ and the optimal Wasserstein dictionary $\mathscr{D}_*=\{a_1^*,\ldots,a_m^*\}$ (with $m\ll N$), in order to minimize the following *dictionary energy*:

$$E_D(\Lambda,\mathscr{D})=\sum_{n=1}^{N}W^2\big(Y(\boldsymbol{\lambda}_n,\mathscr{D}),X_n\big). \tag{2}$$

Our overall strategy for optimizing Eq. 2 consists in iteratively interleaving two sub-optimizations:

1) For a fixed dictionary $\mathscr{D}$, the set of barycentric weights Λ is optimized with one step of gradient descent (Sec. 3.2);
2) For a fixed set of barycentric weights Λ, the dictionary $\mathscr{D}$ is optimized with one step of gradient descent (Sec. 3.3).

Then, this sequence of two sub-procedures is iterated until a predefined stopping condition is reached (Sec. 4.2).

Finally, the output of our approach is the optimized Wasserstein dictionary $\mathscr{D}_*$ (a set of m atom diagrams) and, for each input diagram X_n, a vector of weights $\boldsymbol{\lambda}_n^*\in\Sigma_m$, which can be interpreted as the barycentric coordinates of X_n in $\mathscr{D}_*$ (thus capturing linear relations between the input diagrams on the Wasserstein dictionary).

3.2 Weight optimization

This section details the optimization of the barycentric weights $\Lambda=\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_N$. Let $\mathscr{D}=\{a_1,\ldots,a_m\}$ be a fixed dictionary of atom diagrams, with $m>0$. Let X be a diagram of the input ensemble. For a given set of weights $\boldsymbol{\lambda}=(\lambda_1,\ldots,\lambda_m)$, let $Y(\boldsymbol{\lambda})=\{y^1(\boldsymbol{\lambda}),\ldots,y^K(\boldsymbol{\lambda})\}$ be its barycentric approximation, relative to $\mathscr{D}$ (i.e. each point $y^j(\boldsymbol{\lambda})$ of $Y(\boldsymbol{\lambda})$ approximates a point in X).

We recall that after augmentation (Sec. 2.3), $Y(\boldsymbol{\lambda})$ and the atoms contain $\sum_{i=1}^{m}|a_i|+|Y(\boldsymbol{\lambda})|$ points each, where $|a_i|$ and $|Y(\boldsymbol{\lambda})|$ denote the number of *non-diagonal* points in a_i and $Y(\boldsymbol{\lambda})$ respectively.

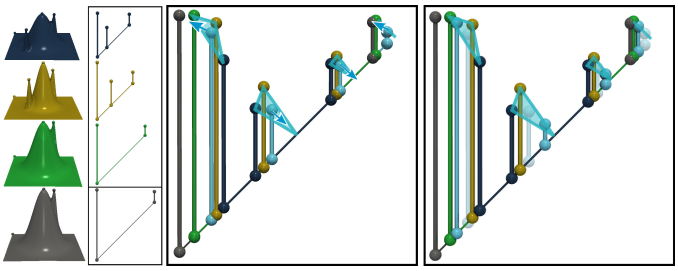


Fig. 4. Optimizing the weights of the barycenter $Y(\boldsymbol{\lambda})$ (cyan diagram) to improve its approximation of X (grey diagram), given a fixed Wasserstein dictionary $\mathscr{D}$ of 3 atoms (dark blue, yellow, green). At a given iteration t (center), a step $\rho_{\boldsymbol{\lambda}}$ is made along the gradient of the weight energy E_W (cyan arrows), resulting in an improved estimation at iteration $t+1$ (right).

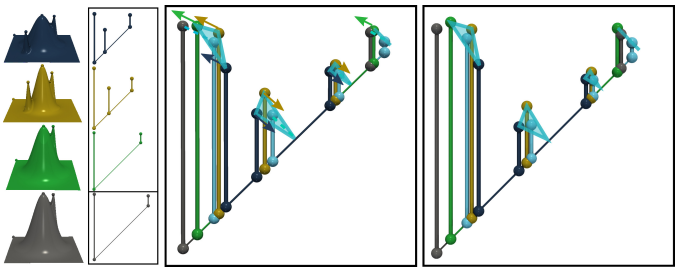


Fig. 5. Optimizing the atoms of the Wasserstein dictionary $\mathscr{D}$ (dark blue, yellow and green diagrams). At a given iteration t (center), a step $\rho_{\mathscr{D}}$ is made along the gradient of the pointwise atom energy e_A (arrows on each triangle), resulting in a dictionary (right) that enables an improved barycentric approximation ($Y(\mathscr{D})$, cyan) of the input diagram X (grey).

Then, in order to compare it to X, $Y(\boldsymbol{\lambda})$ is further augmented by projecting on its diagonal the $|X|$ non-diagonal points of X. Then, at this stage, the size K of $Y(\boldsymbol{\lambda})$ is given by $K=\sum_{i=1}^{m}|a_i|+|Y(\boldsymbol{\lambda})|+|X|$. We augment similarly X (i.e. by projecting the non-diagonal points of $Y(\boldsymbol{\lambda})$ to its diagonal) and the m atoms (i.e. by projecting the non-diagonal points of X to their diagonals). Then, at this point, $Y(\boldsymbol{\lambda})$, X, and the m atoms a_i all have the same size $K=\sum_{i=1}^{m}|a_i|+|Y(\boldsymbol{\lambda})|+|X|$.

In this section, we describe a gradient descent on $\boldsymbol{\lambda}$ to minimize the *weight energy*:

$$E_W(\boldsymbol{\lambda}) = W^2\big(Y(\boldsymbol{\lambda}),X\big). \tag{3}$$

A step of the corresponding gradient descent is illustrated in Fig. 4.

Given the set of optimal matchings $\phi_1,\dots,\phi_m$ between $Y(\boldsymbol{\lambda})$ and the m atoms, the j^{th} point of $Y(\boldsymbol{\lambda})$, noted $y^j(\boldsymbol{\lambda})$, is given by:

$$\forall j\in\{1,\dots,K\},\ y^j(\boldsymbol{\lambda}) = \sum_{i=1}^{m}\lambda_i a_i^{\phi_i(j)}. \tag{4}$$

In other words, the j^{th} point $y^j(\boldsymbol{\lambda})$ of the diagram $Y(\boldsymbol{\lambda})$ is a linear combination (with the weights $\boldsymbol{\lambda}$) of the m points it matches to in the atoms (one point per atom a_i), under the optimal assignments ϕ_i (i.e. minimizing Eq. 1).

For a fixed set of assignments $\phi_1,\dots,\phi_m$, the Wasserstein distance (Eq. 1) between X and its approximation $Y(\boldsymbol{\lambda})$ is then:

$$E_W(\boldsymbol{\lambda}) = W^2\big(Y(\boldsymbol{\lambda}),X\big) = \sum_{j=1}^{K} c\big(y^j(\boldsymbol{\lambda}),x^{\psi(j)}\big),$$

where ψ denotes the optimal assignment (Eq. 1) between X and its approximation $Y(\boldsymbol{\lambda})$. When $y^j(\boldsymbol{\lambda})$ and $x^{\psi(j)}$ are not both diagonal points, the cost $c\big(y^j(\boldsymbol{\lambda}),x^{\psi(j)}\big)$ is given by their squared Euclidean distance in the birth/death space (it is zero otherwise, see Sec. 2.2). Then, by exploiting Eq. 4, $E_W(\boldsymbol{\lambda})$ can be re-written as:

$$\begin{aligned} E_W(\boldsymbol{\lambda}) = W^2\big(Y(\boldsymbol{\lambda}),X\big) &= \sum_{j=1}^{K}\|y^j(\boldsymbol{\lambda})-x^{\psi(j)}\|^2 \\ &= \sum_{j=1}^{K}\Big\|\Big(\sum_{i=1}^{m}\lambda_i a_i^{\phi_i(j)}\Big)-x^{\psi(j)}\Big\|^2. \end{aligned}$$

Since $\sum_{i=1}^{m}\lambda_i=1$, $E_W(\boldsymbol{\lambda})$ can finally be re-written as:

$$E_W(\boldsymbol{\lambda}) = W^2\big(Y(\boldsymbol{\lambda}),X\big) = \sum_{j=1}^{K}\Big\|\sum_{i=1}^{m}\lambda_i(a_i^{\phi_i(j)}-x^{\psi(j)})\Big\|^2. \tag{5}$$

Intuitively, this energy measures the error (in terms of Wasserstein distance) induced by approximating the input diagram X with its barycentric approximation $Y(\boldsymbol{\lambda})$. In Eq. 5, it is computed for each j^{th} point $y^j(\boldsymbol{\lambda})$ of the diagram $Y(\boldsymbol{\lambda})$, by considering the birth/death distances between the points $y^j(\boldsymbol{\lambda})$ maps to, in the atoms on one hand and in the input diagram X on the other.

Then, by applying the chain rule on Eq. 5, the gradient of the weight energy (Eq. 3) is given by:

$$\nabla E_W(\boldsymbol{\lambda}) = 2\sum_{i=1}^{m}\sum_{j=1}^{K}\begin{bmatrix}(a_1^{\phi_1(j)}-x^{\psi(j)})^T\\ \vdots \\ (a_m^{\phi_m(j)}-x^{\psi(j)})^T\end{bmatrix}\big(\lambda_i(a_i^{\phi_i(j)}-x^{\psi(j)})\big). \tag{6}$$

Now that the gradient of the weight energy is available (Eq. 6), we can proceed to gradient descent. Specifically, the barycentric weights at the iteration $t+1$ (noted $\boldsymbol{\lambda}^{t+1}$) are obtained by a step $\rho_{\boldsymbol{\lambda}}$ from the weights at the iteration t (noted $\boldsymbol{\lambda}^t$) along the gradient:

$$\boldsymbol{\lambda}^{t+1} = \Pi_{\Sigma_m}\big(\boldsymbol{\lambda}^t-\rho_{\boldsymbol{\lambda}}\nabla E_W(\boldsymbol{\lambda}^t)\big), \tag{7}$$

where Π_{Σ_m} is the projection onto the simplex of admissible barycentric weights (i.e. positive and summing to 1, c.f. Sec. 2.3). Since ∇E_W is L-Lipschitz (see the computation details in Appendix A), a gradient step will guarantee an energy decrease as long as:

$$\rho_{\boldsymbol{\lambda}} \le \left[2\sum_{j=1}^{K}\left\|\begin{bmatrix}(a_1^{\phi_1(j)}-x^{\psi(j)})^T\\ \vdots \\ (a_m^{\phi_m(j)}-x^{\psi(j)})^T\end{bmatrix}\right\|^2\right]^{-1} < \frac{1}{L}.$$

Overall, for a given input diagram X, each iteration t of gradient descent for the optimization of E_W consists in the following steps:

1) Computing the Wasserstein barycenter $Y(\boldsymbol{\lambda}^t)$ (Sec. 2.3);
2) Computing the Wasserstein distance $W^2\big(Y(\boldsymbol{\lambda}^t),X\big)$ (Eq. 3);
3) Estimating the gradient $\nabla E_W(\boldsymbol{\lambda})$ (Eq. 6);
4) Applying one step $\rho_{\boldsymbol{\lambda}}$ of gradient descent (Eq. 7).

