Homology-Preserving Dimensionality Reduction via Manifold Landmarking and Tearing

Lin Yan* University of Utah
Yaodong Zhao† University of Utah
Paul Rosen‡ University of South Florida
Carlos Scheidegger§ University of Arizona
Bei Wang¶ University of Utah

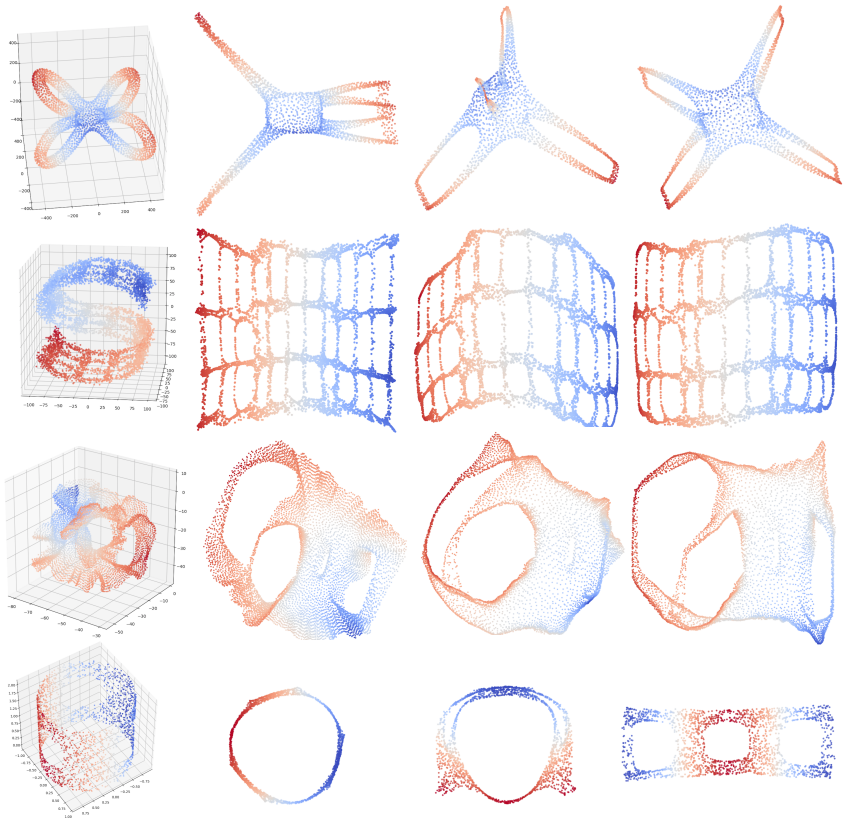

Figure 1: Homology-preserving dimensionality reduction using manifold landmarking (row 1 to 3, for *Octa*, *Fishing Net* and *4elt* datasets, respectively) and manifold tearing (row 4 for *Cylinder-3* dataset). Rows 1, 2 and 3, from left to right: original point clouds, Isomap embeddings, random landmark Isomap embeddings, and homological landmark Isomap embeddings. Row 4, from left to right: original point cloud, Isomap embeddings without tearing, with partial tearing and with the optimal tearing, respectively.

Abstract

Dimensionality reduction is an integral part of data visualization. It is a process that obtains a structure preserving low-dimensional representation of the high-dimensional data. Two common criteria can be used to achieve a dimensionality reduction: distance preservation and topology preservation. Inspired by recent work in topological data analysis, we are on the quest for a dimensionality reduction technique that achieves the criterion of homology preservation, a specific version of topology preservation. Specifically, we are interested in using topology-inspired manifold landmarking and manifold tearing to aid such a process and evaluate their effectiveness.

Index Terms: Topological data analysis, dimensionality reduction, manifold landmarking, manifold learning, high-dimensional data visualization

*e-mail: lin.yan@utah.edu
†e-mail: yaodong.zhao@utah.edu
‡e-mail: prosen@usf.edu
§e-mail: cscheid@cs.arizona.edu
¶e-mail: beiwang@sci.utah.edu

1 Introduction

Dimensionality reduction (DR) plays an important role in high-dimensional data visualization in both static and interactive settings. DR techniques could be classified based on structure preservation, namely, distance preservation and topology preservation. The preservation of pairwise distances ensures that the low-dimensional embedding inherits the geometric properties of the original data, while the notion of topology preservation refers to the preservation of neighborhood relations between subregions.

We focus on a special version of topology preservation, namely, *homology preservation*, where we are interested in the preservation of both 0-dimensional (a.k.a. *connected components*) and 1-

dimensional (a.k.a. *loops*) homological features. Particularly in this paper, we develop DR techniques that preserve as much as possible of the 1-dimensional homological features of the data.

Motivations. Our first motivation to study homology-preserving DR is from the perspective of visualization. As technologies advance, we are collecting and generating a wide variety of large, complex, and high-dimensional datasets that demand insight-generating analysis and visualization. However, limitations on our visual systems as well as display devices have prevented us from the rapid recognition of structures beyond three dimensions. Visualization approaches therefore play an essentially role in *visually* conveying and interpreting high-dimensional structural information by utilizing low-dimensional embeddings and abstractions: from DR to visual encoding, and from quantitative analysis to interactive exploration [29]. We believe homology-preserving DR helps to expand the existing DR toolset and encodes additional structural information of high-dimensional data for visual exploration.

Our work is also inspired by the study of interesting datasets with nontrivial homology, in particular, from imaging and signal processing. In studying the space of images, Lee et al. [24] have found that the majority of high-contrast 3 by 3 patches are concentrated near a circle. Follow-up work by Carlsson et al. [3] and Xia [49] has shown that a subspace of the space of natural image patches either exhibits circular behavior or is topologically equivalent to a Klein bottle, depending on the patch size. In signal processing, using delayed window embedding, a 1-dimensional signal can be encoded into a high-dimensional point cloud for topological data analysis. Specifically, 1-dimensional homology (i.e. loop) of such a point cloud captures the periodicity of the signal [34]. For modeling processes such as Cahn-Hilliard equation [14], our work may capture 0-dimensional homology of time series data, where each time series corresponds to a trajectory in the feature space.

Contributions. The goal of this paper is to generalize topology preservation from the perspective of 0-dimensional connectivity (0-dimensional homology) to 1-dimensional connectivity (1-dimensional homology). We present examples in the paper illustrating that we can achieve homology preservation while at the same time maintaining (and sometimes even improving) the preservation of distances. Our contributions are:

- We introduce a new class of homology-preserving DR techniques that combine the strengths of landmark Isomap (L-Isomap) with the power of homology-preserving landmarks.
- For complex data such as circular manifolds, we provide a simple and fast procedure that tears those manifolds, while at the same time preserves as much homology as possible.
- We conduct experiments for homology-preserving manifold landmarking and manifold tearing to evaluate their effectiveness.

