Topological Point Cloud Clustering

Vincent P. Grande [1] **Michael T. Schaub** [1]

Abstract

We present Topological Point Cloud Clustering (TPCC), a new method to cluster points in an arbitrary point cloud based on their contribution to global topological features. TPCC synthesizes desirable features from spectral clustering and topological data analysis and is based on considering the spectral properties of a simplicial complex associated to the considered point cloud. As it is based on considering sparse eigenvector computations, TPCC is similarly easy to interpret and implement as spectral clustering. However, by focusing not just on a single matrix associated to a graph created from the point cloud data, but on a whole set of Hodge-Laplacians associated to an appropriately constructed simplicial complex, we can leverage a far richer set of topological features to characterize the data points within the point cloud and benefit from the relative robustness of topological techniques against noise. We test the performance of TPCC on both synthetic and real-world data and compare it with classical spectral clustering.

1. Introduction

A central quest of unsupervised machine learning and pattern recognition is to find meaningful (low-dimensional) structures within a dataset, where there was only apparent chaos before. In many cases, a data set consist of a point cloud in a high-dimensional space, in which each data point represents a real-world object or relation. Dimensionality reduction and clustering methods are thus often used as a first step towards extracting a more comprehensible description of the data at hand, and can yield meaningful insights into previously hidden connections between the objects.

The paradigm of most classical clustering algorithms assumes that there are only a few "fundamental types" within the observed data and every data point can be assigned to one of those types. How the notion of type is interpreted varies in different approaches, but in most cases, the data is assumed to be a disjoint union of these types plus noise, and the focus is on identifying an optimal *local* assignment of the points to the respective types (clusters). For instance, many prototypical clustering algorithms like k-means clustering (Steinhaus, 1957) or mixture models like Gaussian mixtures (Day, 1969) aim to group points together that are close according to some local distance measure in $\mathbb{R}^n$. Other variants, like DBSCAN, aim to track dense subsets within the point cloud (Ester et al., 1996), and subspace clustering aims to find a collection of low-dimensional linear subspaces according to which the points can be grouped (Chen & Lerman, 2009). On the other hand, quantifying and utilizing the overall shape of the point cloud, i.e., how it is *globally* assembled from the different clusters or how to find the best possible cluster types to build up the data is typically not a concern.

In comparison, topological data analysis (TDA), which has gained significant interest in the last decades (Carlsson & Vejdemo-Johansson, 2021) emphasises an opposite perspective. Here the dataset is effectively interpreted as one complex object, a topological space, whose "shape" we try to determine by measuring certain topological features (typically homology) to understand the *global* make-up of the entire point cloud. Such topological features are virtually omnipresent and are very flexible to describe highly complex shapes. For instance, in medicine, they can measure the topology of vascular networks and can distinguish between tumorous and healthy cells (Stolz et al., 2022). In public health studies, they have been used to analyse health care delivery network efficiency (Gebhart et al., 2021), and in Biochemistry, persistent homology has been used to analyse protein binding behaviour (Kovacev-Nikolic et al., 2016). In Data Science, the Mapper algorithm uses topological features of data sets to produce a low dimensional representation of high dimensional data sets (Singh et al., 2007).

One key insight that has driven the success of the ideas of TDA is that insightful higher-order information is often encoded in the topological features of (some amenable representation of) the data. However, in contrast to classical

[1] Department of Computer Science, RWTH Aachen University, Aachen, Germany. Correspondence to: Vincent P. Grande <grande@cs.rwth-aachen.de>, Michael T. Schaub <schaub@cs.rwth-aachen.de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

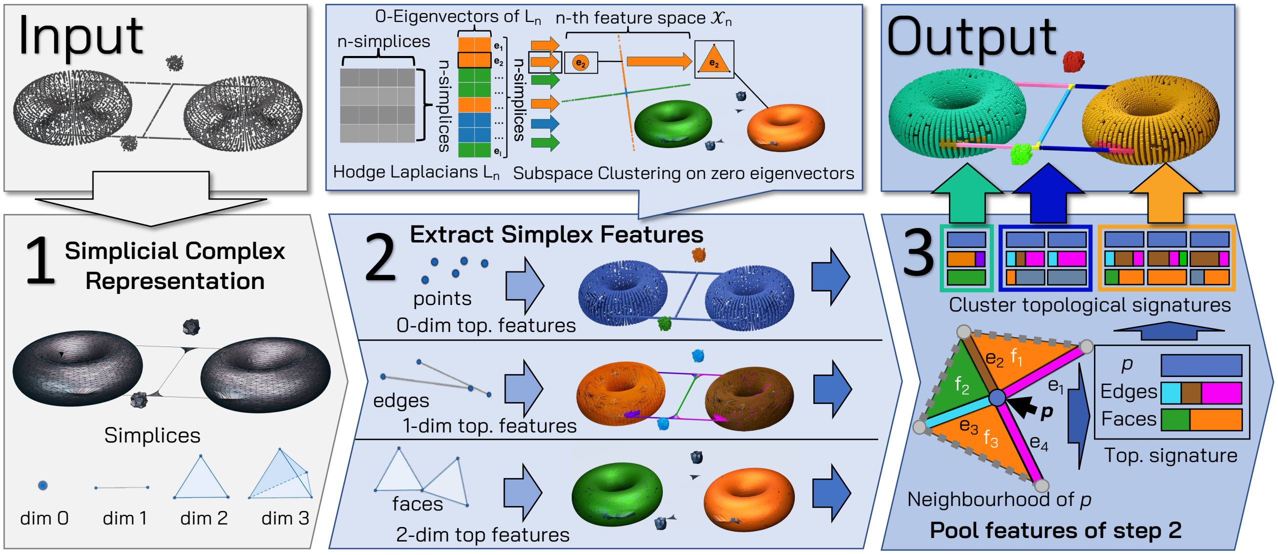

Figure 1. **Schematic of topological point cloud clustering (TPCC). Step 1.** To capture the topological shape of the point cloud a *simplicial complex* is constructed. **Step 2.** Associated *Hodge-Laplace* operators are constructed separately for each dimension. The method extracts information from the sparse Hodge-Laplace operators by computing their 0-eigenvectors. The 0-eigenvectors are indexed by the simplices in the respective dimensions. We use these eigenvectors to embed the simplices into a single featurespace $\mathcal{F}_n$ for each dimension of the simplices and perform subspace clustering on these feature spaces. **Step 3.** For each simplex, we relay the clustering information back to its vertices. Every point is thus equipped with a *topological signature*, aggregating information on topological features over all dimensions. Finally, the original points are clustered using a standard clustering approach on their topological signature.

clustering, the question of how the individual data points contribute to the make-up of the overall topological object is typically not a result of these types of analysis. This can render the interpretation of the results difficult, as often the individual data points have a concrete and meaningful (often physical) interpretation and we would thus like to know how these points relate to the overall measured topological object.