3.3 Atom optimization

This section details the optimization of the atoms of the dictionary $\mathscr{D}=\{a_1,\dots,a_m\}$. Similarly to Sec. 3.2, let X be a diagram of the input ensemble and let $\boldsymbol{\lambda}\in\Sigma_m$ be its – fixed – vector of barycentric weights. For a given dictionary $\mathscr{D}$, let $Y(\mathscr{D})=\{y^1(\mathscr{D}),\dots,y^K(\mathscr{D})\}$ be the barycentric approximation of X, relative to $\boldsymbol{\lambda}$. In this section, we describe a step of gradient descent on $\mathscr{D}$ to minimize the following *atom energy*:

$$E_A(\mathscr{D}) = W^2\big(Y(\mathscr{D}),X\big).$$

A step of the corresponding gradient descent is illustrated in Fig. 5.

Given the set of optimal matchings $\phi_1,\dots,\phi_m$ between $Y(\mathscr{D})$ and the m atoms, the j^{th} point of $Y(\mathscr{D})$, noted $y^j(\mathscr{D})$, is given by:

$$\forall j \in \{1,\dots,K\},\ y^j(\mathscr{D}) = \sum_{i=1}^{m} \lambda_i a_i^{\phi_i(j)}.$$

This expression is identical to Eq. 4 (Sec. 3.2). However, y^j now depends on $\mathscr{D}$, which is the variable of the current optimization. Then, the gradient of $y^j(\mathscr{D})$ with regard to $\mathscr{D}$ is simply given by:

$$\nabla y^j(\mathscr{D}) = \begin{bmatrix}\lambda_1 & \cdots & \lambda_m\end{bmatrix}^T. \tag{8}$$

For a fixed set of assignments $\phi_1,\dots,\phi_m$, the Wasserstein distance (Eq. 1) between X and its approximation $Y(\boldsymbol{\lambda})$ is then:

$$E_A(\mathscr{D}) = W^2\big(Y(\mathscr{D}),X\big) = \sum_{j=1}^{K} c\big(y^j(\mathscr{D}),x^{\psi(j)}\big),$$

where $\psi(j)$ denotes the optimal assignment between X and its barycentric approximation $Y(\mathscr{D})$. Similarly to Eq. 5 (Sec. 3.2), the above equation can be re-written as:

$$E_A(\mathscr{D}) = W^2\big(Y(\mathscr{D}),X\big) = \sum_{j=1}^{K} \| \sum_{i=1}^{m} \lambda_i (a_i^{\phi_i(j)} - x^{\psi(j)})\|^2.$$

Let $\mathscr{D}^j = [a_1^{\phi_1(j)},\dots,a_m^{\phi_m(j)}]^T$ be the $(m\times 2)$-matrix formed by the atom points matching to a given point $y^j(\mathscr{D})$ of $Y(\mathscr{D})$, via the fixed assignments $\phi_1,\dots,\phi_m$. Specifically, the i^{th} line of this matrix refers to the point $a_i^{\phi_i(j)}$ in the atom a_i where $y^j(\mathscr{D})$ maps to (via the optimal assignment ϕ_i). For this line, the two columns of the matrix encode the birth/death coordinates of the point $a_i^{\phi_i(j)}$. Then, the *pointwise atom energy* of $y^j(\mathscr{D})$, noted $e_A(\mathscr{D}^j)$, is given by:

$$e_A(\mathscr{D}^j) = \left\| \sum_{i=1}^{m} \lambda_i (a_i^{\phi_i(j)} - x^{\psi(j)}) \right\|^2. \tag{9}$$

Then, by applying the chain rule on Eq. 9 (using Eq. 8), the gradient of the pointwise atom energy is given by:

$$\nabla e_A(\mathscr{D}^j) = 2\begin{bmatrix}\lambda_1 & \cdots & \lambda_m\end{bmatrix}^T \big(\sum_{i=1}^{m} \lambda_i (a_i^{\phi_i(j)} - x^{\psi(j)})^T\big). \tag{10}$$

Now that the gradient of the pointwise atom energy is available (Eq. 10), we can proceed to a step of gradient descent. Specifically, the matrix of atom points matched to $y^j(\mathscr{D})$ at the iteration $t+1$ (noted $\mathscr{D}^j_{t+1}$) is obtained by a step $\rho_{\mathscr{D}}$ from the same matrix at the iteration t (noted $\mathscr{D}^j_t$) along the gradient:

$$\mathscr{D}^j_{t+1} = \Pi_{\mathscr{X}}\big(\mathscr{D}^j_t - \rho_{\mathscr{D}} \nabla e_A(\mathscr{D}^j_t)\big), \tag{11}$$

where $\Pi_{\mathscr{X}}$ projects each atom point to an admissible region of the 2D birth/death space (i.e. above the diagonal and within the global scalar field range). Since ∇e_A is L-Lipschitz (see the computation details in Appendix B), a gradient step will guarantee an energy decrease as long as: $\rho_{\mathscr{D}} < (4m)^{-1} < L^{-1}$.

Note that, in order to control the final size S_m of the dictionary $\mathscr{D}$, after each iteration of atom optimization, each atom a_i is thresholded by removing its $\overline{K} = (mK - S_m)/m$ least persistent points (at the subsequent optimization iteration, all diagrams will be re-augmented again in a pre-preprocess, as detailed in Sec. 3.2).

Overall, for a given input diagram X, each iteration t of gradient descent for the optimization of E_A consists in the following steps:

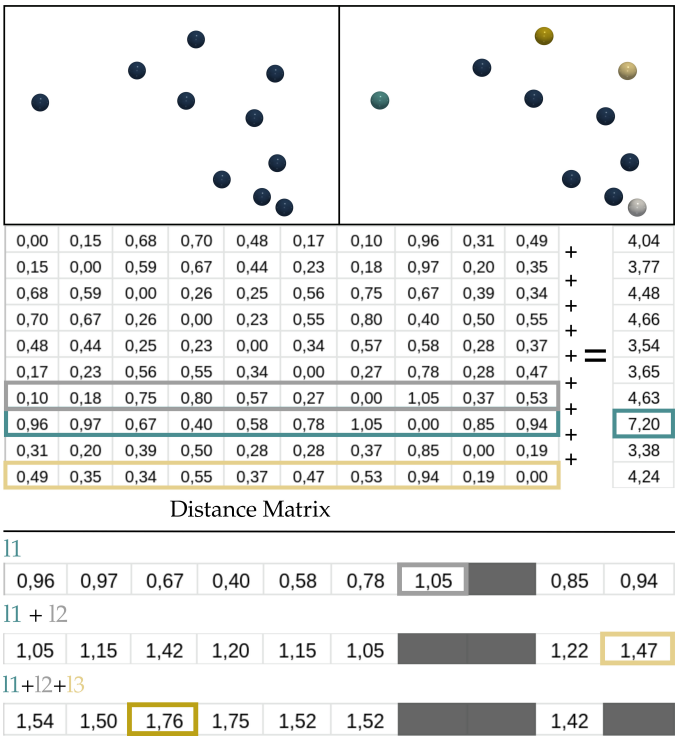


Fig. 6. Illustration of our initialization strategy on a toy 2D point set (top left). First, the entries of the distance matrix of the input (middle) are summed on a per-line basis. The line maximizing this sum (cyan), noted l_1, identifies the first atom, noted a_1, as the point which is the *furthest away* from all the others (cyan sphere, top right). Next, the atom a_2 (grey sphere, top right) is selected as the point which maximizes its distance to a_1. At this point, the line l_2 (grey, corresponding to the point a_2) is added to the line l_1, to encode the distances to these two atoms (a_1 and a_2). Then, the point a_3 is selected as the maximizer of l_1+l_2: it is the point which is the furthest away from all the previously selected atoms. Then, the corresponding line, l_3, is added to l_1+l_2 and the process is iterated until the target number of atoms has been achieved.

1) Computing the Wasserstein barycenter $Y(\mathscr{D}_t)$ (Sec. 2.3);
2) Computing the Wasserstein distance $W^2\big(Y(\mathscr{D}_t),X\big)$ (Eq. 3);
3) For each point $y^j(\mathscr{D})$ of $Y(\mathscr{D})$:
   a) Estimating the gradient $\nabla e_A(\mathscr{D}^j)$ (Eq. 10);
   b) Applying one step $\rho_{\mathscr{D}}$ of gradient descent on $\mathscr{D}^j$ (Eq. 11);
4) Remove the $\overline{K}$ least persistent points from each atom a_i.

4 Algorithm

This section presents our overall algorithm for the resolution of the optimization formulated in Sec. 3. Sec. 4.1 details our initialization strategy. Our overall multi-scale scheme is presented in Sec. 4.2. Finally, shared-memory parallelism is discussed in Sec. 4.3.