2 Related Work

Dimensionality reduction. Dimensionality reduction (DR) is the process of finding a lower dimensional representation of a high-dimensional random variable that captures its content according to some criterion [17]. DR techniques can be studied following various taxonomies. For instance, they are considered as linear (resp. nonlinear) methods if they produce low-dimensional linear (resp. nonlinear) mapping of the input high-dimensional data that preserve certain features of interest. They can be thought of as conducting convex or nonconvex optimizations, full or sparse spectral eigendecompositions, global or local structure preservation. Commonly used DR techniques include PCA [33], Isomap [43], Laplacian eigenmaps [1], LLE [36], etc., see [8] for a thorough review. We largely follow the classification from [26] in terms of distance or topology preservation (see Section 1).

Quality assessment and visualization. To assess the performance of DR techniques, different quality measures have been proposed that can be roughly classified as global- or local-based approaches. The former quantifies the preservation of local neighborhoods/subregions, and the latter studies the preservation of global shape of data. Global measures include Shepard diagram [38], stress [23], and residual variance [43] (as described in Section 3), and local measures consist of rank-based criteria such as co-ranking matrix [27], normalization independent embedding quality assessment [51], and many more [22].

Manifold landmarking. Our proposed strategy takes advantage of *manifold landmarking*, that is, finding a subset of points along the manifold that captures its structural characteristics [28]. Manifold landmarking is useful for DR, for example, in the case of landmark MDS and landmark Isomap [9, 10]. It can also be employed to generate sparse manifolds for machine learning tasks [31] or sparse matrices for semidefinite programming [47], as well as supervised learning [44].

Topology-inspired data skeletonization. Compared to existing landmarking approaches, our strategy is one that is topological in nature. Our work utilizes advances in topology-inspired data skeletonization, that is, the process of extracting the topological structure of data using a low-dimensional (e.g., 1-dimensional) representation, in order to better interpret complex, noisy, nonlinear, and high-dimensional data.

Topology-inspired data skeletonization from [19, 32] are most relevant to our framework. Ge et al. [19] give a framework to extract and simplify a 1-dimensional skeleton using the Reeb graph. Natali et al. [32] introduce a *Point Cloud Graph* as a data abstraction that is a generalization of the Reeb graph to arbitrary high-dimensional point clouds. Reeb graphs play a fundamental role in computational topological, topological data analysis and shape analysis; see [2] for a survey.

In this paper, we extract a 1-dimensional skeleton (referred to as *skeleton* for the remaining of the paper) from the input space based on an approximation of the Reeb graph. Compared to previous work, our work is novel in the sense that it utilizes such a skeleton for the purpose of landmark selection and DR.

Manifold tearing and loop detection. Most classic DR techniques do not perform well when the data manifolds contain essential (i.e., non-contractable) loops, such as cylinders, tori or spheres. The so-called loopy manifolds [30] are in fact manifolds with nontrivial homology. Such manifolds typically cannot be embedded into the target space without introducing significant distortions.

Some recent efforts have been made to detect and cut essential loops in such manifolds. Lee and Verleysen [25] introduce a two-stage tearing procedure: first, a k-nearest neighbor (kNN) graph among the point cloud sample is used to represent the underlying space; second, a minimum spanning tree (MST) or a shortest path tree (SPT) that contains no cycles is computed on the kNN graph; Finally, edges that do not generate non-contractible cycles with more than 4 edges are reintroduced to form the torn graph for downstream DR.

Our work differs from [25] significantly in the following sense. We use a topology-inspired data skeleton that consists of landmarks and landmark connections to describe all candidate essential loops, and employ a homological criterion to choose the proper loop to tear while preserving as much as possible the homological characteristics of the data. Whereas other techniques cut all or a large number of loops, we try to cut, roughly, as few loops as possible while preserving the remaining ones.

3 Homology-Based Quality Assessment

We employ a homology-based and a distance-based quality measure to assess the quality of our proposed DR techniques.

Background on homology and persistent homology. Homology was originally defined so that it can be used to tell two things (a.k.a. topological spaces) apart by examining their holes. It is a process that associates a topological space with a sequence of abelian groups called homology groups, which, roughly speaking, count and collate *holes* in a space [21]. In a nutshell, homology groups generalize a common-sense notion of connectivity. They detect and describe the connected components (0-dimensional holes), tunnels/loops (1-dimensional holes), voids (2-dimensional holes), and holes of higher dimensions in the space. Betti numbers b_i count the number of i-dimensional holes, and are used to distinguish spaces based on the connectivity across all dimensions. Formally, b_i is defined as the rank of the i-dimensional homology groups. For a torus, $b_0 = 1, b_1 = 2$ and $b_2 = 1$; this means that a torus has 1 connected component, 2 holes/loops and 1 void.

Simply put, persistent homology studies homology at multiple scales. As illustrated in Fig. 2, we begin with a point cloud X equipped with a distance metric D_X (i.e. Euclidean distance). We study the homology of a sequence of spaces formed by a union of balls of increasing radius t centered at the points. Using persistent homology, we investigate the homological changes within this growing sequence of spaces indexed by time (this is referred to as a *filtration*).

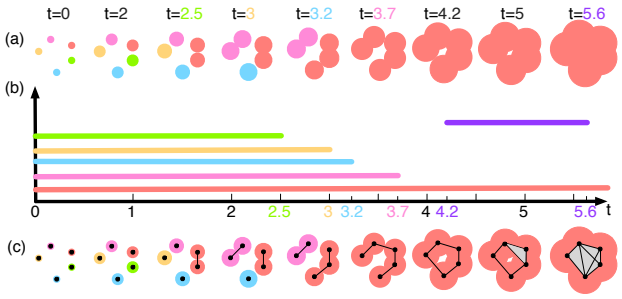

Figure 2: Computing persistent homology of a point cloud (image courtesy of [48]).

In Fig. 2(a), at time $t = 0$, each colored point is *born* (appears) as its own (connected) component. As t increases, we focus on the important events when components merge with one another to form larger components or tunnels. We begin by tracking the birth and death times of each component or tunnel as well as its lifetime in the filtration. At $t = 2.5$, the green component merges into the red component and *dies* (disappears); therefore the green component has a lifetime (i.e., *persistence*) of 2.5. At $t = 3$, the orange component merges into the pink component and dies; therefore it has a persistence of 3. Similarly, the blue component dies at $t = 3.2$ while the pink component dies at $t = 3.7$. At time $t = 4.2$, the collection of components forms a tunnel; and the tunnel disappears at $t = 5.6$. The red component born at time 0 never dies, therefore it has a persistence of ∞. We record and visualize the appearance (birth), the disappearance (death), and the persistence of homological features in the filtration via persistence diagrams [6] (Fig. 3), or equivalently, persistence *barcodes* [20] (Fig. 2(b)). A point $p = (a,b)$ in the persistent diagram of X records a homological feature that is born at time a and dies at time b. 0- and 1-dimensional persistence diagrams, denoted as $\mathrm{PD}_0(D_X)$ and $\mathrm{PD}_1(D_X)$, captures the births and deaths of components and tunnels, respectively. Equivalently in the barcode of Fig. 2(b), such a feature is summarized by a horizontal bar that begins at a and ends at b.