The aim of this paper is to combine the advantages of these two perspectives and to establish a synthesis of traditional clustering algorithms with their easily interpretable output and the powerful notion of topological features of TDA. Topological Point Cloud Clustering (TPCC) bridges this gap between the local nature of classical clustering and the global features of TDA, by aggregating information gained from multiple levels of a form of generalized spectral clustering on the k-simplices.

Contributions We develop a novel topological point cloud clustering method that clusters the points according to what topological features of the point cloud they contribute to. We prove that the clustering algorithm works on a class of synthetic point clouds with an arbitrary number of topological features across arbitrary dimensions. Finally, we verify the accuracy of topological point cloud clustering on a number of synthetic and real-world data and compare it with other approaches on data sets from the literature.

Organisation of the paper We introduce necessary topological notions in Section 2. In Section 3, we describe the main idea of topological point cloud clustering. A theoretical result on the accuracy of the algorithm on a class of synthetic point clouds is then presented in 4. Finally, we show the distinguishing power of topological point cloud clustering on synthetic data, protein data and physically inspired real-world data in Section 5. In particular, we compare the results of our algorithms with other clustering methods and study the robustness of TPCC against noise on synthetic data. Certain technical aspects of our algorithm and our running example are explained in more detail in Appendix A and Appendix B.

Related Work Our work builds upon several ideas that have been promoted in the literature. In particular, TPCC may be seen as a generalization of spectral clustering (von Luxburg, 2007). Spectral clustering starts with the construction of a graph from an observed point cloud, by identifying each data point with a vertex and connecting close-by points with an edge. Vertices are then clustered according to their spectral embedding, i.e., the dominant eigenvectors of the graph representation considered (typically in terms of an associated Laplacian matrix). However, these eigenvectors used by spectral clustering are merely related to connectivity properties (0-homology), and the produced clustering is thus restricted in terms of the topological features it considered. Topological Mode Analysis (Chazal et al., 2013) clusters point clouds using persistent homology. However, because only 0-dimensional homology is considered the approach cannot cluster according to higher-order topological features like holes, loops and voids.

Our work does not just build a graph from the point cloud data but employs a simplicial complex to describe the observed point cloud (similar to how it is done in persistent homology) and embeds and clusters all k-simplices into the 0-eigenvector space of the k-th Hodge Laplacian. Related ideas of using embeddings based on the Hodge-Laplacian can be found in (Schaub et al., 2020; Chen & Meila, 2021; Ebli & Spreemann, 2019): The idea of defining a harmonic embedding to extract meaningful information about a simplicial complex has been discussed in the context of trajectory classification (Schaub et al., 2020; Frantzen et al., 2021). In (Chen & Meila, 2021), the authors study how this embedding is affected by constructing more complex manifolds from simpler building blocks. However, they do not study how to decompose the underlying points based on this embedding. In (Ebli & Spreemann, 2019), the authors develop a notion of harmonic clustering on the simplices of a simplicial complex. We use an extended version of this clustering as one step in TPCC. (Krishnagopal & Bianconi, 2021) have as well considered harmonic clustering of simplices but only use it to detect large numbers of communities in small simplicial complexes. In (Perea, 2020), the author uses a smoothed version of cohomology generators to quantify homology flows and build circular coordinates. From a certain point of view, this is surprisingly similar to considering zero eigenvectors of Hodge Laplace operators. Some related ideas to our work are also investigated in (Stolz et al., 2020), where the authors provide a tool for detecting anomalous points of intersecting manifolds. As we will see, our algorithm is able to detect not only these points but can provide additional information about all remaining points as well. There has been some work on surface and manifold detection in point clouds (Martin & Watson, 2011; Hoppe et al., 1992). In contrast to TPCC, these algorithms don't provide any clustering or additional information on the points and are confined to manifold-like data, which is usually assumed to be a 2-dimensional surface in 3-dimensional space. Approaches utilising tangent bundle constructions assume that the data corresponds to intersecting manifolds and that the desired clusters are represented by individual manifolds (Wang et al., 2011; Gong et al., 2012; Tinarrage, 2023). However, this may not be the case in real-world applications. TPCC does not make such a restrictive assumption and is thus more widely applicable

The Hodge-Laplacian has also featured in a number of works from graph signal processing and geometric deep learning. A homology-aware simplicial neural network is constructed in (Keros et al., 2022), extending previous models (Roddenberry et al., 2021; Bunch et al., 2020) on simplices of dimension two (Ebli et al., 2020; Bodnar et al., 2021)). However, these approaches focus on a scenario where the higher-order simplices have some real-world meaning, e.g., 1-simplices can be identified by streets, neural links, or pairs of co-authors. In contrast here our primary focus is on a scenario in which we are only given a point cloud to start with and thus only the points have a real-world meaning, whereas the higher dimensional features are added via some artificial simplicial complex simply to extract further information about the shape of the data. This is the case in most standard application scenarios.

2. A Topological Notion of Features

A main goal of topology is to capture the essence of spaces. Topological tools try to describe globally meaningful features of spaces that are indifferent to local perturbations and deformations. This indifference of topological features to local perturbations can be a crucial asset when analysing large-scale datasets, which are often high-dimensional and noisy. To leverage these ideas, we need to explain what we mean by *topological features* throughout the paper. A key assumption in this context is that high dimensional data sets may be seen as samplings from topological spaces — most of the time, even low-dimensional manifolds (Fefferman et al., 2016). Rather than providing a complete technical account, in the following, we try to emphasize the relevant conceptual ideas and refer the interested reader to (tom Dieck, 2008; Bredon et al., 1993; Hatcher, 2002) for further details.

Simplicial Complexes The prototypical topological space is a subset of $\mathbb{R}^n$ and hence continuous. Storing the infinite number of points in such a space individually is impossible. On the other hand, our observed point cloud will always be discrete and non-connected. *Simplicial complexes* (SC) bridge this gap between the continuous spaces of topology, and the discrete nature of our point cloud. They offer a way to build topological spaces from easy-to-define building blocks. Indeed, a well-known theorem in topology (Quillen, 1967) asserts that any topological space with the homotopy type of a CW complex can be approximated by a simplicial complex.

Definition 2.1 (Abstract simplicial complex)**.** An abstract simplicial complex $\mathcal{S}$ consists of a set of vertices X and a set of finite non-empty subsets of X, called simplices S, such that **(i)** S is closed under taking non-empty subsets and **(ii)** the union over all simplices $\bigcup_{\sigma \in S} \sigma$ is X. For simplicity, we often identify $\mathcal{S}$ with its set of simplices and use $\mathcal{S}_n$ to denote the subset of simplicies with $n+1$ elements.