4.1 Initialization

Our strategy for the initialization of the Wasserstein dictionary $\mathscr{D}$, illustrated in Fig. 6, is inspired by the celebrated *k-means++* strategy [21]. Specifically, we iteratively select the m atoms among the N input diagrams. At the first iteration, we select as first atom the diagram which maximizes the sum of its Wasserstein distances (Eq. 1) to all the input diagrams (cyan point in Fig. 6). Next, each iteration selects as the next atom the diagram which maximizes the sum of its Wasserstein distances to all the previously selected atoms. This process stops when the desired number of atoms, m,

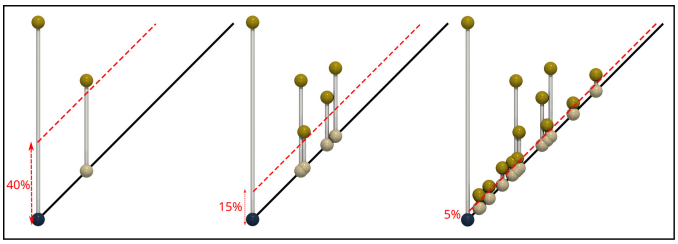


Fig. 7. Multi-resolution representation of an input persistence diagram (taken from the *Isabel* ensemble). At a given resolution (from left to right), only the persistence pairs above a given persistence threshold (red dash line) are considered in the optimization.

Algorithm 1: Multi-scale Wasserstein Dictionary Optimization.

Input: Set of persistence diagrams $\{X_1, \ldots, X_N\}$;
Output 1: Wasserstein Dictionary $\mathscr{D}_*$;
Output 2: Barycentric weights $\boldsymbol{\lambda}_1^*, \ldots, \boldsymbol{\lambda}_N^*$;
for $\tau \in \{\tau_0, \ldots, \tau_r\}$ **do**
  if $\tau == \tau_0$ **then**
    Initialization (Sec. 4.1);
  end
  while E_D *(Eq. 2) decreases* **do**
    for $n \in \{1, \ldots, N\}$ **do**
      Perform a gradient step $\rho_{\boldsymbol{\lambda}_n}$ along ∇E_W relative to X_n (Sec. 3.2);
    end
    for $n \in \{1, \ldots, N\}$ **do**
      Perform a gradient step $\rho_{\mathscr{D}}$ along ∇E_A relative to X_n (Sec. 3.3);
    end
  end
end

has been selected. As illustrated in Fig. 6 in the case of a toy 2D point set, this initialization strategy has the nice property that it tends to select atoms on the convex hull of the input point set, which ensures that the non-atom points can indeed be expressed as a convex combination of the atoms, hence leading to accurate initial barycentric approximations. As for the barycentric weights, these are uniformly initialized (i.e. to $1/m$).

4.2 Multi-scale optimization algorithm

In real-life data, persistence diagrams tend to contain many low-persistence features, which essentially encode the noise in the data (see Fig. 1, right). In this section, we present a multi-scale optimization strategy which addresses this issue by prioritizing the optimization on the most persistent pairs, which correspond to the most salient features of the data. As detailed in Sec. 6.2, this strategy leads the optimization to solutions of improved energy in comparison to a naive (non-multi-scale) approach.

Our multi-scale strategy consists in iterating our optimization procedure by progressively increasing the *resolution* (in terms of persistence) of the input diagrams. This is inspired by the progressive strategy by Vidal et al. [102] for the problem of Wasserstein barycenter optimization. Specifically, given an input diagram X, let Δf be the span in scalar values in the corresponding ensemble member f (i.e. $\Delta f = \max_{v \in \mathscr{M}} f(v) - \min_{v \in \mathscr{M}} f(v)$). Given a threshold $\tau \in [0,1]$, we note $X^\tau = \{x \in X | \; d_x - b_x \geq \tau \Delta f\}$ the version of X at resolution τ. It is a subset of X which contains persistence pairs whose relative persistence is above τ. Note that the input diagrams are not normalized by persistence, which would prevent the capture of variability in data ranges within the ensemble. Instead, we normalize the above persistence threshold, by expressing it as a fraction $\tau \in [0,1]$ of the scalar field range Δf.

Then, our multi-scale optimization will first consider the input diagrams at a resolution τ_0 and then will progressively consider finer resolutions $\tau_1, \ldots, \tau_r$ until the full diagrams are considered at $\tau_r = 0$. This multi-resolution strategy, based on a per-diagram normalized persistence threshold ($\tau \in [0,1]$) prevents diagrams from being empty in the early resolutions in case of large variations in data range within the ensemble (which would occur for instance with a per-ensemble normalization). The multi-resolution is illustrated in Fig. 7. In our experiments, we set $\tau_0 = 0.2$ and decrease τ by 0.05 at each resolution (i.e. $\tau_1 = 0.15, \tau_2 = 0.10, \tau_3 = 0.05, \tau_4 = 0$). At each resolution, the solution for the previous resolution is used as an initialization. Note that alternative strategies were considered for decreasing τ (for instance by dividing it by 2 at each resolution), but the best experimental results were obtained for the above decrease strategy.

Alg. 1 summarizes our overall approach. For each sub-optimization (i.e. weight and atom optimization), although each gradient step is guaranteed to decrease the corresponding energy (see the end of Secs. Sec. 3.2 and Sec. 3.3), this is only true for fixed assignments (between a diagram X and its barycentric approximation as well as between the barycentric approximation and the atoms). Since the assignments can change along the iterations of the optimization, the overall energy E_D (Eq. 2) may increase between consecutive iterations. Hence, pragmatic stopping conditions need to be considered. In practice, if E_D has not decreased for more than 10 iterations, we return the solutions $\boldsymbol{\lambda}^*$ and $\mathscr{D}_*$ reached by the optimization with the lowest energy E_D.

4.3 Parallelism

Our approach can be trivially parallelized with shared-memory parallelism. First, its most computationally demanding task, the N barycentric approximations of the input diagrams can be computed independently. Thus, for each barycentric approximation, we use one parallel task per input diagram. Next, the estimation of the gradient of E_W (Sec. 3.2) is done on a per input diagram basis, independently. Thus, we use one parallel task per input diagram. Regarding the estimation of the gradient of E_A (Sec. 3.3), given a barycentric approximation $Y(\mathscr{D})$ of an input diagram X, each of its points $y^j(\mathscr{D})$ defines independently a pointwise version of the gradient of the atom energy (see the last paragraph of Sec. 3.3). Thus, we use one parallel task per point $y^j(\mathscr{D})$ of a barycentric approximation $Y(\mathscr{D})$ of an input diagram X.

5 Applications

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

5.1 Data reduction

Like any data representation, persistence diagrams can benefit from lossy compression. This can be useful in in-situ [7] use-cases, where time-steps are represented on permanent storage with topological signatures [17]. In such scenarios, lossy compression is useful to facilitate the manipulation (i.e. storage and transfer) of the resulting ensemble of persistence diagrams. We present now an application to data reduction where the input ensemble of persistence diagrams is compressed, by only storing to disk:
(i) the Wasserstein dictionary of persistence diagrams $\mathscr{D}_*$ and
(ii) the N barycentric weights $\boldsymbol{\lambda}_1^*, \ldots, \boldsymbol{\lambda}_N^*$.

The compression quality can be controlled with two input parameters *(i)* the number of atoms m and *(ii)* the maximum S_m

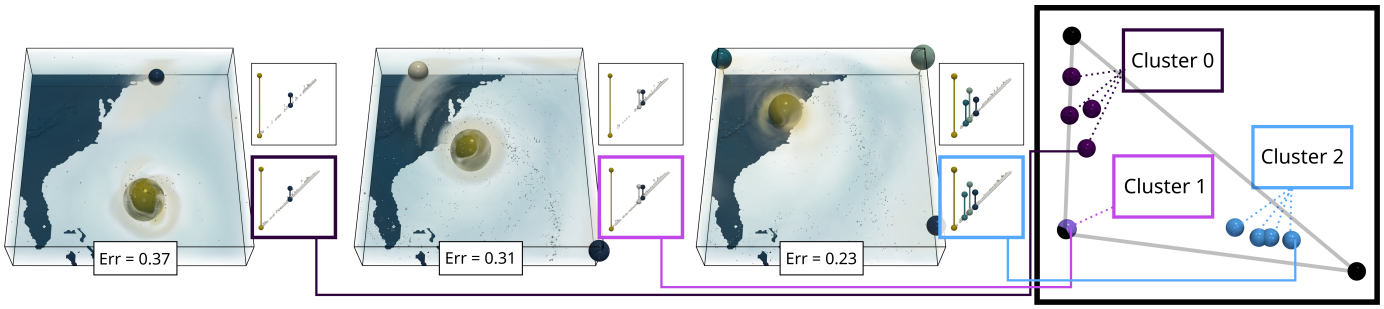


Fig. 8. Visual comparison (left) between the input persistence diagrams (top insets, saddle-maximum persistence pairs only) and our compressed diagrams (bottom insets, Sec. 5.1, saddle-maximum persistence pairs only) for three members of the *Isabel* ensemble (one member per ground-truth class). For each member, the sphere color encodes the matching between the input and the compressed diagrams (for the meaningful persistence pairs, above 10% of the function range). This visual comparison shows that the main features of the diagrams (encoding the main hurricane wind gusts in the data) are well preserved by the data reduction, especially for the members coming from the cluster 2, for which a lower relative reconstruction error (*Err*) can be observed. The planar overview of the ensemble (right) generated by our dimensionality reduction (Sec. 5.2) enables the visualization of the relations between the different diagrams of the ensemble. Specifically, this illustration shows a larger disparity for two clusters.