Computationally, the above nested sequence of spaces can be combinatorially represented by a nested sequence of simplicial complexes (i.e. collections of vertices, edges and triangles) with a much smaller footprint, as illustrated in Fig. 2(c), see [13] for computational details.

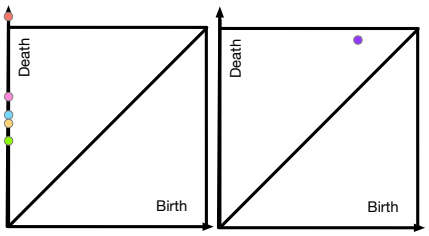

Figure 3: 0-dimensional (left) and 1-dimensional (right) persistence diagrams.The Betti Numbers are $b_0 = 5$, and $b_1 = 1$.

Homology-based quality measure: persistent Betti numbers. Betti numbers count the number of homological features and can be used as summary statistics to differentiate topological spaces. However, Betti numbers alone do not differentiate between significant and noisy homological features. We use the *persistent Betti numbers* (PB) as a way to quantify how much homological information is preserved during the DR. Let PB_i denote the number of significant i-dimensional homological features, that is, the number of points in the i-dimensional persistence diagram that is above a certain threshold that separates features from noise.

Finding a suitable threshold requires checking the separation of points in the persistence diagram (e.g., [35]). Significant features can be extracted from the persistence diagram if it has an empty band (of a certain width) parallel to the diagonal that does not contain any points [5]. More sophisticated methods from statistics based on bootstrapping can be used to improve the threshold estimation that separate signals from noise, based on the notion of a *confidence band* [16] (see Section 7 for a discussion). In this paper, we use PB_1 as a rule of thumb to assess the quality of DR in terms of its preservation of significant 1-dimensional features.

Distance-based quality measure: residual variance. To evaluate the fits of various DR techniques on comparable grounds, Tenenbaum et al. [43] introduce the *residual variance* (RV):

$$\mathrm{RV}(X,Y) = 1 - R^2(D_X, D_Y).$$

D_Y is the matrix of Euclidean distances in the low-dimensional embedding produced by a DR technique, and D_X is a best estimate of the intrinsic manifold distance for a given technique. In the case of Isomap, D_X corresponds to the geodesic distance matrix approximated by the graph distance matrix D_G (see Section 4.1). R is the standard Pearson correlation coefficient that measures the linear correlation between all entries of D_X and D_Y (that are reshaped into vectors).

4 Homology-Preserving Manifold Landmarking

We begin this section with a review of the nonlinear DR techniques known as the Isomap [43] and landmark Isomap (L-Isomap) [9]. We then discuss homology-preserving landmark selection based on the Reeb graph and its discrete approximation. We summarize the pipeline for a new class of techniques by combining the utilities of homology-preserving landmarks with the efficiency of landmark-based DR.

4.1 Isomap and L-Isomap

Isomap. Suppose the original input data contains N samples in D dimensions, $X \in R^{D\times N}$. Isomap embeds the points onto a lower dimensional space $Y \in R^{d\times N}$ ($d < D$) while preserving geodesic distances between all input points [43]:

1. *Construct a neighborhood graph.* A weighted, undirected k-nearest neighbor (kNN) graph G is constructed over all data points, where an edge between a point $x_i \in X$ and its neighbor

$x_j \in X$ is assigned a weight that represents the Euclidean distance between them. An appropriate k can be chosen based on the residual variance [43].

2. *Compute shortest paths*. All pairwise shortest paths between points in the KNN graph G are computed to approximate the geodesic distances between them, which leads to an $N \times N$ graph distance matrix D_G.
3. *Construct a d-dimensional embedding*. Classical MDS is applied to the above graph distance matrix D_G to obtain a low-dimensional embedding.

Isomap suffers from two computational inefficiencies: calculating the shortest-paths distance matrix and eigenvalues within MDS. The former has a complexity of $O(kN^2 \log N)$ using Dijkstra's algorithm with Fibonacci heaps, while the latter takes $O(N^3)$ [9].

L-Isomap. L-Isomap [9] addresses the two inefficiencies of Isomap at once. It is based on the landmark MDS (L-MDS) [10]:

1. *Construct a neighborhood graph* (same as in Isomap).
2. *Select landmarks*. n-points ($n \ll N$) from X are randomly selected to be landmark points.
3. *Compute shortest paths.* Compute the shortest paths from each data point to the landmarks, resulting in a $n \times N$ geodesic distance matrix.
4. *Apply L-MDS to obtain d-dimensional embedding*. First, apply classical MDS to the landmarks only, embedding them in R^d using as input the $n \times n$ distance matrix between pairs of landmarks. Second, the embedding coordinates for the remaining data points are computed based on a fixed linear transformation of their geodesic distances to the landmarks [39].
5. *PCA normalization (optional)*. This normalization is to re-orient the axes of the embedding to reflect the overall distribution, rather than the distribution of the set of landmarks; see [9, 10] for details.

L-Isomap leads to enormous savings when $n \ll N$: Computing the shortest paths in step 3 takes $O(knN \log N)$ using Dijkstra's algorithm and L-MDS in step 4 runs in $O(n^2 N)$ [9].

Here, to differentiate different versions of L-Isomap based on various landmark selection schemes, L-Isomap using randomly selected landmarks is referred to as the *random L-Isomap*, while the one using homology-preserving landmarks in the next section is called the *homological L-Isomap*.

4.2 Homology-Preserving Landmark Selection

Our work uses the idea of a data skeleton based on the Reeb graph for the purpose of landmark selection in DR. Although Reeb graphs have been used in the context of shape abstraction and comparison [19,32], to the best of our knowledge, this is the first time they are used in the context of landmark selection and DR.

In this section, we first review relevant topological notions and computations for Reeb graphs. We then describe our landmark selection algorithm using a skeleton based on the Reeb graph.

Reeb graph. Let $f : X \to R$ be a continuous function defined on a topological space X. The level set of f at a value $a \in R$ is defined as $f^{-1}(a) = \{x \in X \mid f(x) = a\}$. The *Reeb graph* [40] of f is constructed by identifying every connected component in a level set to a single point [19]. Fig. 4 gives an example of a Reeb graph of a height function on the torus.

Extracting homological skeleton. Given point cloud data, the domain can be approximated by a neighborhood graph (such as the kNN graph or the ε-neighborhood graph) among the data points, and efficient algorithms exist [19, 32] to approximate the Reeb graph in such a discrete setting.

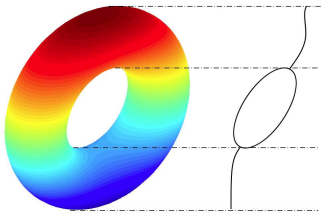

Figure 4: Reeb graph of a height function on the torus.

In this paper, we employ a mapper-based implementation to approximate the Reeb graph [37] as our homology-preserving data skeleton, referred to as the *homological skeleton* (or simply skeleton). The mapper algorithm [41] approximates the Reeb graph by considering the connected components of interval regions (i.e. $f^{-1}(a,b)$) instead of the connected components of level sets (i.e., $f^{-1}(a)$).