Intuitively, in order to build a simplicial complex $\mathcal{S}$, we first start with a set of vertices V. These are called the 0-simplices. We can then add building blocks of increasing dimension. The 1-simplices represent edges between 2 vertices, the 2-simplices are triangles between 3 vertices that are already connected by edges. An n-simplex resembles an n-dimensional polyhedra. An n-simplex σ_n connects $(n+1)$ vertices, given that they are already connected by all possible $(n-1)$-simplices. These $(n-1)$-simplices are then called the faces of σ_n. We call two $(n-1)$-simplices *upper-adjacent* if they are faces of the same n-simplex. Correspondingly, we call two n-simplices *lower-adjacent* if they share a common $(n-1)$-simplex as a face.

Vietoris-Rips complex Building the Vietoris-Rips complex is a method of turning a point cloud into a simplicial complex, approximating the topological features of the space it was sampled from. The Vietoris-Rips complex takes 2 arguments as input: The point cloud X and a minimal distance ε. It then builds a simplicial complex $\mathcal{S}$ by taking X as the set of vertices (and thus of 0-simplices) of $\mathcal{S}$. Between every two distinct vertices of distance $d < \epsilon$ it adds an edge, i.e. an 1-simplex. Inductively, it then adds an n-simplex for each set of $(n+1)$ vertices in X with pair-wise distance smaller than ε. In practice, one often restricts this process to simplices of dimension $n \leq N$ for some finite number N.

Boundary matrices and the Hodge-Laplacians All topological information of a simplicial complex $\mathcal{S}$ can be encoded in its *boundary matrices* $\mathcal{B}_n$. The rows of $\mathcal{B}_n$ are indexed by the n-simplices of $\mathcal{S}$, the columns are indexed by the $(n+1)$-simplices.

Definition 2.2. Let $\mathcal{S} = (S, X)$ be a simplicial complex and $\preceq$ a total order on its set of vertices X. For $n \geq i, n \geq 1$ we define the i-th face map $f_i^n \colon \mathcal{S}_n \to \mathcal{S}_{n-1}$ by

$$f_i^n \colon \{x_0, x_1, \ldots, x_n\} \mapsto \{x_0, x_1, \ldots, \widehat{x}_i, \ldots, x_n\}$$

where we have that $x_0 \preceq x_1 \preceq \cdots \preceq x_n$ and $\widehat{x}_i$ denotes the omission of x_i. Then we define the n-th boundary operator $\mathcal{B}_n \colon \mathbb{R}[\mathcal{S}_{n+1}] \to \mathbb{R}[\mathcal{S}_n]$ by

$$\mathcal{B}_n \colon \sigma \mapsto \sum_{i=0}^{n+1} (-1)^i f_i^{n+1}(\sigma).$$

We identify $\mathcal{B}_n$ with its matrix representation in lexicographic ordering of the simplex basis.

Note that with this definition, $\mathcal{B}_0$ is simply the familiar vertex-edge-incidence matrix of the associated graph built from the 0- and 1-simplices of $\mathcal{S}$.

Definition 2.3. The n-th *Hodge-Laplacian* L_n of $\mathcal{S}$ is a square matrix indexed by the n-simplices of $\mathcal{S}$:

$$L_n := \mathcal{B}_{n-1}^\top \mathcal{B}_{n-1} + \mathcal{B}_n \mathcal{B}_n^\top \tag{1}$$

where we take $\mathcal{B}_{-1}$ to be the empty matrix.

The key insight about the $\mathcal{B}_n$ is the following lemma:

Lemma 2.4. *For a simplicial complex $\mathcal{S}$ with boundary matrices $\mathcal{B}_i$ we have that $\mathcal{B}_n \circ \mathcal{B}_{n+1} = 0$ for $n \geq 0$.*

Algorithm 1 Topological Point Cloud Clustering (TPCC)

```
Input: Point cloud $X$, maximum dimension $d$
Pick $\varepsilon$ and construct VR complex $\mathcal{S}$ of $X$
Construct Hodge Laplacians $L_0, \ldots, L_d$ of $\mathcal{S}$
for $i = 0$ to $d$ do
    Compute basis $v_0^i, \ldots, v_{B_i}^i$ of 0-eigenvectors of $L_i$
    Subspace Clustering on rows of $[v_0^i, \ldots, v_{B_i}^i]$
    Assign clusters to corresponding $i$-simplices of $\mathcal{S}$
    for $x \in X$ do
        (Top. signature of $x$:) Collect cluster information of
        $i$-simplices $\sigma_i$ with $x \in \sigma_i$
    end for
end for
Cluster $X$ according to topological signatures
Output: Labels of $x \in X$
```

Topological features: Homology and Betti numbers One of the main topological concepts is *homology*. The k-th *homology module* $H_k(X)$ of a space X encodes the presence and behaviour of k-dimensional loops, enclosing generalised $(k+1)$-dimensional voids/cavities. The k-th *Betti number* $B_k(X)$ of X denotes the rank $\operatorname{rk} H_k(X)$ of the corresponding homology module. The 0-th Betti number $B_0(X)$ is the number of connected components of X, $B_1(X)$ counts the number of loops and $B_2(X)$ counts how many 3-dimensional cavities with 2-dimensional borders are enclosed in X, and so on.

The following connection between the homology of an SC and its Hodge Laplacian will prove essential to us:

Lemma 2.5 ((Eckmann, 1944/45; Friedman, 1998))**.** *For a simplicial complex $\mathcal{S}$, let L_n be the Hodge Laplacians and B_n be the Betti numbers of $\mathcal{S}$. Then we have that* $\operatorname{rk}\ker L_n = B_n$.

The dimension of the kernel of the Hodge-Laplacian is equal to the number of orthogonal zero eigenvectors of L_n over $\mathbb{R}$. Hence the Hodge-Laplacian provides a gateway for accessing topological features by computing eigenvectors.

3. TPCC: Algorithm and Main Ideas

In this section, we will describe Topological Point Cloud Clustering and its main ideas. A pseudocode version can be found in Algorithm 1.

Running example To illustrate our approach, we use the example displayed in Figure 1 consisting of two 4-dimensional tori, depicted here in their projection to 3d space. We connected the tori with two lines, which are again connected by a line. Additionally, the point cloud includes two separate connected components without higher dimensional topological features. Our point cloud has thus

11 topological features across 3 dimensions. In terms of Betti numbers, we have $B_0 = 3$, $B_1 = 6$, and $B_2 = 2$. For an in-depth discussion of the topology and construction of the running example, see Appendix B.