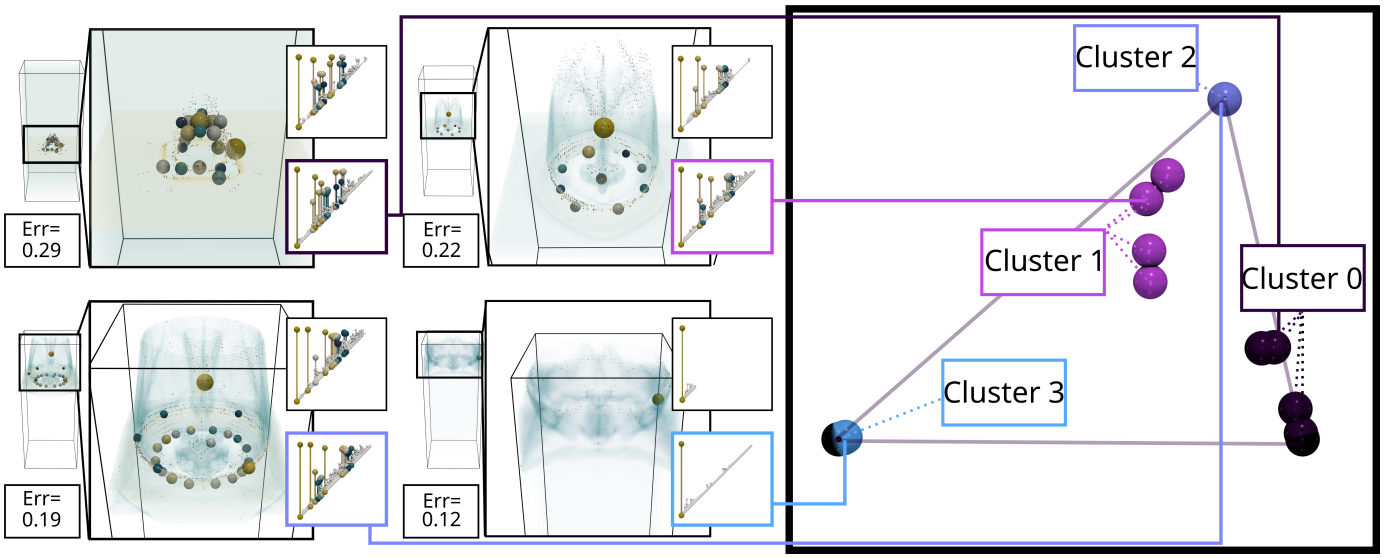


Fig. 9. Visual comparison (left) between the input persistence diagrams (top insets) and our compressed diagrams (bottom insets, Sec. 5.1) for four members of the *Ionization front (3D)* ensemble (one member per ground-truth class). The color encoding is the same as in Fig. 8. This visual comparison shows that the main features of the diagrams (the extremities of the ionization front) are well preserved by the data reduction, especially for the members coming from the clusters 2 and 3, for which a lower reconstruction error (*Err*) can be observed. The planar overview of the ensemble (right) generated by our dimensionality reduction (Sec. 5.2) enables the visualization of the relations between the diagrams of the ensemble. Specifically, it shows a larger disparity for the clusters 0 and 1 (spread out purple and pink spheres), which are also the most difficult to reconstruct.

of the total size of the atoms (i.e. $\sum_{i=1}^{m} |a_i|$). The reconstruction error (given by the energy E_D, Eq. 2.) will be minimized for large values of both parameters, while the compression factor will be maximized for low values. In our data reduction experiments, we set the number of atoms m to the number of ground-truth classes of each ensemble, as documented in the ensemble descriptions [79]. Moreover, we set S_m to $c_f^{-1} \sum_{i=1}^{N} |X_n|$, where c_f is a target compression factor and $|X_n|$ is the number of non-diagonal points in the input diagram X_n (see Sec. 6.2 for a quantitative evaluation).

Fig. 8 (left) provides a visual comparison between the diagram compressed with this strategy (bottom insets) and the original diagram (top insets), for three members of the *Isabel* ensemble. This experiment shows that diagrams can be significantly compressed ($c_f = 5.49$), while still faithfully encoding the main features of the data. Fig. 9 (left) provides a similar visual comparison for the *Ionization front (3D)* ensemble ($c_f = 2.9$).

We have applied our data reduction approach to topological clustering [102], where the main trends within the ensemble are identified by clustering the ensemble members based on their persistence diagrams. For the large majority of our test ensembles, the outcome of the clustering algorithm [102] was identical when used with the input diagrams or our compressed diagrams (Sec. 6.3 documents a counter-example). This confirms the viability and utility of our data reduction scheme.

5.2 Dimensionality reduction

Our framework can also be used to generate low-dimensional layouts of the ensemble, for its global visual inspection. Specifically, we generate 2D planar layouts by using $m = 3$ atoms and by embedding our Wasserstein dictionary $\mathscr{D}_*$ as a triangle in the plane, such that its edge lengths are equal to the Wasserstein distances between the corresponding atoms. Next, each diagram X of the input ensemble is embedded as a point in this triangle by using its barycentric weights $\boldsymbol{\lambda}^*$ as barycentric coordinates.

As illustrated in Figs. 8 (right) and 9 (right), our dimensionality reduction provides a planar overview of the ensemble which groups

together diagrams which are close in terms of Wasserstein distances. Specifically, in both examples, the ground-truth classification of the ensemble is visually respected: the points of a given class (same color) indeed form a distinct cluster in the planar view.

6 Results

This section presents experimental results obtained on a computer with two Xeon CPUs (3.2 GHz, 2x10 cores, 96GB of RAM). The input persistence diagrams were computed with the *Discrete Morse Sandwich* algorithm [41]. We implemented our approach in C++ (with OpenMP), as modules for TTK [98], [12]. Experiments were ran on the benchmark of public ensembles [78] described in [79], which includes simulated and acquired 2D and 3D ensembles from previous work and past SciVis contests [68]. The considered type of persistence pairs (i.e. the index of the corresponding critical points, Sec. 2.1) was selected on a per-ensemble basis, depending on the features of interest present in the ensemble. All types of pairs (i.e. minimum-saddle pairs, saddle-saddle pairs and saddle-maximum pairs) were considered for the following ensembles: *Cloud processes*, *Isabel*, *Starting Vortex*, *Sea Surface Height*, *Vortex Street*. Only the persistence pairs including extrema were considered for the ensembles *Ionization front (2D)* and *Ionization front (3D)*. Finally, only the persistence pairs containing maxima were considered for the remaining ensembles: *Asteroid Impact*, *Dark Matter*, *Earthquake*, *Viscous Fingering*, *Volcanic Eruptions*.

6.1 Time performance

The most computationally expensive part of our approach is the computation of the N Wasserstein barycenters, for which we use the algorithm by Vidal et al. [102]. Each iteration of barycenter optimization approximatively requires $\mathcal{O}(mK^2)$ steps in practice (where K is the size of the augmented diagrams, cf. Sec. 2.2). As discussed in Sec. 4.3, each barycenter is computed in parallel. The evaluations of the gradient of the weight energy (Sec. 3.2) and the atom energy (Sec. 3.3) both require $\mathcal{O}(NmK)$ steps. As described in Secs. 3.2 and 3.3, both evaluations can be run in parallel.

Tab. 1 evaluates the practical time performance of our multi-scale algorithm for the optimization of the Wasserstein dictionary. In sequential, the runtime is roughly a function of the number of input diagrams (N) as well as their average size ($|X|$). The parallelization of our algorithm (with 20 cores) induces a significant speedup (up to 18 for the largest ensembles), resulting in an average computation time below 5 minutes, which we consider to be an acceptable pre-processing time, prior to interactive exploration. In comparison to the principal geodesic analysis of persistence diagrams (Tab. 1 of [80]), on a per ensemble basis, our approach is 1.56 times faster on average (on the same hardware).

6.2 Framework quality

Tab. 2 reports compression factors and average relative reconstruction errors for our application to data reduction (Sec. 5.1). For each ensemble, the compression factor is the ratio between the storage size of the input diagrams and that of the Wasserstein dictionary $\mathscr{D}_*$ (the m atoms, of average size $|a|$, plus the N sets of barycentric weights). The relative reconstruction error is obtained by considering the Wasserstein distance between an input diagram and its barycentric approximation, divided by the maximum pairwise Wasserstein distance observed in the input ensemble. Then this relative reconstruction error is averaged over all the diagrams of the ensemble. Tab. 2 compares a naive optimization (Sec. 3) to our multi-scale strategy (Sec. 4.2). Specifically, for a given ensemble, the same target compression factor was used for both approaches (by imposing the same upper boundary on the total size of the atoms, Sec. 5.1). Tab. 2 shows that our multi-scale strategy (Sec. 4.2) enables the optimization to progress towards better solutions, as assessed by the improvement in reconstruction error of 30% on average. In comparison to the principal geodesic analysis of persistence diagrams (Appendix D of [80]), for the same compression factors, the error induced by our approach is on average 1.79 times larger. However, our approach is simpler, more flexible (our optimization is not subject to restrictive constraints, such as geodesic orthogonality) and slightly faster (Sec. 6.1).

Fig. 10 provides a visual comparison between the 2D layouts obtained with our approach on the *Isabel* ensemble and those obtained with two typical dimensionality reduction techniques, namely MDS [54] and tSNE [101], directly applied on the distance matrix obtained by computing the Wasserstein distance between all the pairs of input diagrams. For a given technique, to quantify

TABLE 1
Running times (in seconds) of our multi-scale algorithm (1 and 20 cores).

Dataset	N	$\|X\|$	1 core	20 cores	Speedup
Asteroid Impact (3D)	20	220	259	35	7.50
Dark matter (3D)	40	216	1,323	188	7.04
Earthquake (3D)	12	97	113	92	1.23
Ionization front (3D)	16	757	4,230	595	7.11
Isabel (3D)	12	1,310	1,609	270	5.96
Viscous Fingering (3D)	15	158	252	49	5.14
Cloud processes (2D)	12	1,176	914	64	14.28
Ionization front (2D)	16	186	145	45	3.22
Sea surface height (2D)	48	1,567	14,587	792	18.42
Starting vortex (2D)	12	125	140	24	5.83
Vortex street (2D)	45	43	1,061	241	4.40
Volcanic eruptions (2D)	12	860	2,798	706	3.96

TABLE 2
Comparison of the average relative reconstruction error (for a common target compression factor), between a naive optimization (Sec. 3) and our multi-scale strategy (Sec. 4.2). Our multi-scale algorithm improves the error by 30% on average over the naive approach.