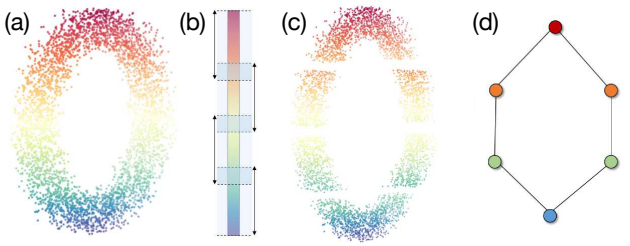

Figure 5: A mapper construction of a height function on a point cloud sampled from the torus. (a) The original point cloud X colored by the height function f (red means high, blue means low function values). (b) The cover of $\mathscr{U}$ of $f(X)$ using open intervals with $n = 4$ and $p = 0.25$. Light blue highlights overlapping regions between the intervals. (c) The cover $\mathscr{V}$ of X using clusters of points that arise from $f^{-1}(a_i, b_i)$ for each i. (d) The mapper construction as the nerve of $\mathscr{V}$.

A mapper construction is illustrated in Fig. 5. We start with a function $f : X \to R$ defined on a point cloud X, and a cover $\mathscr{U}$ of $f(X)$ consisting of finitely many open intervals $\mathscr{U} = \{(a_i, b_i)\}_{i=1}^n$. To specify such as cover, we pick two resolution parameters n and p, where n is the number of intervals and p is the percentage of overlap between a pair of adjacent intervals. *Pulling back* the cover $\mathscr{U}$ through f, that is, observing the points in $f^{-1}(a_i, b_i)$ for each i, gives an open cover of the point cloud X, which is then refined into a connected cover by splitting each cover element into various clusters using a user-defined clustering algorithm [4]. Such a cover of X is denoted as $\mathscr{V} := f^*(\mathscr{U})$, where we write $f^*(\mathscr{U})$ as the cover of X by considering the clusters of points in $f^{-1}(a_i, b_i)$ for each i. In this paper, we use DBSCAN [15] for clustering. It is a widely used density-based clustering algorithm that groups together points that are closely packed together; however the choice of clustering algorithm is not essential to our experiments.

The 1-dimensional skeleton of the nerve of $\mathscr{V}$ is considered a discrete approximation of the Reeb graph of f on X; it is referred to as the (1-dimensional) *homological skeleton* or *mapper* for the remainder of the paper. Such a skeleton is a graph with nodes representing the elements of $\mathscr{V}$, and edges representing the pairs of cover elements in $\mathscr{V}$ with nonempty intersections.

In the original mapper algorithm, the node of a skeleton represents abstractly a cover element of the point cloud, that is, a cluster of points in X. However, in our setting, we choose the *centroid* of each cluster as its representative; and all such representatives are selected as the landmarks for L-Isomap.

Filter function. The key idea behind the Reeb graph is that it explores the topology of a space by analyzing the behavior of a real-valued function defined on it [2] (referred to as a *filter func-*

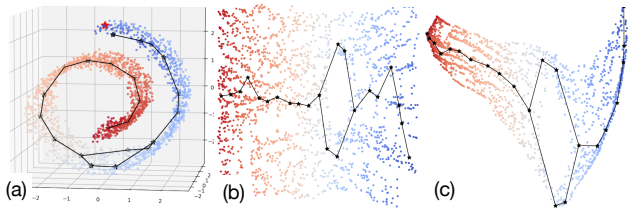

Figure 6: Generating a 2-dimensional embedding of a swiss roll with a hole. (a) The original point cloud is colored by the *distance to a base point* function (blue means low and red means high values). The base point is marked by a red star. (b) Isomap embedding. (c) Homological L-Isomap embedding. The homological skeleton together with the landmarks is highlighted in black.

tion). Reeb graphs encode topological information on data in a 1-dimensional structure, disregarding the dimension of the data in the ambient space [2]. The data can be regarded as being parameterized with respect to the filter function being used, in other words, the filter function "plays the role of the *lens*" through which we look at the properties of the data [2].

Different filter functions lead to different insights into the point cloud. It remains an open question as how to choose an appropriate filter function beyond a best practice or a guesstimation. Commonly used options include height functions, distances from the barycenter of a space, surface curvature, integral or average geodesic distances and geodesic distances from a source point in the space [2].

In this paper, we use mainly the geodesic distance from a source point as the filter function, referred to as the distance-to-the-base-point or simply the DTB function. Such a filter function has shown desirable properties in capturing the 1-dimensional homological information of the space [2, 19]. We demonstrate via experiments in Section 6 that a skeleton induced by DTB is homology-preserving for landmark-based DR; it also serves as a compact and informative summary for guiding the manifold tearing process.

Homological L-Isomap pipeline. Given a point cloud X in R^D, we now summarize our homology-preserving landmark selection pipeline and its combination with L-Isomap (referred to as the homological L-Isomap):

1. *Construct a neighborhood graph* (same as Isomap). Let G denote the resulting kNN graph.
2. *Compute a filter function f on X.* f captures certain desirable structural information of X suitable for DR. In our experiments, we use DTB as the filter function and a base point is chosen from extreme points or the barycenter. DTB can be computed based on G from a given base point.
3. *Compute a skeleton and the landmarks.* Compute a discrete approximation of the Reeb graph of f as a homological skeleton, using the mapper algorithm. The cover $\mathscr{U}$ of $f(X)$ is given by user-specified resolution parameters n and p. The nodes of the skeleton correspond to clusters of points in X; the cluster centroids are chosen as the landmarks for L-Isomap, denoted as $X_L \subseteq X$.
4. *Apply L-Isomap.* Replace randomly generated landmarks (step 2 of L-Isomap) with the homology-preserving landmarks (a.k.a. homological landmarks) X_L and apply the rest of L-Isomap algorithm.

The above pipeline is not restricted to L-Isomap. It can be easily extended to other graph-based DR techniques [50].

Extracting homological skeleton only slightly increases the asymptotic complexity of L-Isomap. Computing DTB in step 2 and the shortest paths in step 4 takes $O(kN\log N) + O(knN\log N) = O(k(n+1)N\log N)$ using Dijkstra's algorithm, and the running time of L-MDS in step 4 remains $O(n^2N)$. The running time of these two computational inefficiencies dominates the complexity of mapper algorithm and the computation of cluster centroids in step 3. In our experiments, the running time of homological L-Isomap is comparable with that of random L-Isomap.

In order to get a reasonable approximation of the Reeb graph of f as a homological skeleton, we explore parameters associated with the mapper algorithm (n and p) and the DBSCAN algorithm (see Section 6 for a discussion).