Step 1: Approximating the space To characterize our point cloud in terms of topological information, we suggest using the framework of simplicial complexes and the Vietoris-Rips Complex due to their straightforward definitions. The goal of this paper is to show that even with this naive approach of constructing a simplicial complex, a topologically meaningful clustering can be achieved. However, we note that TPCC is agnostic towards the method the simplicial complex was constructed. In low dimensions, the α-complex provides a computationally efficient alternative with a lower number of simplices. Complexes built using DTM-based filtrations are another alternative more robust to outliers (Anai et al., 2020).

The general assumption is that the points of the point cloud are in some general sense sampled, potentially with some additional noise, from a geometrical space. Now we would like to retrieve the topology of this original geometrical space from the information provided via the sampled points. Hence, following common ideas within TDA, we construct a computationally accessible topological space in terms of a simplicial complex on top of the point cloud approximating the ground truth space. We denote the simplicial complex associated to our toy point cloud by $\mathcal{S}$. We note that the TPCC framework works both with simplicial as well as with cellular complexes. For simplicity however, we chose to stick with simplicial complexes throughout this paper.

Step 2A: Extracting topological features Having built the simplicial complex $\mathcal{S}$, we need to extract its topological features. However, standard measures from topological data analysis only provide global topological features: For instance, Betti numbers are global features of a space, and persistence landscapes measure all features at once (Bubenik et al., 2015). In contrast, we are interested in how individual simplices and points are related to the topological features of the space. It is possible to extract a homology generator for a homology class in persistent homology (Obayashi, 2018). This approach is however not suitable for us, because the choice of a generator is arbitrary, and only the contribution of a small number of simplices can be considered.

TPCC utilises a connection between the simplicial Hodge-Laplace operators and the topology of the underlying SC. The dimension of the 0-space of the k-th Hodge-Laplacian L_k is equal to the k-th Betti number B_k (Eckmann, 1944/45; Friedman, 1998). Furthermore, the rows and columns of the Hodge-Laplacian L_k are indexed by the k-simplices of $\mathcal{S}$ and describe how simplices relate to each other, and in particular how they contribute to homology in terms of the null space of the L_k.

Let us now consider a concrete loop/boundary $\mathcal{F}$ of an $(k+1)$-dimensional void. We can then pick a collection S of edges/k-simplices that represents this loop/boundary. By assigning each simplex in S the entry ± 1 based on the orientation of the simplex, and every other simplex the entry 0, we obtain a corresponding vector e_S. The Hodge Laplace operator $L_k = \mathcal{B}_{k-1}^\top \mathcal{B}_{k-1} + \mathcal{B}_k \mathcal{B}_k^\top$ consists of two parts. The kernel of the down-part, $\mathcal{B}_{k-1}^\top \mathcal{B}_{k-1}$, is spanned by representations of the boundaries of $(k+1)$-dimensional voids. Hence, e_S lies in this kernel: $\mathcal{B}_{k-1}^\top \mathcal{B}_{k-1} e_S = 0$. The kernel of the up-part of the Hodge Laplacian, $\mathcal{B}_k \mathcal{B}_k^\top$, is spanned by vectors that represent smooth flows along the k-simplices. Thus by smoothing along the k-simplices one can turn e_S into an eigenvector $\widehat{e}_S$ of the entire Hodge Laplace operator L_k:

$$L_k \widehat{e}_S = \mathcal{B}_{k-1}^\top \mathcal{B}_{k-1} \widehat{e}_S + \mathcal{B}_k \mathcal{B}_k^\top \widehat{e}_S = 0. \tag{2}$$

We call $\widehat{e}_{\mathcal{F}} := \widehat{e}_S$ the *characteristic eigenvector* associated to the loop/void $\mathcal{F}$.

For simplicity, let us first consider the case where the k-th Betti number $B_k(\mathcal{S})$ is 1. Then the zero-eigenvector v_0 of L_k has one entry for every k-simplex and is the characteristic eigenvector $\widehat{e}_{\mathcal{F}}$ for the single topological feature $\mathcal{F}$ in dimension k. The entries of v_0 measure the contribution of the corresponding simplices to $\mathcal{F}$. Intuitively, we can visualise the homology "flowing" through the simplices of the simplicial complex. The entries of the eigenvector correspond to the intensity of the flow in the different k-simplices. Because of the way we constructed $\widehat{e}_{\mathcal{F}}$, the homology flow is then concentrated along the k-dimensional boundary of a hole/void in the space. In the 1-dimensional setting, this corresponds to harmonic flows along edges around the holes of an SC (Schaub et al., 2021). The case for the Betti number larger one $B_k > 1$ will be discussed in more detail in the following paragraph.

Step 2B: Clustering the n-simplices Extending ideas from (Ebli & Spreemann, 2019; Schaub et al., 2020) we use the obtained coordinates for each simplex to cluster the simplices. In the case where L_k has a single 0-eigenvalue, we can easily cluster the simplices by simply looking at the entries of the 0-eigenvector e: We can ignore the sign of the entry e_σ of e corresponding to a simplex σ because this only reflects whether the arbitrarily chosen orientation of σ aligns with the direction of the "homology flow". Then, we assign all simplices σ with absolute value of e_σ above a certain threshold $|e_\sigma| > \varepsilon$ to the cluster of homologically significant simplices. The remaining simplices are assigned to a separate cluster.

In the case of multiple boundaries of voids of the same dimension, i.e. $B_k > 1$, each boundary $\mathcal{F}$ again corresponds

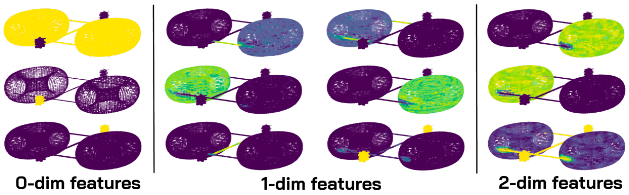

Figure 2. Above we depict the heatmaps for all 16 distinct combinations of topological features encoded in the topological signature across 3 dimensions of our toy example. Note that some of the features are redundant, as both edges and faces can measure membership of a torus.

to a "homology flow" with an associated characteristic eigenvector $\widehat{e}_{\mathcal{F}_i}$ of L_k. The $\widehat{e}_{\mathcal{F}_i}$ span the zero-eigenspace E_k of L_k. However, an eigenvector solver will yield an arbitrary orthonormal basis $e_1, \ldots, e_{B_k}$ of E_k which is only unique up to unitary transformations. For a k-simplex $\sigma \in \mathcal{S}_k$, let $e_i(\sigma)$ denote the coordinate associated to σ of the i-th basis vector e_i of E_k obtained by the eigenvector solver. Now we denote by $\iota \colon \mathcal{S}_k \to \mathbb{R}^{B_k}$,

$$\iota \colon \sigma \mapsto (e_1(\sigma), e_2(\sigma), \ldots, e_{B_k}(\sigma)) \in \mathbb{R}^{B_k}$$

the embedding of the simplices into the k-th *feature space* $\mathcal{X}_k := \mathbb{R}^{B_k}$. Note that because we could have started with any orthonormal basis of E_k the feature space is only defined up to arbitrary unitary transformations. The points of the feature space $\mathcal{X}_k$ represent different linear combinations of the basis vectors of the zero eigenspace of L_k. They also represent linear combinations of the $\widehat{e}_{\mathcal{F}_i}$, and hence intuitively of the topological features.