Dataset	N	$\|X\|$	m	$\|a\|$	Factor	Error (Naive)	Error (Multi-Scale)
Asteroid Impact (3D)	20	220	4	493	2.20	0.09	0.06
Dark matter (3D)	40	216	4	215	10.87	0.15	0.12
Earthquake (3D)	12	98	3	120	3.05	0.16	0.04
Ionization front (3D)	16	757	4	1,044	2.90	0.29	0.20
Isabel (3D)	12	1,310	3	1,049	5.49	0.34	0.37
Viscous Fingering (3D)	15	158	3	41	2.78	0.15	0.11
Cloud processes (2D)	12	1,176	3	381	5.97	0.38	0.41
Ionization front (2D)	16	186	4	300	2.68	0.38	0.17
Sea surface height (2D)	48	1,567	4	534	20.98	0.54	0.61
Starting vortex (2D)	12	125	2	379	1.98	0.22	0.09
Vortex street (2D)	45	43	5	75	5.08	0.18	0.04
Volcanic eruptions (2D)	12	860	3	345	9.97	0.20	0.20

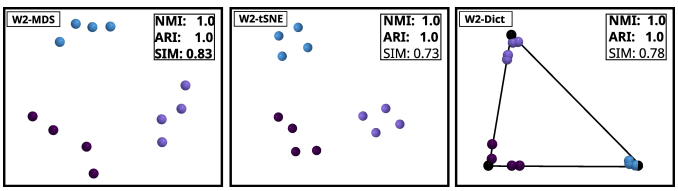


Fig. 10. Comparison between the 2D layouts obtained with our approach (*W2-Dict*) and these obtained with typical dimensionality reduction approaches (*W2-MDS* [54], *W2-tSNE* [101]) on the *Isabel* ensemble (all persistence pairs are considered). Here, the three approaches preserve well the clusters of the ensemble (NMI/ARI). As expected, *W2-MDS* provides (by design) the best metric preservation (SIM, bold). Our approach constitutes a trade-off between *W2-MDS* and *W2-tSNE*.

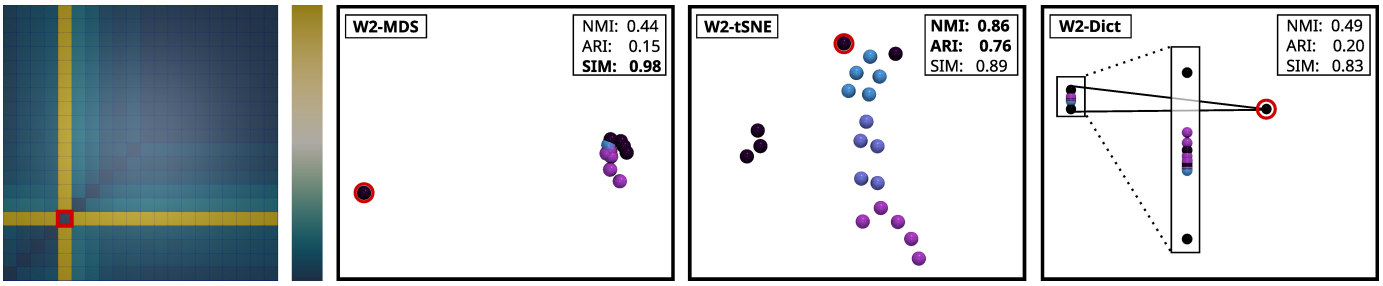


Fig. 11. Comparison between the 2D layouts obtained with our approach (*W2-Dict*) and these obtained with typical dimensionality reduction approaches (*W2-MDS* [54], *W2-tSNE* [101]) on a *challenging* ensemble. In this example (*Asteroid Impact*), the presence of an outlier (time step of the actual impact, red entry in the distance matrix, left) challenges cluster preservation. While *W2-tSNE* provides the best cluster preservation scores (NMI/ARI), it fails at visually depicting the outlier (red circle) as being far away from the other ensemble members. In contrast, *W2-MDS* and *W2-Dict* do a better job at isolating this outlier (red circle), with *W2-Dict* providing slightly improved cluster preservation scores (NMI/ARI).

TABLE 3
Detailed layout quality scores (i.e. bold: best values). On average (bottom row), our approach (*W2-Dict*) provides a trade-off between *W2-MDS* and *W2-tSNE*: it preserves the clusters (NMI/ARI) slightly better than *W2-MDS* and the metric (SIM) clearly better than *W2-tSNE*.

	NMI			ARI			SIM		
Dataset	W2-MDS	W2-tSNE	W2-Dict	W2-MDS	W2-tSNE	W2-Dict	W2-MDS	W2-tSNE	W2-Dict
Asteroid Impact (3D)	0.44	**0.86**	0.49	0.15	**0.76**	0.20	**0.91**	0.89	0.83
Dark Matter (3D)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.91**	0.68	0.84
Earthquake (3D)	**0.65**	0.61	**0.65**	0.37	**0.44**	0.37	**0.96**	0.72	0.91
Ionization Front (3D)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.86**	0.71	0.71
Isabel (3D)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.83**	0.73	0.78
Viscous Fingering (3D)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.91**	0.64	0.89
Cloud Processes (2D)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.79**	0.55	0.68
Ionization Front (2D)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.78	0.74	**0.83**
Sea Surface Height (2D)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.85**	0.73	0.79
Starting Vortex (2D)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.88**	0.72	0.84
Street Vortex (2D)	**1.00**	0.14	**1.00**	**1.00**	-2e-4	**1.00**	0.89	**0.96**	0.81
Volcanic Eruption (2D)	0.66	**1.00**	0.66	0.41	**1.00**	0.41	**0.81**	0.74	0.74
Average	0.896	0.884	**0.900**	0.827	**0.849**	0.832	**0.870**	0.734	0.804

its ability to preserve the *structure* of the ensemble, we run k-means in the 2D layouts and evaluate the quality of the resulting clustering (given the ground-truth [79]) with the normalized mutual information (NMI) and adjusted rand index (ARI). To quantify its ability to preserve the *geometry* of the ensemble, we report the metric similarity indicator SIM [80], which evaluates the preservation of the Wasserstein metric in the 2D layout. All these scores vary between 0 and 1, with 1 being optimal. In Fig. 10, the three approaches preserve well the clusters of the ensemble (NMI/ARI) and our approach provides a trade-off between MDS and tSNE in terms of metric preservation (SIM). Fig. 11 provides another visual comparison on a *challenging* ensemble (*Asteroid Impact*). There, the presence of an outlier (time step of the actual impact) challenges cluster preservation. While tSNE provides the best cluster preservation (NMI/ARI), it fails at visually depicting the outlier (red circle) as being far away from the other ensemble members. In contrast, *MDS* and our approach do isolate this outlier (red circle), with our approach providing slightly improved cluster preservation (NMI/ARI) over *MDS*. This illustrates the viability of our dimensionality reductions for outlier detection. Appendix C extends this visual analysis to all our test ensembles.

Tab. 3 extends our quantitative analysis to all our ensembles. MDS preserves well the metric (high SIM), at the expense of mixing ground-truth classes (low NMI/ARI). tSNE behaves symmetrically (higher NMI/ARI, lower SIM). Our approach provides a trade-off between the extreme behaviors of MDS and tSNE, with a cluster preservation slightly improved over MDS (NMI/ARI), and a clearly improved metric preservation over tSNE (SIM).

Fig. 12 reports the evolution of the normalized energy E_D along the optimization for all test ensembles, for the naive optimization strategy (Sec. 3), by using a number of atoms equal to the number of ground-truth classes (cf. our application to data reduction, Sec. 5.1). In this figure, the energy is normalized on a per ensemble basis, based on its initial value. This figure shows that the energy does decrease for most ensembles, but still with large oscillations due to the non-convex nature of the dictionary energy E_D. In contrast, the energy evolution with our multi-scale strategy (Fig. 13) results in much less oscillations, which indicates the ability of this strategy to help the optimization explore in a more stable manner the locally convex areas of the energy (Appendix D discusses a counter-example). Specifically, in Fig. 13, one can observe sequences of discontinuous decrease patterns, characterized by an abrupt drop followed by a plateau. Each of these patterns corresponds to one persistence scale of our multi-scale strategy (this is particularly apparent on the *Cloud Processes* ensemble).

Fig. 14 provides a closer comparison between the two strategies on a selection of four ensembles. The *Cloud Processes* ensemble is an example where the naive optimization reaches a solution of slightly lower energy. For the other ensembles, our multi-scale strategy leads to solutions of much lower energy, visually confirming the conclusions of Tab. 2. In this figure, one can also observe the characteristic decrease patterns discussed above, particularly apparent on the *Ionization Front (3D)* ensemble, which correspond to the distinct scales of our multi-scale strategy.

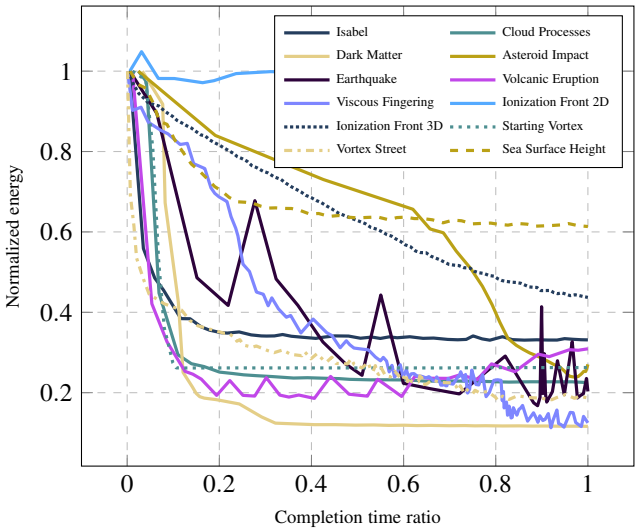


Fig. 12. Evolution of the (normalized) energy E_D along the optimization, with a naive optimization (Sec. 3), for all our test ensembles.