A simple example. Our pipeline is illustrated with a simple example in Fig. 6. We begin with a noisy point cloud with 1983 points sampled from a swiss roll with an irregular, hard-to-spot hole in the middle. First, a DTB filter function is computed with respect to an extremal base point. Second, a homological skeleton connecting a set of 22 landmarks is obtained using the mapper algorithm. We choose parameters $n = 10$ and $p = 0.2$ for the mapper algorithm, and apply DBSCAN with $\varepsilon = 0.8$ and a minimum sample size of 5 (i.e, $minPts = 5$). Finally, we apply L-Isomap using these homological landmarks in the skeleton (black stars). In Fig. 6, the black homological skeleton is highlighted in the input space, as well as in the Isomap and homological L-Isomap embeddings, which clearly captures the location of the significant hole in the data.

This simple example demonstrates that the homological L-Isomap has the potential to preserve as much as possible the 1-dimensional homological feature even with a smaller number of landmarks than the random L-Isomap algorithm (roughly $n = O(\sqrt{N})$). However, homological L-Isomap, in this case, does introduce distance distortion away from the single loop in the data. Surprisingly, homological L-Isomap is shown to outperform both Isomap and random L-Isomap in certain datasets using the widely accepted residual variance (see Section 6).

5 Homology-Preserving Manifold Tearing

Complex data that contain essential loops may or may not be embedded into low-dimensional space without introducing significant distortions. An alternative and complementary approach for homology-preserving DR is through manifold tearing. In this section, we give a simple and fast manifold tearing procedure guided by the homology-preserving skeleton of the point cloud data. Different from prior work, our procedure tries to cut as few loops as possible, using homology-based quality assessment (described in Section 3) while at the same time preserves as much as possible the remaining homological features.

Suppose we have a point cloud equipped with a pre-computed homology-preserving skeleton (see Section 4). Our homology preserving tearing process is as follows:

1. *Construct a neighborhood graph* (same as Isomap). The neighborhood graph is denoted as G. A slightly larger k may be chosen to account for the tearing process (optional).
2. *Tear the neighborhood graph.* A cut plane is specified based on the skeleton. Specifically, a cut location is chosen on an edge of the skeleton, which then defines a cut plane that is orthogonal to the edge to be cut. An edge that spans a pair of nodes on the opposite side of the cut plane is removed from G, resulting a new graph G'.
3. *Compute shortest paths*. Compute shortest paths between all nodes in G' and obtain a geodesic distance matrix $D_{G'}$.
4. *Apply Isomap* to $D_{G'}$.

The above process is exploratory in nature, that is, we can use different evaluation criteria of the resulting embeddings to rank the potential cut locations. In this paper, we use the number of significant homology classes as the criterion. In addition, we envision such a process to be embedded into an interactive visualization framework for DR that involves human-in-the-loop.

6 RESULTS

We present examples in this section illustrating that we can achieve homology preservation while at the same time maintaining (and sometimes improving) distance preservation.

6.1 Data

For manifold landmarking, we demonstrate our technique with datasets that contain nontrivial homology.

Octa is a point cloud sampled from a mesh of octahedron handles. The original mesh contains up to $41K$ vertices. *Fishing Net* is a synthetic, noisy point cloud sampled from a "S"-shaped surface that contains 3×11 irregular holes. *4elt* is derived from a 3-dimensional embedding of the 4elt graph used in [18]. The original graph from [46] contains 15606 nodes and 45878 edges, and is a mesh created to study fluid flow around a 4-element airfoil.

Moreover, we use two high-dimensional datasets to test the performance of homological L-Isomap. *Mice* dataset contains a 300-dimensional point cloud derived from time-varying temperature measurements of pregnant mice [42]. The point cloud is generated by a standard delayed window embedding from signal processing with a window size of 300. Our technique detects and preserves 1-dimensional homological features in the input space that capture periodicity in the signal. *Portraits* dataset contains 82 human images. The size of each image is 700×700 pixels, with 256 gray levels per pixel. All images are taken from one subject from different directions.

For manifold tearing, we use datasets that contain essential loops for demonstration. *Cylinder-3* is a point cloud sampled from a cylinder with 3 holes carved out, and *Cylinder-5* is created similarly. *Airfoil1* comes from a 2-dimensional finite element problem under the AG-Monien Matrix group from the SuiteSparse Matrix Collection [11]. *Bcsstk31* is derived from a 3-dimensional embedding of a stiffness matrix for automobile component [12].

Table 1: The number of landmarks ($|X_L|$) and other parameters for each dataset of size $|X|$. *EP* means extremal point. *BC* means barycenter.

	Octa	*Fishing Net*	*4elt*	*Mice*	*Portraits*	*Cylinder-3*	*Cylinder-5*	*Airfoil1*	*Bcsstk31*
$\|X\|$	2994	6188	7870	674	82	2000	2000	8034	8030
Dim	3	3	3	300	4900	3	3	3	3
n	20	30	10	12	4	25	25	20	30
p	0.2	0.5	0.1	0.2	0.4	0.4	0.4	0.2	0.2
ε	150	0.4	2	5	8	0.35	0.35	1.5	1.5
minPts	5	5	5	10	2	15	15	3	3
BP	BC	EP	EP	EP	EP	BC	BC	EP	EP
$\|\mathbf{X_L}\|$	**76**	**63**	**18**	**18**	**6**	**54**	**82**	**60**	**53**

6.2 Parameter Selection

In order to get a reasonable approximation of the Reeb graph using 1-dimensional mapper as a homological skeleton, there are 5 parameters to explore: n, p for mapper algorithm; *minPts*, ε for DBSCAN; and *BP* as the base point in the DTB function computation. Theoretically, Carriére et al. [4] have shown that the 1-dimensional mapper is an optimal estimator of the Reeb graph, which gives a method to automatically tune its parameters and compute confidence regions on its topological features [4]. For datasets presented in this paper, we instead follow common practices and heuristics [41] in exploring the parameter space. All parameters used in our experiments are reported in Table 1. As a rule of thumb, n is typically chosen between 5 and 20; and it is set to be 30 for datasets with a large number of 1-dimensional homological features (such as the *Fishing Net* dataset). p is typically set to be between 0.2 for dense and 0.5 for sparse point clouds; its variation does not show visible differences in our experiments. The mapper algorithm is very flexible with any domain-specific clustering algorithms [41]. For DBSCAN used in our experiments, Zhou et al. have given an adaptive framework for its parameter selection of *minPts* and ε [52]. We typically choose an extremal point as the base point *BP* in the computation of DBP function; although barycenters also work well for certain datasets such as *Octa*.

6.3 DR with Manifold Landmarking

Results and evaluation with persistence diagrams. For each dataset, 2-dimensional embeddings obtained using homological L-Isomap are compared with Isomap and random L-Isomap in Fig. 1.

Evaluation using persistent Betti numbers are illustrated by the 1-dimensional persistence diagrams in Fig. 7. We determine the number of persistent (significant) features by looking at the separation between points in the diagram. Suppose each dataset contains m persistent features in the original point cloud, then top m features with the highest persistence are marked in red within the persistence diagram associated with each embedding.