In the most simple case, the $\widehat{e}_{\mathcal{F}_i}$ are orthogonal to each other and thus have disjoint support. Then they represent orthogonal linear combinations of the original basis of E_k in the feature space $\mathcal{X}_k$. Hence the "natural" $\widehat{e}_{\mathcal{F}_i}$-basis can be recovered by subspace clustering the k-simplices on the feature space $\mathcal{X}_k$ as depicted in the top of Figure 1. For computational reasons, we subsample the simplices used for the subspace clustering. The remaining simplicies will then be classified using a k-nearest neighbour classifier on the feature space $\mathcal{X}_k$. See Section 3 and Appendix C for a discussion of more complicated special cases.

Step 3A: Aggregating the information to the point level Finally, we can try to relate the information collected so far back to the points. For every point x and every dimension d, we aggregate the cluster ids of the d-simplices which contain x. We call the collected information the *topological signature* of p.

Definition 3.1 (Topological Signature)**.** Let X be a point cloud with associated simplicial complex $\mathcal{S}$. For a simplex $\sigma \in \mathcal{S}$, we denote its cluster assignments from the previous step of TPCC by $C(\sigma)$. Then, the *topological signature* $\tau(x)$ of a point $x \in X$ is the multi-set

$$\tau(x) := \{\{C(\sigma) : \sigma \in \mathcal{S}, x \in \sigma\}\}.$$

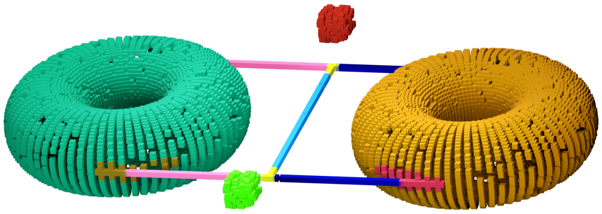

Figure 3. The final clustering obtained with TPCC. There are 10 clusters in total. Two clusters identify the two tori (turquoise and ochre), two disconnected cubes (red and lime), dark blue and salmon for the connecting lines of the tori to the middle, azure for the middle line, yellow for the intersection of the lines, and fuchsia and brown for the gluing points of the points to the tori. Note that there are virtually no outliers.

After normalising for each i by the number of i-simplices containing the point, topologically similar points will have a similar topological signature. Figure 1, Step 3 illustrates how the topological signature is calculated. In Figure 2 we show how the different features of the topological signature highlight topologically different areas of the point cloud. Interestingly, we can even retrieve information on the gluing points between two topologically different parts. In Figure 3, the "gluing points" between the tori and the lines receive their own cluster. This is because roughly half of the simplices adjacent to the gluing points receive their topological clustering information from the torus and the other half from the adjacent lines. Hence the gluing points are characterised by a mixture of different topological signatures.

Step 3B: Computing the final clustering If we apply k-means or spectral clustering to a normalised form of the topological signatures of the points of our toy example, we arrive at the clustering of Figure 3.

In comparison to standard clustering methods, TPCC can assign the same cluster to similar sets of points consisting of multiple connected components if they share the same topological features. In Figure 3, the two dark blue lines are assigned to the same cluster, because they both lie on the same loop and have no additional topological feature. This highlights the ability of TPCC to take higher-dimensional information into consideration,exceeding the results obtainable by proximity-based information.

Choice of parameters TPCC needs two main parameters, ε and d. For the choice of the maximum homology de-

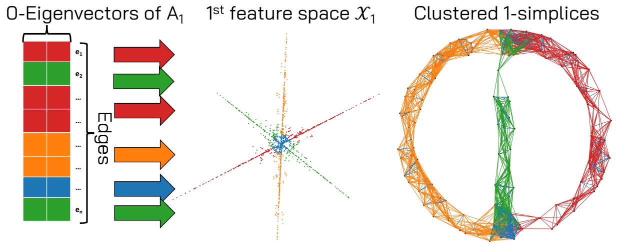

Figure 4. The circle is divided into two parts by a vertical line. This gives the corresponding SC two generating loops in dimension 1, corresponding to a 2-dimensional 0-eigenspace of the Hodge-Laplacian L_1 and a 2-dimensional 1st feature space $\mathcal{X}_1$. However, now there are three linear subspaces corresponding to linear combinations of the two generating loops. TPCC is able to detect three different clusters of topologically significant edges.

gree d to be considered there are three heuristics listed in decreasing importance:

I. When working with real-world data, we usually know which kind of topological features we are interested in, which will then determine d. E.g., if we are interested in the loops of protein chains, we only need 1-dimensional homology and thus choose $d = 1$. When interested in voids and cavities in 3d tissue data, we need 2-dimensional homology and thus choose $d = 2$, and so on.

II. There are no closed n-dimensional submanifolds of $\mathbb{R}^n$. This means that if the point cloud lives in an ambient space of low dimension n, the maximum homological features of interest will live in dimension $n - 1$ and hence we can choose $d = n - 1$.

III. In practice, data sets rarely have non-vanishing highly persistent homology in degree above 2 and considering the dimensions 0–2 usually suffices. Otherwise, one can calculate persistent homology up to the maximum computationally feasible degree to identify dimensions with sufficiently persistent homology classes, and then take d as the maximum of these dimensions.

Picking the correct value of ε means choosing the correct scale. For the experiments in Figure 7, we have implemented a heuristic which computes the persistence diagram of the point cloud, and then picks the ε maximizing the number of topological features with high persistence and minimizing the number of features with low persistence for this value. As can be seen, this method performs comparatively well for considerable noise.

Technical considerations: Linear combinations of features In practice, topological features of the same dimension are not always separated in space. A bubble of soap may consist of two individual compartments divided by a thin layer of soap. This middle layer then contributes to the boundaries of the two voids, i.e. to two topological features of dimension 2. How is this reflected in the $\widehat{e}_{\mathcal{F}_i}$?