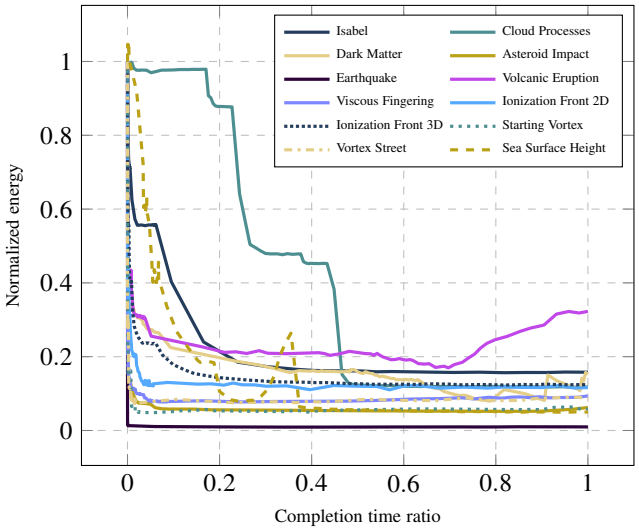


Fig. 13. Evolution of the (normalized) energy E_D along the optimization, with our multi-scale strategy (Sec. 4.2), for all our test ensembles.

6.3 Limitations

Similarly to other optimization problems based on topological descriptors [79], [80], [100], [102], our energy is not convex. Additionally, as shown in Fig. 12, the interleaving of the weight optimization (Sec. 3.2) with atom optimization (Sec. 3.3) can even lead to oscillations in the energy. As discussed in Sec. 6.2, our multi-scale strategy (Sec. 4.2) greatly mitigates both issues, with a more stable optimization than a naive approach (Sec. 3), which leads to relevant solutions which are exploitable in the applications (Sec. 5). However, we have found one example in our test ensembles (the *Sea Surface Height* ensemble), where our multi-scale strategy reached solutions which were arguably worse than these obtained with a naive solution, as described in details in Fig. 15. In this example, the most persistent features in the diagrams are not particularly discriminative for the separation of the ground-truth classes. On the contrary, the variations between these classes seem mostly encoded by the *low* persistence features:

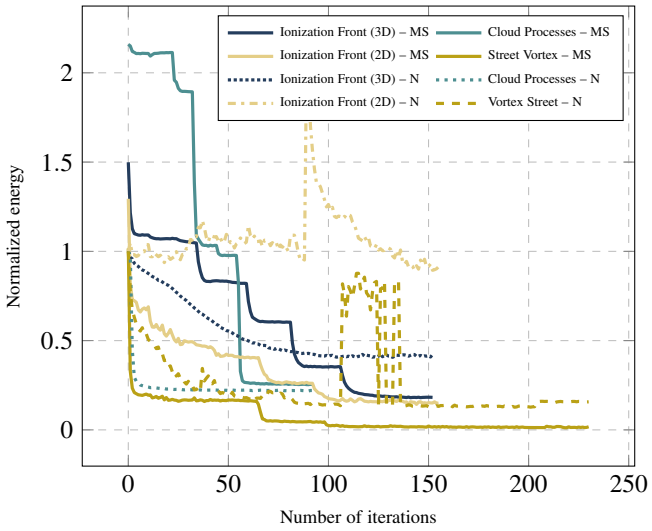


Fig. 14. Comparison of the evolutions of the (normalized) energy E_D between the naive optimization (Sec. 3, *N*, dashed curves) and our multi-scale strategy (Sec. 4.2, *MS*, solid curves) for four ensembles. For this experiment, the energy has been normalized with regard to the initial energy of the naive optimization. The *Cloud Processes* ensemble is an example where the naive optimization reaches a solution of slightly lower energy. For the other three ensembles, our multi-scale strategy leads to solutions of much lower energy, through a sequence of characteristic, discontinuous decrease patterns (abrupt drop followed by a plateau) corresponding to the five persistence scales of our multi-scale strategy.

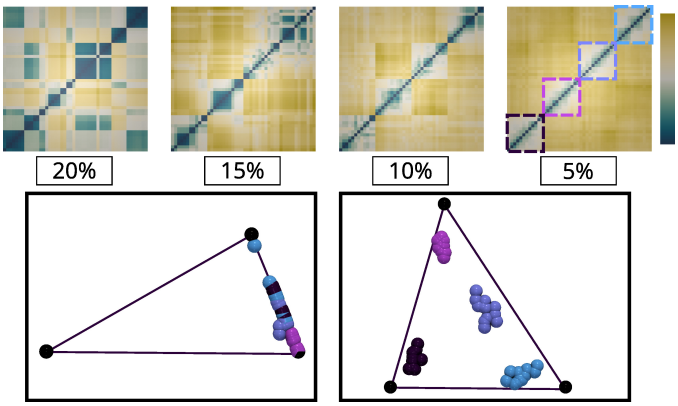


Fig. 15. Counter-example for our multi-scale strategy (*Sea Surface Height* ensemble). Top: Wasserstein distance matrices for the first four persistence scales of our multi-scale strategy. The ground-truth classes only start to become visible in the distance matrix between the third and fourth scale (dashed sub-matrices in the fourth scale). As a result, our multi-scale strategy is attracted in the first scales towards a local minimum of the energy which does not encode well the ground-truth classes (dimensionality reduction, bottom left). In contrast, the naive optimization manages to reach a solution which separates well the ground-truth classes (dimensionality reduction, bottom right).

in Fig. 15 clear separations in the distance matrices between the ground-truth classes only start to occur in the latest persistence scales (dashed sub-matrices, top right inset). This counter-intuitive observation goes against the rule of thumb traditionally used in topological data analysis, which states that the most persistent pairs encode the most important features in the data. For this example, when applying our framework to dimensionality reduction, the non-discriminative aspect of the early persistence scales eventually lead our multi-scale strategy towards a local minium which does not separate the ground-truth classes well (planar layout, bottom left) in

comparison to the naive strategy (planar layout, bottom right). Thus, for this ensemble, we reported dimensionality reduction results (Tab. 3, Appendix C) obtained with the naive optimization. In general, this means that when users are confronted with ensembles where the most persistent pairs are not the most responsible for data variability (hence class separation), the naive optimization may need to be considered additionally as it might provide solutions which better encode the ground-truth classes.

Finally, as detailed in Appendix D, the presence of clear outliers can also challenge our optimization, especially when the selected number of atoms equals the number of ground-truth classes. Then, in this case, the best dictionary encoding will consequently be obtained by increasing the number of atoms, specifically, by considering that each outlier forms a singleton class.

7 Conclusion

In this paper, we presented an approach for the encoding of linear relations between persistence diagrams, given the Wasserstein metric. Specifically, we introduced a dictionary based representation of an ensemble of persistence diagrams, inspired by previous work on histograms [90]. We first documented a naive optimization, which interleaves the optimization of the barycentric weights of the input diagrams with the optimization of the atoms of the dictionary (Sec. 3). Then, we presented a multi-scale strategy (Sec. 4.2) leading to more stable optimizations and relevant solutions (Sec. 6.2). We demonstrated the utility of our contributions in applications (Sec. 5) to data reduction and dimensionality reduction, where the visualizations generated by our framework enable the visual identification of the main trends in the ensembles (Figs. 8, 9), and the quick identification of outliers (Fig. 11). In contrast to previous work on persistence diagram encoding [80], our framework is simpler, less constrained and slightly faster in practice.

A natural direction for future work is the extension of our framework to other topological descriptors such as Reeb graphs or Morse-Smale complexes. However, this requires the definition of key geometrical tools, such as geodesic or barycenter computation algorithms, which is still an active research problem. We believe our framework for the dictionary encoding of persistence diagrams is an interesting practical step for the analysis of collections of persistence diagrams. In the future, we will continue our investigation of the adaptation of tools from optimal transport to the analysis of ensembles of topological descriptors, as we believe it can become a key solution in the long term for the advanced analysis of large-scale ensembles.

Acknowledgments

This work is partially supported by the European Commission grant ERC-2019-COG *"TORI"* (ref. 863464, https://erc-tori.github.io/).

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 PacificViz*, 2015.

[3] 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.

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

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

[6] 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.

[7] 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.

[8] D. P. Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21(1):152–171, 1981.

[9] 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 Comp. Chem.*, 2018.

[10] 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.

[11] S. Biasotti, D. Giorgio, M. Spagnuolo, and B. Falcidieno. Reeb graphs for shape analysis and applications. *TCS*, 2008.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] N. Brown, R. Nash, P. Poletti, G. Guzzetta, M. Manica, A. Zardini, M. Flatken, J. Vidal, C. Gueunet, E. Belikov, J. Tierny, A. Podobas, W. D. Chien, S. Markidis, and A. Gerndt. Utilising urgent computing to tackle the spread of mosquito-borne diseases. In *UrgentHPC@SC*, 2021.

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

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

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

[21] M. E. Celebi, H. A. Kingravi, and P. A. Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Syst. Appl.*, 2013.

[22] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. In *SoCG*, 2005.

[23] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. Lipschitz Functions Have Lp-Stable Persistence. *FCM*, 2010.

[24] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*. 2013.

[25] M. Cuturi and A. Doucet. Fast computation of wasserstein barycenters. In *ICML*, 2014.

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

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

[28] H. Doraiswamy and V. Natarajan. Computing Reeb Graphs as a Union of Contour Trees. *IEEE TVCG*, 2013.

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

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

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

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

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

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

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

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

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

[38] D. Guenther, R. Alvarez-Boto, J. Contreras-Garcia, J.-P. Piquemal, and J. Tierny. Characterizing Molecular Interactions in Chemical Systems. *IEEE TVCG*, 2014.

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

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

[41] P. Guillou, J. Vidal, and J. Tierny. Discrete morse sandwich: Fast computation of persistence diagrams for scalar data–an algorithm and a benchmark. *IEEE TVCG*, 2023.

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

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

[44] A. Gyulassy, P. Bremer, and V. Pascucci. Shared-Memory Parallel Computation of Morse-Smale Complexes with Improved Accuracy. *IEEE TVCG*, 2019.

[45] A. Gyulassy, M. A. Duchaineau, V. Natarajan, V. Pascucci, E. Bringa, A. Higginbotham, and B. Hamann. Topologically Clean Distance Fields. *IEEE TVCG*, 2007.

[46] 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*, 2016.

[47] 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.

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

[49] C. R. Johnson and A. R. Sanderson. A next step: Visualizing errors and uncertainty. *IEEE CGA*, 2003.