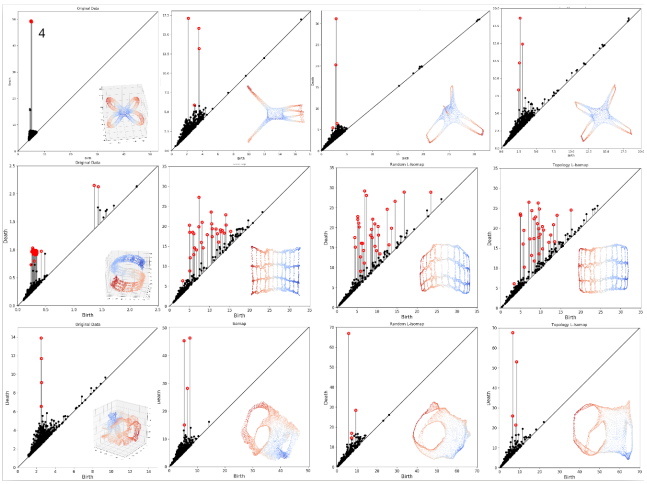

Figure 7: Homology-based quality assessment of DR for *Octa* (row 1), *Fishing Net* (row 2) and *4elt* (row 3). For each row, from left to right: persistence diagrams for the original data, Isomap embedding, random L-Isomap embedding and homological L-Isomap embedding.

For *Octa*, as shown in Fig. 7 (row 1), the original data contains 8 significant features (loops), 4 of which (colored red) correspond to the visible loops via embeddings (4 other features are the interior tunnels within each handle that are not captured by the Reeb graph). All 4 of the significant features in red are shown to be preserved using homological L-Isomap; that is, they remain well-separated from the diagonal of the persistence diagram. On the other hand, Isomap preserves 3 significant features while random L-Isomap preserves only 2. For a more detailed analysis, it is remarkable to see that using only 21 landmarks, the homological skeleton is able to summarize the homological features reasonably well (Fig. 8, left).

For *Fishing Net*, as shown in Fig. 7 (row 2), the original data has 33 significant features in the persistence diagram. Isomap and homological L-Isomap perform comparatively in terms of preserving the shape of each feature in the embeddings. However, homological L-Isomap uses only 66 points as landmarks (roughly 1% of the size of the point cloud), and is therefore more computationally efficient. Furthermore, its homological skeleton in Fig. 8 (middle) captures the underlying homological features pretty well.

For *4elt*, as shown in Fig. 7 (row 3), the original data contains 4 significant features; 3 of which are readily visible in the

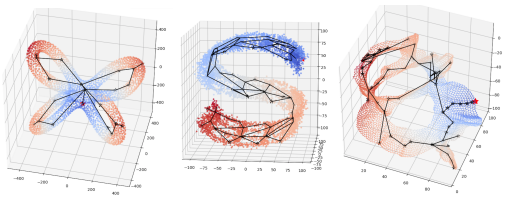

Figure 8: Homological skeletons for *Octa* (left), *Fishing Net* (middle) and *4elt* (right) using a small number of landmarks.

3-dimensional embedding of its homological skeleton in Fig. 8 (right). Both Isomap and homological L-Isomap preserve these features reasonably well, while random L-Isomap preserves only 2 of them. In addition, homological L-Isomap does slightly better in preserving the shape of a couple of features.

For *Mice*, we combine the results of DR with 1-dimensional persistence diagrams in Fig. 9(a)-(d). There are 2 significant features in the original 300-dimensional input space. Such features likely correspond to periodicity in the temperature profile of a mice relevant to circadian and ultradian rhythms respectively. Both Isomap and homological L-Isomap perform comparatively in terms of preserving the most dominant feature, while homological L-Isomap only uses a small fraction of the points as landmarks. Random L-Isomap is able to detect the significant feature but does not preserve its shape as well as the homological L-Isomap.

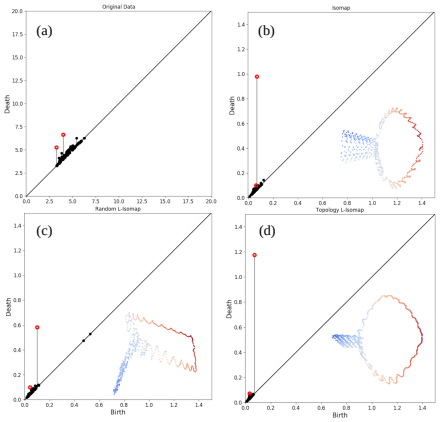

Figure 9: For *Mice*, (a) persistence diagrams for original data, (b) Isomap embedding, (c) random L-Isomap embedding and (d) homological L-Isomap embedding are combined with DR results.

For *Portraits*, as shown in Fig. 10(a)-(c), both Isomap and homological L-Isomap perform comparatively in terms of preserving the single significant loop, while homological L-Isomap only uses 6 landmark points. On the other hand, random L-Isomap fails to detect the loop using equivalent number of landmarks (see Fig. 10(b)).

Quality assessment with residual variance. We also assess the quality of DR using RV quality measure introduced in Section 3.

For *Octa*, we evaluate the quality of each embedding using the RV measure by varying the number of landmarks, see Fig. 11. As the number of chosen landmarks increases, we are interested in how well homological L-Isomap preserves distances, when compared with Isomap and random L-Isomap. For a fixed landmark size, the blue box plot corresponds to the RV measures for 20 instances of random L-Isomap, each drawing landmarks randomly from a fixed point cloud.

A surprising observation is that homological L-Isomap outperforms Isomap and random L-Isomap in terms of distance preservation, when the number of landmarks is small (below 100). In fact, homological L-Isomap beats random L-Isomap with just 21

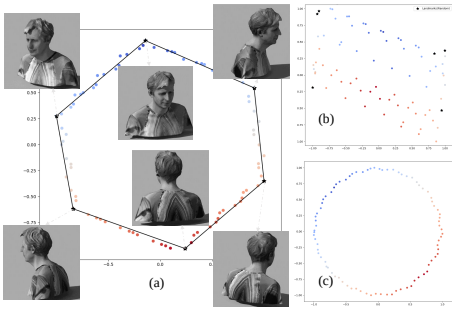

Figure 10: For *Portraits*: (a) homological L-Isomap embedding shown with sampled landmarks, (b) random L-Isomap embedding and (c) Isomap embedding.

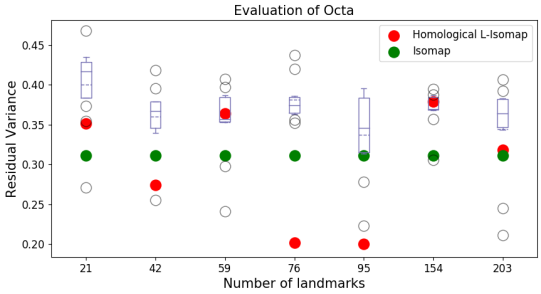

Figure 11: Quality assessment of embeddings using residual variance for *Octa*. Solid blue line in the box plot is the median, dotted blue line is the mean, the boundary of the box is the standard deviation, and black hollow circles are outliers. Solid red circles are RV measures for homological L-Isomap, while solid green circles are for Isomap.

landmarks and it outperforms Isomap with 42 landmarks. The optimal landmark size that achieves both computational efficiency and quality is at around 76 landmarks. When the number of landmarks goes beyond 150, homological L-Isomap does not seem to have an obvious advantage over other methods. In fact, at 203 landmarks, homological L-Isomap performs comparably with Isomap. This is not surprising, with a large number of landmarks, both L-Isomap and Isomap preserve the geometry of the data almost equally well.