This time, the characteristic eigenvectors $\widehat{e}_{\mathcal{F}_i}$ corresponding to boundaries $\mathcal{F}_i$ of voids of the same dimension are not orthogonal anymore. The supports of the $\widehat{e}_{\mathcal{F}_i}$ overlap in the same simplices the corresponding boundaries $\mathcal{F}_i$ overlap. In the feature space $\mathcal{X}_1$ of the example in Figure 4, this is represented by the red, the green and the orange line having an approximate angle of 60° to each other. The left loop is represented by an eigenvector $\widehat{e}_{\mathcal{F}}$ with support on the green and orange edges, and vice-versa the right loop by $\widehat{e}_{\mathcal{F}'}$ with support on the green and red edges. The homology flow on the middle line on the green edges is a linear combination of the homology flows of both generating loops.

4. Theoretical Guarantees for Synthetic Data

In this section, we give a result showing that the algorithm works on a class of synthetic point clouds with an arbitrary number of topological features in arbitrary dimensions. The proof utilises the core ideas of the previous section. An easy way to realise a flexible class of topological space is to work with the wedge sum operator $\vee$ gluing the two spaces together at a fixed base point. For $k > 0$ and two topological spaces X and Y we have that $B_k(X \vee Y) = B_k(X) + B_k(Y)$. Hence the wedge sum combines topological features.

Theorem 4.1. *Let $\mathbb{P} \subset \mathbb{R}^n$ be a finite point cloud in $\mathbb{R}^n$ that is sampled from a space X. Furthermore, let $X = \bigvee_{i \in \mathcal{I}} \mathbb{S}_i^{d_i}$ with finite indexing set $\mathcal{I}$ with $|\mathcal{I}| > 1$ and $0 < d \in \mathbb{N}$ be a bouquet of spheres . We assume that the geometric realisation of the simplicial approximation $\mathcal{S}$ is homotopy-equivalent to X, and furthermore that the simplicial subcomplexes for the $\mathbb{S}^{d_i}$ only overlap in the base-point, and divide $\mathbb{S}^{d_i}$ into d_i-simplices.*

Then topological point cloud clustering recovers the different spheres and the base point accurately.

The full proof is given in Appendix D.

5. Numerical Experiments

Comparison with k-means and spectral clustering We validated the effectiveness of TPCC on a number of synthetic examples. In Figure 5, we have clustered points sampled randomly from two spheres and two circles. The algorithm recovers the spheres and circles. Normal (zero-dimensional) Spectral Clustering and k-means fail in choosing the right notion of feature, as the figure shows. For a visual comparison of TPCC with other clustering algorithms on various datasets see Figure 9 in the appendix.

	TPCC	SpC	k-means	OPTICS	DBSCAN	AC	Mean Shift	AP	ToMATo
2 spheres, 2 circles (Figure 5)	**0.97**	0.70	0.48	0.01	0.00	0.66	0.84	0.01	0.90
Toy example (Figure 3)	**0.98**	0.33	0.28	0.19	0.11	0.33	0.81	0.00	0.91
Circle with line (Figure 4)	**0.85**	0.23	0.16	0.11	0.00	0.25	0.00	0.23	0.09
Sphere in circle, `noise` $= 0$ (Figure 7 top)	**1.00**	0.34	0.02	0.19	0.00	0.29	0.00	0.12	0.06
Sphere in circle, `noise` $= 0.3$ (Figure 7 bottom)	**0.53**	0.28	0.01	0.22	0.30	0.27	0.00	0.13	0.46
Energy landscape (Figure 6 left)	**0.88**	0.01	0.01	0.00	0.00	0.13	0.00	0.01	-0.02

Table 1. **Quantitative performance comparison of TPCC with popular clustering algorithms.** We show the Adjusted Rand Index of TPCC, Spectral Clustering (SpC), k-means, OPTICS, DBSCAN, Agglomerative Clustering (AC), Mean Shift Clustering, Affinity Propagation (AP), and Topological Mode Analysis Tool clustering (ToMATo) evaluated on six data sets. On every data set TPCC performs best, indicating that the other algorithm are not designed for clustering points according to higher-order topological features.

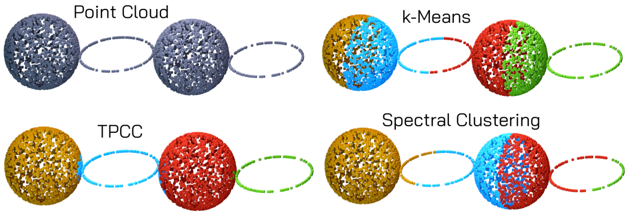

Figure 5. TPCC is the only approach correctly distinguishing the spheres and circles.

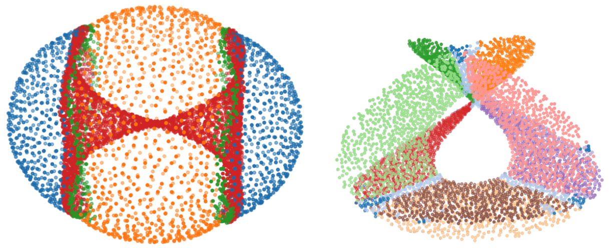

Figure 6. Left: Energy landscape of cyclo-octane clustered by topological point cloud clustering. We have four different clusters, with the green one being the anomalous points. *Right:* Clustering of the Henneberg surface.

Comparison to Manifold Anomaly Detection In (Stolz et al., 2020), the authors propose a topological method for detecting anomalous points on manifolds. In Figure 6 we use TPCC on the same datasets (Martin & Watson, 2011; Adams et al., 2014) to show that our approach is also able to detect the anomalous points. Additionally, our method can classify the remaining points based on topological features.

Experiments with Synthetic Data As we make use of topological features, TPCC is robust against noise by design. We compare the accuracy of the clustering algorithm against k-means and spectral clustering on a point cloud consisting of a sphere, a circle, and a connecting line in Figure 7.

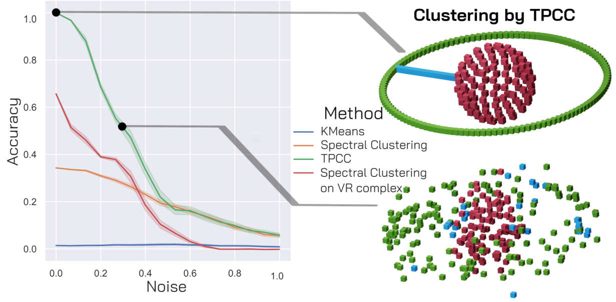

Figure 7. We have added i.i.d. Gaussian noise with varying standard deviation specified by the parameter `noise` on all three coordinates of every point. (For scale: The radius of the inner sphere is 1.) *Left:* Accuracy of TPCC, k-Means and two versions of Spectral Clustering with increasing noise level. *Spectral clustering* uses the radial basis affinity matrix, as implemented in scikit-learn. *Spectral Clustering on VR complex* uses the underlying graph of the simplicial complex used for TPCC. Accuracy is measured by adjusted rand index and averaged over 100 samples. *Right:* Example point clouds used for testing and clustering obtained by TPCC for `noise` $= 0.0$ and `noise` $= 0.3$.