[50] L. Kantorovich. On the translocation of masses. *AS USSR*, 1942.

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

[52] M. Kerber, D. Morozov, and A. Nigmetov. Geometry helps to compare persistence diagrams. *ACM J. of Experimental Algorithmics*, 22, 2017.

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

[54] J. B. Kruskal and M. Wish. Multidimensional Scaling. In *SUPS*, 1978.

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

[56] D. E. Laney, P. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE TVCG*, 2006.

[57] M. Li, S. Palande, L. Yan, and B. Wang. Sketching merge trees for scientific visualization. In *IEEE TopoInVis*, 2023.

[58] T. Liebmann and G. Scheuermann. Critical Points of Gaussian-Distributed Scalar Fields on Simplicial Grids. *CGF*, 2016.

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

[60] 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.

[61] 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 PacificViz*, 2016.

[62] M. Mirzargar, R. Whitaker, and R. Kirby. Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE TVCG*, 20(12):2654–2663, 2014.

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

[64] J. Munkres. Algorithms for the assignment and transportation problems. *J. of SIAM*, 1957.

[65] F. Nauleau, F. Vivodtzev, T. Bridel-Bertomeu, H. Beaugendre, and J. Tierny. Topological Analysis of Ensembles of Hydrodynamic Turbulent Flows – An Experimental Study. In *IEEE LDAV*, 2022.

[66] 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.

[67] M. Olejniczak and J. Tierny. Topological Data Analysis of Vortices in the Magnetically-Induced Current Density in LiH Molecule. *Physical Chemistry Chemical Physics*, 2023.

[68] Organizers. The IEEE SciVis Contest. http://sciviscontest.ieeevis.org/, 2004.

[69] M. Otto, T. Germer, H.-C. Hege, and H. Theisel. Uncertain 2D vector Field Topology. *CGF*, 2010.

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

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

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

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

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

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

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

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

[78] M. Pont, J. Vidal, J. Delon, and J. Tierny. Wasserstein Distances, Geodesics and Barycenters of Merge Trees – Ensemble Benchmark. https://github.com/MatPont/WassersteinMergeTreesData, 2021.

[79] M. Pont, J. Vidal, J. Delon, and J. Tierny. Wasserstein Distances, Geodesics and Barycenters of Merge Trees. *IEEE TVCG*, 2022.

[80] M. Pont, J. Vidal, and J. Tierny. Principal geodesic analysis of merge trees (and persistence diagrams). *IEEE TVCG*, 2023.

[81] K. Pöthkow and H.-C. Hege. Positional Uncertainty of Isocontours: Condition Analysis and Probabilistic Measures. *IEEE TVCG*, 2011.

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

[83] K. Pöthkow, B. Weber, and H.-C. Hege. Probabilistic Marching Cubes. *CGF*, 2011.

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

[85] K. Potter, P. Rosen, and C. R. Johnson. From quantification to visualization: A taxonomy of uncertainty visualization approaches. *IFIP AICT*, 2012.

[86] 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.

[87] V. Robins, P. J. Wood, and A. P. Sheppard. Theory and Algorithms for Constructing Discrete Morse Complexes from Grayscale Digital Images. *IEEE PAMI*, 2011.

[88] 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.

[89] S. Schlegel, N. Korn, and G. Scheuermann. On the interpolation of data with normally distributed uncertainty for visualization. *IEEE TVCG*, 18(12):2305–2314, 2012.

[90] M. A. Schmitz, M. Heitz, N. Bonneel, F. Ngole, D. Coeurjolly, M. Cuturi, G. Peyré, and J.-L. Starck. Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning. *SIAM Journal on Imaging Sciences*, 2018.

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

[92] 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.

[93] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *American Mathematical Monthly*, 1967.

[94] M. Soler, M. Petitfrere, G. Darche, M. Plainchault, B. Conche, and J. Tierny. Ranking Viscous Finger Simulations to an Acquired Ground Truth with Topology-Aware Matchings. In *IEEE LDAV*, 2019.

[95] T. Sousbie. The Persistent Cosmic Web and its Filamentary Structure: Theory and Implementations. *Royal Astronomical Society*, 2011.

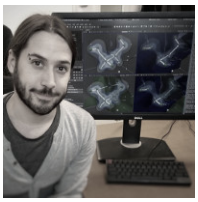
[96] A. Szymczak. Hierarchy of stable Morse decompositions. *IEEE TVCG*, 2013.

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

[98] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. The Topology ToolKit. *IEEE TVCG*, 2017. https://topology-tool-kit.github.io/.
[99] J. Tierny, A. Gyulassy, E. Simon, and V. Pascucci. Loop surgery for volumetric meshes: Reeb graphs reduced to contour trees. *IEEE TVCG*, 2009.
[100] K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer. Fréchet Means for Distributions of Persistence Diagrams. *DCG*, 2014.
[101] L. P. van der Maaten and G. Hinton. Visualizing Data Using t-SNE. *JMLR*, 2008.
[102] J. Vidal, J. Budin, and J. Tierny. Progressive Wasserstein Barycenters of Persistence Diagrams. *IEEE TVCG*, 2020.
[103] 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.
[104] D. P. Woodruff. *Sketching as a Tool for Numerical Linear Algebra*. Now Publishers, 2014.
[105] K. Wu and S. Zhang. A contour tree based visualization for exploring data with uncertainty. *IJUQ*, 2013.
[106] L. Yan, Y. Wang, E. Munch, E. Gasparovic, and B. Wang. A structural average of labeled merge trees for uncertainty visualization. *IEEE TVCG*, 2019.

Keanu Sisouk is a Ph.D. student at Sorbonne University. He received his master degree in Mathematics from Sorbonne University in 2021. His fields of interests lie on topological methods for data analysis, optimal transport, optimization methods, statistics and partial derivative equations.

Julie Delon received the Pd.D. degree in Mathematics from the Ecole Normale Supérieure Cachan in 2004. She is currently a professor at Paris-Cité University since 2013. Prior to her professor tenure, she was a CNRS researcher affiliated with TELECOM ParisTech. Her research interests lies in optimal transport, image processing, inverse problems and stochastic models for image restoration and editing.



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

APPENDIX A
∇E_W IS L-LIPSCHITZ

Proposition 1 : Let X be a persistence diagram and $\mathscr{D} = (a_1,\dots,a_m)$ a Wasserstein dictionary of persistence diagrams. If the optimal matchings are constant, then $E_W(\boldsymbol{\lambda}) = W^2\big(Y(\boldsymbol{\lambda}),X\big)$ is convex and ∇E_W is L-Lipschitz on Σ_m.

Proof. Let $\boldsymbol{\lambda} = (\lambda_1,\dots,\lambda_m) \in \Sigma_m$, $Y(\boldsymbol{\lambda}) = \big(y^1(\boldsymbol{\lambda}),\dots,y^K(\boldsymbol{\lambda})\big)$ the barycenter computed and $\phi_{\boldsymbol{\lambda},1},\dots,\phi_{\boldsymbol{\lambda},m}$ the matchings between $Y(\boldsymbol{\lambda})$ and each atom $(a_1,\dots,a_m)$:

$$\forall j \in \{1,\dots,K\},\ y^j(\boldsymbol{\lambda}) = \sum_{i=1}^{m} \lambda_i a_i^{\phi_{\boldsymbol{\lambda},i}(j)}. \tag{12}$$

We suppose the optimal matchings to be constant, thus we write $\phi_i = \phi_{\boldsymbol{\lambda},i}$. Like before we consider the following gradient:

$$\nabla y^j(\boldsymbol{\lambda}) = \begin{bmatrix} a_1^{\phi_1(j)} & \dots & a_m^{\phi_m(j)} \end{bmatrix}. \tag{13}$$

Now recall the following expression for:

$$W^2\big(Y(\boldsymbol{\lambda}),X\big) = \min_{\psi_{\boldsymbol{\lambda}}:i_X \xrightarrow{bij} i_X} \left(\sum_{j=1}^{K} \|y^j(\boldsymbol{\lambda}) - x^{\psi_{\boldsymbol{\lambda}}(j)}\|^2 \right). \tag{14}$$

This minimum is always attained, and with the hypothesis on the optimal matchings we write $\psi = \psi_{\boldsymbol{\lambda}}$. Thus we rewrite:

$$W^2\big(Y(\boldsymbol{\lambda}),X\big) = \sum_{j=1}^{K} \|y^j(\boldsymbol{\lambda}) - x^{\psi(j)}\|^2 = \sum_{j=1}^{K} \| \sum_{i=1}^{m} \lambda_i (a_i^{\phi_i(j)} - x^{\psi(j)})\|^2. \tag{15}$$

$W^2\big(Y(\boldsymbol{\lambda}),X\big)$ is convex with $\boldsymbol{\lambda}$ and the gradient follows naturally:

$$\nabla W^2\big(Y(\boldsymbol{\lambda}),X\big) = 2 \sum_{j=1}^{K} \begin{bmatrix} (a_1^{\phi_1(j)} - x^{\psi(j)})^T \\ \vdots \\ (a_m^{\phi_m(j)} - x^{\psi(j))^T} \end{bmatrix} \big(y^j(\boldsymbol{\lambda}) - x^{\psi(j)}\big). \tag{16}$$

For the following part we denote $H^j = \begin{bmatrix} a_1^{\phi_1(j)} - x^{\psi(j)} & \dots & a_m^{\phi_m(j)} - x^{\psi(j)} \end{bmatrix}$. The Hessian then writes as $H = H(\boldsymbol{\lambda}) = 2\sum_{j=1}^{K} (H^j)^T H^j$. This shows that $\boldsymbol{\lambda} \mapsto W^2\big(Y(\boldsymbol{\lambda}),X\big)$ is convex. Indeed for $u \in \mathbb{R}^m$ we have:

$$u^T H u = 2\sum_{j=1}^{K} u^T (H^j)^T (H^j) u = 2\sum_{j=1}^{K} \|H^j u\|^2 \geq 0 \tag{17}$$

This also shows that ∇E_W is L-Lipschitz with $L = \|H\|$. For numerical reasons, we bound L as follows:

$$L = \|H\| = 2\left\| \sum_{j=1}^{K} (H^j)^T (H^j) \right\| \leq 2\sum_{j=1}^{K} \|(H^j)^T (H^j)\| = 2\sum_{j=1}^{K} \|H^j\|^2. \tag{18}$$

Thus for our algorithm, we consider the following gradient step:

$$\rho \leq \left[2\sum_{j=1}^{K} \|H^j\|^2 \right]^{-1}. \tag{19}$$

□

APPENDIX B
∇e_A IS L-LIPSCHITZ

Proposition 2 : Let X be a persistence diagrams and $\boldsymbol{\lambda} = (\lambda_1,\dots,\lambda_m) \in \Sigma_m$. If the optimal matchings are constant, the functions e_A are convex and ∇e_A is L-Lipschitz.