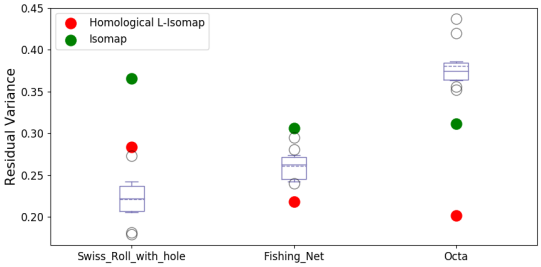

Figure 12: Quality assessment of embeddings using residual variance for *Swiss roll with a hole*, *Fishing Net* and *Octa*.

We also compare several datasets, the *swiss roll with a hole*, *Fishing Net* and *Octa*, using their respective optimal landmark sizes, in Fig. 12. Notice that homological L-Isomap outperforms the others for both *Fishing Net* and *Octa*, while it does not do well with *Swiss roll with a hole*. Intuitively, homological L-Isomap performs best when the data is complex, and has (possibly many) non-trivial homological features. In this case, both *Fishing Net* and *Octa* are a lot more complex and homologically interesting than the *Swiss roll with a hole*.

6.4 Dimensionality Reduction with Manifold Tearing

Manifold tearing results are shown in Fig. 13 for *Cylinder-5* and Fig. 14 for *Airfoil1* and *Bcsstk31* respectively. See Fig. 15 and Fig. 16 for quality assessment using persistence diagrams.

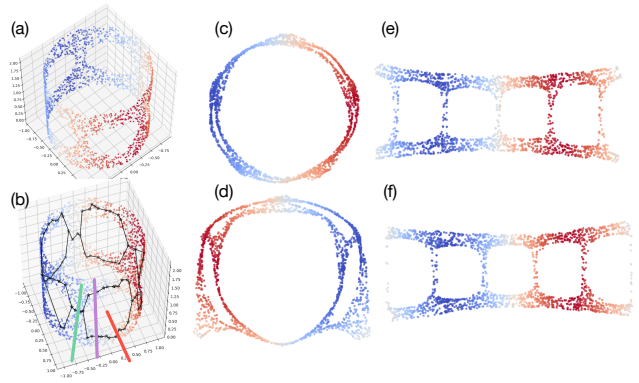

Figure 13: Results without and with manifold tearing for *Cylinder-5*. (a) Original point cloud. (b) Homological skeleton with 3 cutting options colored red, purple and green. (c) Isomap embedding without tearing. (d) Partial tearing with the red option. (e) Non-optimal tearing with purple option. (f) Optimal tearing with the green option.

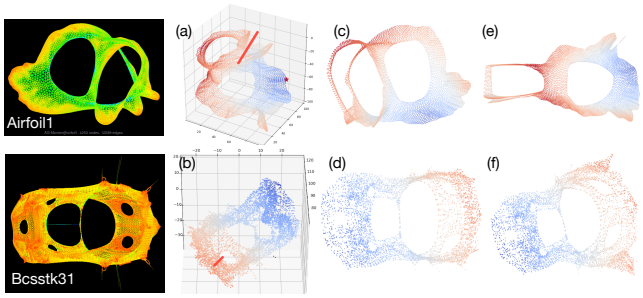

Figure 14: Results without and with manifold tearing for *Airfoil1* (top) and *Bcsstk31* (bottom). (a)-(b) original point clouds with marked cutting location. (c)-(d) Isomap embeddings without tearing. (e)-(f) Isomap embedding with tearing.

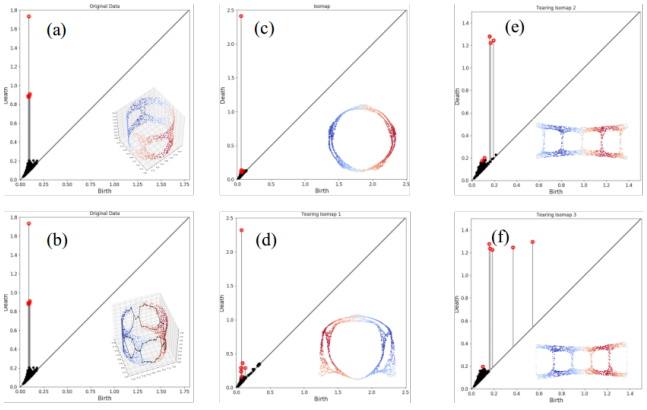

Figure 15: Homology-based quality assessment of DR for *Cylinder-5*.

Given a homological skeleton for *Cylinder-5*, we apply multiple cutting options to the skeleton and rank the resulting embeddings by the number of preserved homology classes. As shown in Fig. 13, without manifold tearing, the Isomap embedding destroys 5 out of 6 homological features, while optimal tearing preserves 5 out of 6 persistent homological features.

While Isomap preserves reasonably well the 3 persistent features for *Airfoil1*, manifold tearing further preserves 2 of the 3 homological features if we are willing to destroy one of them (Fig. 14 top, and Fig. 16).

For *Bcsstk31*, we focus on manifold tearing by cutting a short edge in the homological skeleton of *Bcsstk31* as shown in Fig.17, therefore "open up" the space further to reveal more geometric structures of the data.

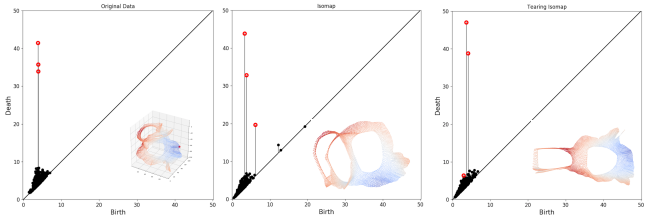

Figure 16: Homology based quality assessment of DR for *Airfoil1*.

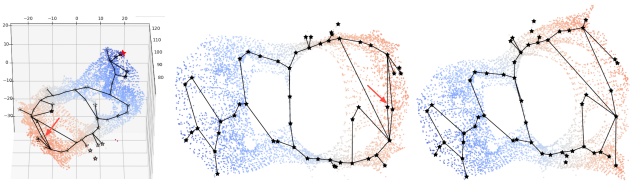

Figure 17: Using homological skeleton to aid manifold tearing for *Bcsstk31*. The location marked by the red arrow is where skeleton cutting takes place.