On low to medium noise levels, TPCC significantly outperforms all other clustering methods. On higher noise levels, the topological features of the point cloud degenerate to features that can be measured by ordinary spectral clustering. Then, TPCC and spectral clustering achieve similar accuracy scores. In Figure 7 we see that already a noise setting of `noise` $= 0.3$ distorts the point cloud significantly, yet TPCC still performs well.

Proteins Proteins are molecules that consist of long strings of amino acid residues. They play an integral role in almost every cellular process from metabolism, DNA replication, to intra-cell logistics. Their diverse functions are hugely influenced by their complex 3d geometry, which arises by folding the chains of amino acid residues. The available data of protein sequences and 3d structure has increased dramatically over the last decades. However, functional annotations of the sequences, providing a gateway for understanding protein behaviour, are missing for most of the proteins. (Smaili et al., 2021) have shown that harnessing structural information on the atoms can significantly increase prediction accuracy of ML pipelines for functional annotations. Thus being able to extract topological information on individual atoms of proteins is very desirable for applications in drug discovery, medicine, and biology.

We tested TPCC on NALCN channelosome, a protein found in the membranes of human neurons (Zhou et al., 2022;

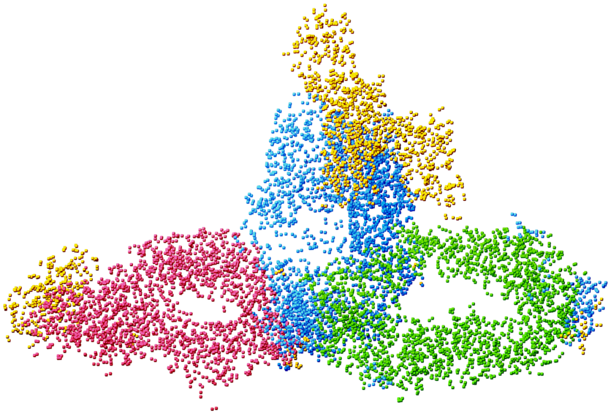

Figure 8. Clustered atoms of NALCN channelosome. Points that border one of the holes are coloured red, blue, and green. The points without contribution to a loop are marked in yellow.

Kschonsak et al., 2022). The NALCN channel regulates the membrane potential, enabling neurons to modulate respiration, circadian rhythm, locomotion and pain sensitivity. It has a complex topological structure enclosing 3 holes that are linked to its function as a membrane protein.

The core idea is that when biological and topological roles correlate, TPCC offers a way to better understand *both*.

6. Discussion

Limitations TPCC can only cluster according to features that are visible to homology, e.g. connected components, loops, holes, and cavities. For example, TPCC cannot distinguish differently curved parts of lines or general manifolds. TPCC constructs a simplicial complex (SC) to extract topological information Thus it needs to pick a single scale for every SC. If the topological information of the point cloud lie in different scales, TPCC thus needs to do multiple feature aggregation steps for SCs of different scale. Finally, the points can be clustered according to the combined features. However, for each different scale the entire zero-eigenspace of the Hodge-Laplacian needs to be considered. Future work will focus on a method to cluster points based on the most persistent topological features across all scales.

Persistent homology and the calculation of the zero eigenvectors of the Hodge Laplacian are computationally expensive and thus running TPCC directly is not feasible on large data sets. However, usually the topological information can already be encoded in small subsets of the entire point cloud. In Table 2 we show that TPCC in combination with landmark sampling scales well for larger data sets while achieving high clustering performance. In addition, we believe that the main advantage of TPCC is that it can do something no other existing point cloud clustering algorithm can do or was designed for, namely clustering points according to higher order topological features. Future work will focus on additionally improving efficiency by removing the need to compute the entire zero-eigenspace of the Hodge-Laplace operators.

Because TPCC uses persistent homology, it is robust against small perturbations by design. In Figure 7 we analysed its clustering performance under varying levels of noise. However, with high noise levels, topological features vanish from persistent homology and thus TPCC cannot detect them anymore. In future work, we try to take near-zero eigenvectors of the Hodge Laplacian into account, representing topological features contaminated by noise. This is similar to Spectral Clustering, where the near-zero eigenvectors represent almost-disconnected components of the graph.

Conclusion TPCC is a novel clustering algorithm respecting topological features of the point cloud. We have shown that it performs well both on synthetic data and real-world data and provided certain theoretical guarantees for its accuracy. TPCC produces meaningful clustering across various levels of noise, outperforming k-means and classical spectral clustering on several tasks and incorporating higher-order information.

Due to its theoretical flexibility, TPCC can be built on top of various simplicial or cellular representations of point clouds. Interesting future research might explore combinations with the mapper algorithms or cellular complexes. In particular, applications in large-scale analysis of protein data constitute a possible next step for TPCC. TPCC or one of its intermediate steps has potential as a pre-processing step for deep learning techniques, making topological information about points accessible for ML pipelines.

Acknowledgments

The authors thank the anonymous reviewers for helpful feedback that improved the paper.

VPG and MTS acknowledge funding by the DFG (German Research Foundation) – Research Training Group UnRAVeL 2236/2. MTS acknowledges funding by the Ministry of Culture and Science (MKW) of the German State of North Rhine-Westphalia (NRW Rückkehrprogramm) and the European Union (ERC, HIGH-HOPeS, 101039827). Views and opinions expressed are however those of MTS only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency; neither the European Union nor the granting authority can be held responsible for them.

References

Adams, H., Tausz, A., and Vejdemo-Johansson, M. javaplex: A research software package for persistent (co)homology. In Hong, H. and Yap, C. (eds.), *Mathematical Software – ICMS 2014*, pp. 129–136, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

Anai, H., Chazal, F., Glisse, M., Ike, Y., Inakoshi, H., Tinarrage, R., and Umeda, Y. Dtm-based filtrations. In *Topological Data Analysis: The Abel Symposium 2018*, pp. 33–66. Springer, 2020.

Bodnar, C., Frasca, F., Wang, Y., Otter, N., Montufar, G. F., Lio, P., and Bronstein, M. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pp. 1026–1037. PMLR, 2021.

Bredon, G., Ewing, J., Gehring, F., and Halmos, P. *Topology and Geometry*. Graduate Texts in Mathematics. Springer, New York, 1993.

Bubenik, P. et al. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, 2015.

Bunch, E., You, Q., Fung, G., and Singh, V. Simplicial 2-complex convolutional neural networks. In *NeurIPS Workshop: TDA & Beyond*, 2020.