Proof. Let $U = (u_1,\dots,u_m) \in (\mathbb{R}^2)^m$, for $j \in \{1,\dots,K\}$ we have:

$$e_A(U) = \left\| \sum_{i=1}^{m} \lambda_i (u_i - x^{\psi(j)}) \right\|^2 \tag{20}$$

The gradient follows naturally:

$$\nabla e_A(U) = 2 \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_m \end{bmatrix} \big(u_i - x^{\psi(j)}\big)^T \tag{21}$$

Immediately we have the Hessian $H_j = H_{g_j}(U) = 2\boldsymbol{\lambda}\boldsymbol{\lambda}^T$, giving us the convexity of e_A and the L-Lipschitzianity of ∇e_A with $L = \|H_j\| \leq 2\|\boldsymbol{\lambda}\|^2 \leq 2m$. For numerical reasons, we consider the larger upper bound: $L \leq 4m$. Thus for our algorithm, we consider the following gradient step $\rho \leq (4m)^{-1}$. □

APPENDIX C
DIMENSIONALITY REDUCTION

Fig. 16 extends Figure 10 (main manuscript) to all our test ensembles and it confirms visually the conclusions of the table of quality scores (Table 3 of the main manuscript).

APPENDIX D
VOLCANIC ERUPTION ENSEMBLE

This appendix discusses the special case of the *Volcanic eruption* ensemble (12 members), for which a consistent energy increase can be observed in the Figure 13 of the main manuscript (normalized energy of our multi-scale optimization as a function of computation time), beyond 70% of the completion time (the optimization reaches the stopping conditions at 100%).

The ground-truth classification of this ensemble contains 3 classes [79]. However, one of these classes contains a clear outlier (light purple entry in the bottom views of Fig. 17), corresponding to a peak of activity in the eruption (see the terrain views of 4 members, bottom left of Fig. 17, including the outlier, light purple frame). The corresponding persistence diagram (light purple diagram in the aggregated birth/death space, bottom middle of Fig. 17) contains features which are significantly more persistent than the other diagrams (taken from distinct ground-truth classes, one color per class). Then, this outlier exhibits an excessively high distance to the rest of the ensemble, as illustrated in the Wassertein distance matrix (bottom right of Fig. 17, light purple entry).

The presence of this outlier challenges our optimization when using a number of atoms equal to the number of ground-truth classes (which is the default strategy documented in the main manuscript). As shown in the energy plots (Fig. 17, top), a consistent energy increase can be observed when using only 3 atoms (1 per ground-truth class, black curve). When removing the outlier, the energy evolution exhibits a more characteristic oscillating behavior (green curve). Finally, when initializing the optimization with 4 atoms (1 per class, plus 1 for the outlier), the

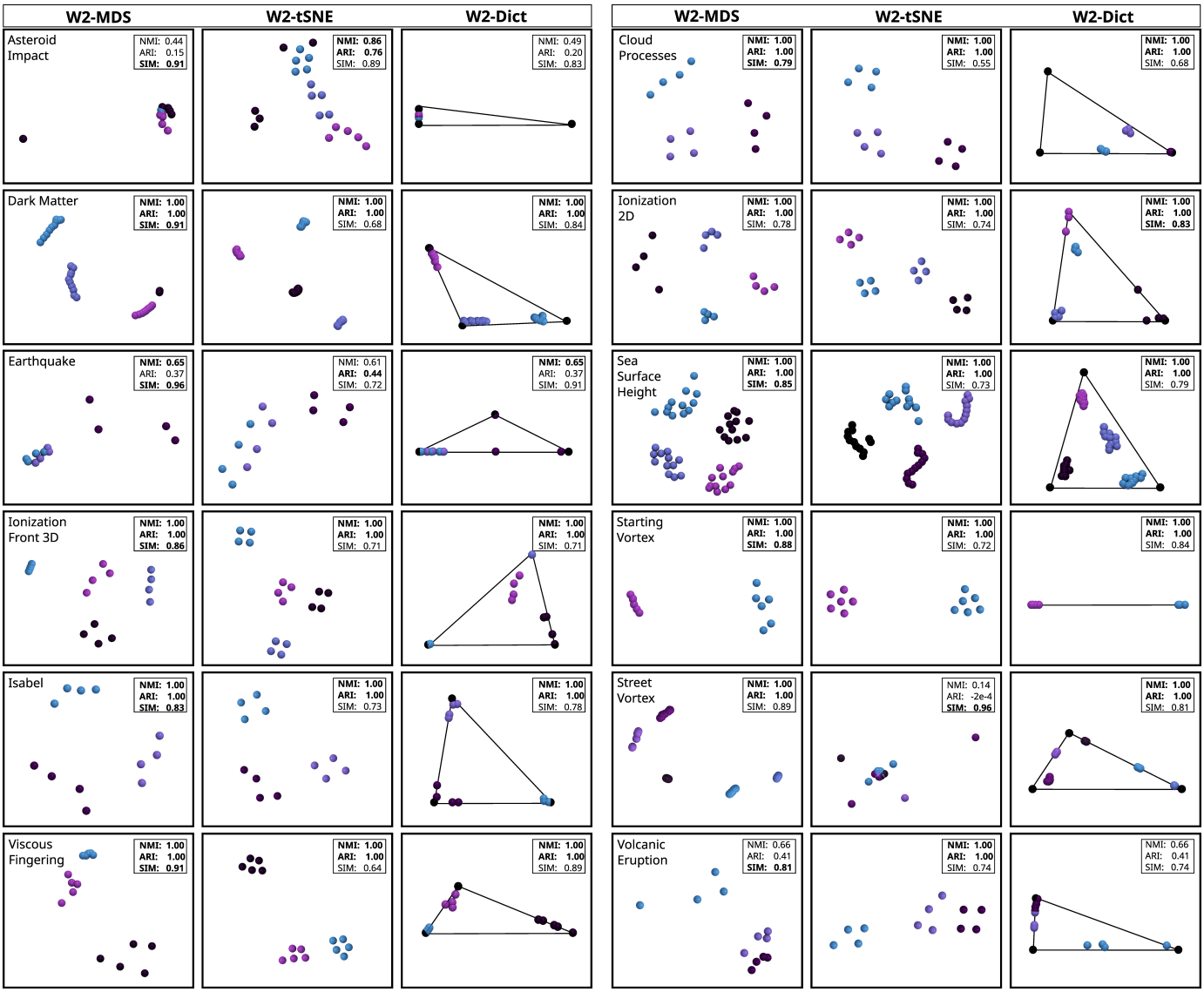


Fig. 16. Comparison of the planar layouts for typical dimensionality reduction techniques on all our test ensembles. The color encodes the classification ground-truth [79]. For each quality score, the best value appears bold. For the *Sea Surface Height* ensemble, the naive optimization procedure has been used (cf. Sec. 6.3 of the main manuscript).

optimization results in few oscillations and a consistent energy decrease (yellow curve). This indicates that the outlier member (light purple) should be interpreted as a singleton class and that the best dictionary encoding will consequently be obtained with 4 atoms.

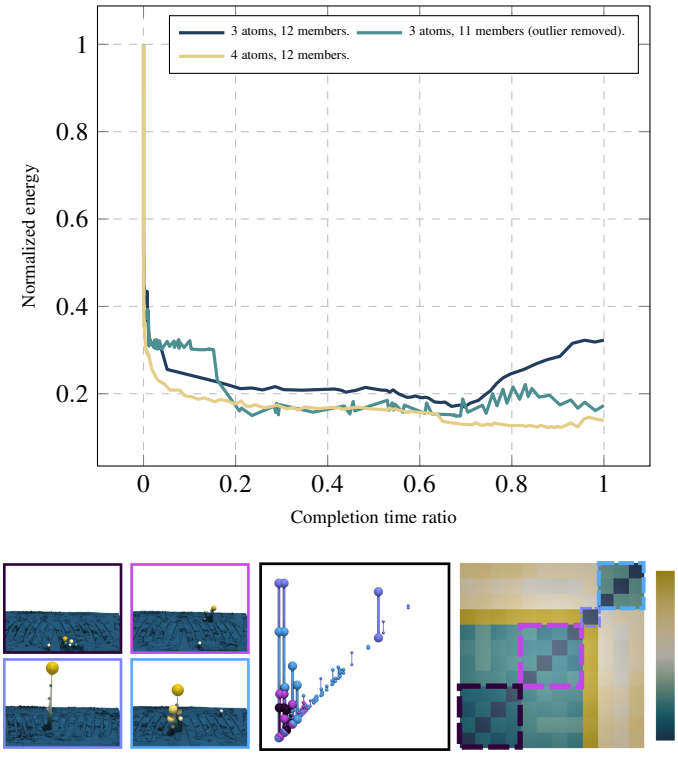


Fig. 17. Evolution of the (normalized) energy E_D along the optimization (top curves), with our multi-scale strategy, for the *Volcanic eruption ensemble*, for distinct initializations. The ground-truth classification of this ensemble contains 3 classes [79], including one outlier (light purple entry in the bottom views, from left to right: terrain view of the data, aggregated birth/death space, distance matrix). A clear energy increase can be observed when considering the entire ensemble (black curve), while a more characteristic oscillating behavior occurs when discarding the outlier (green curve). When initializing the optimization with 4 atoms (1 per class, plus 1 for the outlier), the optimization results in few oscillations and a consistent energy decrease (yellow curve).