7 Discussion

We demonstrate in this paper that we can achieve 1-dimensional homology preservation while maintaining and possibly improving distance preservation using homology-based manifold landmarking and tearing. There are many research questions for future study. First, although we have provided a guideline for parameter selection and performed sensitivity analysis on many datasets, auto-tuning of parameters remains a challenge. Second, we need to find better and quantitative evaluation methods to locate the optimal locations for manifold tearing. Variations of *Wasserstein distance* [7] may be explored. Third, we have experimented with bootstrapping to calculate confidence intervals [16] to enhance our homology-based quality assessment, which aims to better separate topological signals from noise. Bootstrapping does well in estimating confidence intervals for datasets with simple homological structures, such as *Octa*, *Mice*, *Portraits*, and *Cylinder-3*; however it performs rather poorly for datasets containing a large number of homological features with varying sizes, such as *Fishing Net*, and *4elt*. Further study is needed in this direction. Finally, we are interested in exploring higher-dimensional homological skeletons [45] for DR to preserve homological features beyond 1-dimensions.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[2] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. Reeb graphs for shape analysis and applications. *Theoretical Computer Science*, 392:5–22, 2008.

[3] G. Carlsson, T. Ishkhanov, V. De Silva, and A. Zomorodian. On the local behavior of spaces of natural images. *International journal of computer vision*, 76(1):1–12, 2008.

[4] M. Carriére, B. Michel, and S. Oudot. Statistical analysis and parameter selection for mapper. *ArXiv: 1706.00204*, 2017.

[5] F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. Persistence-based clustering in Riemannian manifolds. *Journal of the ACM*, 60(6), 2013.

[6] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete and Computational Geometry*, 37(1):103–120, 2007.

[7] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. Lipschitz functions have l_p-stable persistence. *Foundations of Computational Mathematics*, 10(2):127–139, 2010.

[8] J. P. Cunningham and Z. Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, pages 2859–2900, 2015.

[9] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems*, pages 705–712, 2003.

[10] V. de Silva and J. B. Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Stanford University, 2004.

[11] R. Diekmann and R. Preis. Ag-monien graph collection.

[12] I. Duff, R. Grimes, and J. Lewis. Sparse matrix problems. *ACM Transactions on Mathematical Software*, 14(1):1–14, 1989.

[13] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, Providence, RI, USA, 2010.

[14] C. M. Elliott and Z. Songmu. On the Cahn-Hilliard equation. *Archive for Rational Mechanics and Analysis*, 96(4):339–357, 1986.

[15] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[16] B. T. Fasy, F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, and A. Singh. Confidence sets for persistence diagrams. *The Annals of Statistics*, 42(6):2301–2339, 2014.

[17] I. Fodor. A survey of dimension reduction techniques. Technical Report UCRL-ID-148494, Center for Applied Scientific Computing, Lawrence Livermore National, 2002.

[18] E. R. Gansner, Y. Koren, and S. C. North. Topological fisheye views for visualizing large graphs. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):457–468, 2005.

[19] X. Ge, I. Safa, M. Belkin, and Y. Wang. Data skeletonization via Reeb graphs. *Advances in Neural Information Processing Systems*, 2011.

[20] R. Ghrist. Barcodes: The persistent topology of data. *Bullentin of the American Mathematical Society*, 45:61–75, 2008.

[21] R. Ghrist. Three examples of applied & computational homology. *Nieuw Archief voor Wiskunde (The Amsterdam Archive, Special issue on the occasion of the fifth European Congress of Mathematics)*, pages 122–125, 2008.

[22] A. Gracia, S. González, V. Robles, and E. Menasalvas. A methodology to compare dimensionality reduction algorithms in terms of loss of quality. *Information Sciences*, 270(20):1–27, 2014.

[23] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[24] A. B. Lee, K. S. Pedersen, and D. Mumford. The non-linear statistics of high-contrast patches in natural images. *International Journal of Computer Vision*, 54:83–103, 2003.

[25] J. A. Lee and M. Verleysen. Nonlinear dimensionality reduction of data manifolds with essential loops. *Neurocomputing*, 67:29–53, 2005.

[26] J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer, 2007.

[27] J. A. Lee and M. Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7):1431–1443, 2009.

[28] D. Liang and J. Paisley. Landmarking manifolds with gaussian processes. *Proceedings of Machine Learning Research*, 37:466–474, 2015.

[29] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1249–1268, 2017.

[30] D. Meng, Y. Leung, and Z. Xu. Detecting intrinsic loops underlying data manifold. *IEEE Transactions on Knowledge and Data Engineering*, 25(2):337–347, 2013.

[31] J. C. Nascimento and G. Carneiro. Deep learning on sparse manifolds for faster object segmentation. *IEEE Transactions on Image Processing*, 26(10):4978–4990, 2017.

[32] M. Natali, S. Biasotti, G. Patané, and B. Falcidieno. Graph-based representations of point clouds. *Graphical Models*, 73(5):151–164, 2011.

[33] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901.

[34] J. A. Perea and J. Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3):799–838, 2015.

[35] B. Rieck and H. Leitte. Agreement analysis of quality measures for dimensionality reduction. *Topological Methods in Data Analysis and Visualization IV*, pages 103–117, 2015.

[36] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[37] N. Saul and H. J. van Veen. MIwave/kepler-mapper: 186f (version 1.0.1). zenodo., November 2017.

[38] R. Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140, 1962.

[39] H. Shi, B. Yin, Y. Kang, C. Shao, and J. Gui. Robust L-Isomap with a novel landmark selection method. *Mathematical Problems in Engineering*, 2017.

[40] Y. Shinagawa, T. Kunii, and Y. Kergosien. Surface coding based on morse theory. *IEEE Computer Graphics and Applications*, 11(5):66–78, 1991.

[41] G. Singh, F. Mémoli, and G. Carlsson. Topological methods for the analysis of high dimensional data sets and 3D object recognition. *Eurographics Symposium on Point-Based Graphics*, 22, 2007.

[42] B. L. Smarr, I. Zucker, and L. J. Kriegsfeld. Detection of successful and unsuccessful pregnancies in mice within hours of pairing through frequency analysis of high temporal resolution core body temperature data. *PLoS One*, 2016.

[43] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[44] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[45] S. K. Verovšsek, V. Kurlin, and D. Lešnik. A higher-dimensional homologically persistent skeleton. *ArXiv: 1701.08395*, 2017.

[46] C. Walshaw. A multilevel algorithm for force-directed graph drawing. *International Symposium on Graph Drawing*, pages 171–182, 2000.

[47] K. Q. Weinberger, B. D. Packer, and L. K. Saul. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. *International Workshop on Artificial Intelligence and Statistics*, 2005.

[48] E. Wong, S. Palande, B. Wang, B. Zielinski, J. Anderson, and P. T. Fletcher. Kernel partial least squares regression for relating functional brain network topology to clinical measures of behavior. *IEEE International Symposium on Biomedical Imaging (ISBI)*, 2015.

[49] S. Xia. A topological analysis of high-contrast patches in natural images. *Journal of Nonlinear Sciences and Applications*, 9:126–138, 2016.

[50] S. Yan, D. Xu, B. Zhang, H. jiang Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007.

[51] P. Zhang, Y. Ren, and BoZhang. A new embedding quality assessment method for manifold learning. *Neurocomputing*, 97(15):251–266, 2012.

[52] H. Zhou, P. Wang, and H. Li. Research on adaptive parameters determination in dbscan algorithm. *Journal of Information & Computational Science*, 9(7):1967–1973, 2012.