Carlsson, G. and Vejdemo-Johansson, M. *Topological Data Analysis with Applications*. Cambridge University Press, 2021.

Chazal, F., Guibas, L. J., Oudot, S. Y., and Skraba, P. Persistence-based clustering in riemannian manifolds. *J. ACM*, 60(6), nov 2013.

Chen, G. and Lerman, G. Spectral curvature clustering (scc). *International Journal of Computer Vision*, 81(3): 317–330, Mar 2009. doi: 10.1007/s11263-008-0178-9.

Chen, Y.-C. and Meila, M. The decomposition of the higher-order homology embedding constructed from the k-laplacian. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 15695–15709. Curran Associates, Inc., 2021.

Day, N. E. Estimating the components of a mixture of normal distributions. *Biometrika*, 56(3):463–474, 1969.

Ebli, S. and Spreemann, G. A notion of harmonic clustering in simplicial complexes. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1083–1090, 2019. doi: 10.1109/ICMLA.2019.00182.

Ebli, S., Defferrard, M., and Spreemann, G. Simplicial neural networks. In *NeurIPS Workshop: TDA & Beyond*, 2020.

Eckmann, B. Harmonische Funktionen und Randwertaufgaben in einem Komplex. *Commentarii mathematici Helvetici*, 17:240–255, 1944/45.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pp. 226–231, 1996.

Fefferman, C., Mitter, S., and Narayanan, H. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, Oct 2016. doi: 10.1090/jams/852.

Frantzen, F., Seby, J.-B., and Schaub, M. T. Outlier detection for trajectories via flow-embeddings. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pp. 1568–1572. IEEE, 2021.

Friedman, J. Computing betti numbers via combinatorial laplacians. *Algorithmica*, 21(4):331–346, Aug 1998. doi: 10.1007/PL00009218.

Gebhart, T., Fu, X., and Funk, R. J. Go with the flow? a large-scale analysis of health care delivery networks in the united states using hodge theory. In *2021 IEEE International Conference on Big Data (Big Data)*, pp. 3812–3823, Dec 2021. doi: 10.1109/BigData52589.2021.9671805.

Gong, D., Zhao, X., and Medioni, G. Robust multiple manifolds structure learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pp. 25–32, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

Halko, N., Martinsson, P. G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011. doi: 10.1137/090771806.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2.

Hatcher, A. *Algebraic Topology*. Cambridge University Press, Cambridge, 2002.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. Surface reconstruction from unorganized points. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '92, pp. 71–78, New York, NY, USA, 1992. Association for Computing Machinery.

Keros, A. D., Nanda, V., and Subr, K. Dist2cycle: A simplicial neural network for homology localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7133–7142, 2022.

Kovacev-Nikolic, V., Bubenik, P., Nikolić, D., and Heo, G. Using persistent homology and dynamical distances to analyze protein binding. *Statistical Applications in Genetics and Molecular Biology*, 15(1), Jan 2016. doi: 10.1515/sagmb-2015-0057.

Krishnagopal, S. and Bianconi, G. Spectral detection of simplicial communities via hodge laplacians. *Physical Review E*, 104(6):064303, Dec 2021. doi: 10.1103/PhysRevE.104.064303.

Kschonsak, M., Chua, H. C., Weidling, C., Chakouri, N., Noland, C. L., Schott, K., Chang, T., Tam, C., Patel, N., Arthur, C. P., Leitner, A., Ben-Johny, M., Ciferri, C., Pless, S. A., and Payandeh, J. Structural architecture of the human nalcn channelosome. *Nature*, 603(7899): 180–186, Mar 2022. doi: 10.1038/s41586-021-04313-5.

Lehoucq, R., Sorensen, D., and Yang, C. *ARPACK users' guide: Solution of large-scale eigenvalue problems with implicitly restarted arnoldi methods*. Software, environments, tools. Society for Industrial and Applied Mathematics, 1998.

Martin, S. and Watson, J.-P. Non-manifold surface reconstruction from high-dimensional point cloud data. *Computational Geometry*, 44(8):427–441, 2011.

Obayashi, I. Volume-optimal cycle: Tightest representative cycle of a generator in persistent homology. *SIAM Journal on Applied Algebra and Geometry*, 2(4):508–534, 2018.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Perea, J. A. Sparse circular coordinates via principal $\mathbb{Z}$-bundles. In Baas, N. A., Carlsson, G. E., Quick, G., Szymik, M., and Thaule, M. (eds.), *Topological Data Analysis*, pp. 435–458, Cham, 2020. Springer International Publishing.

Quillen, D. G. *Homotopical Algebra*, volume 43 of *Lecture Notes in Mathematics*. Springer, Berlin, 1967.

Roddenberry, T. M., Glaze, N., and Segarra, S. Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning*, pp. 9020–9029. PMLR, 2021.

Schaub, M. T., Benson, A. R., Horn, P., Lippner, G., and Jadbabaie, A. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Review*, 62 (2):353–391, 2020.

Schaub, M. T., Zhu, Y., Seby, J.-B., Roddenberry, T. M., and Segarra, S. Signal processing on higher-order networks: Livin' on the edge... and beyond. *Signal Processing*, 187: 108149, 2021. doi: https://doi.org/10.1016/j.sigpro.2021.108149.

Silva, V. d. and Carlsson, G. Topological estimation using witness complexes. In Gross, M., Pfister, H., Alexa, M., and Rusinkiewicz, S. (eds.), *SPBG'04 Symposium on Point - Based Graphics 2004*. The Eurographics Association, 2004. doi: 10.2312/SPBG/SPBG04/157-166.

Singh, G., Mémoli, F., and Carlsson, G. Topological methods for the analysis of high dimensional data sets and 3d object recognition. *Eurographics Symposium on Point-Based Graphics*, pp. 10, 2007.

Smaili, F. Z., Tian, S., Roy, A., Alazmi, M., Arold, S. T., Mukherjee, S., Hefty, P. S., Chen, W., and Gao, X. Qaust: Protein function prediction using structure similarity, protein interaction, and functional motifs. *Genomics, Proteomics and Bioinformatics*, 19(6):998–1011, 2021.

Steinhaus, H. Sur la division des corps matériels en parties. *Bull. Acad. Pol. Sci., Cl. III*, 4:801–804, 1957.

Stolz, B. J., Tanner, J., Harrington, H. A., and Nanda, V. Geometric anomaly detection in data. *Proceedings of the National Academy of Sciences*, 117(33):19664–19669, 2020.

Stolz, B. J., Kaeppler, J., Markelc, B., Braun, F., Lipsmeier, F., Muschel, R. J., Byrne, H. M., and Harrington, H. A. Multiscale topology characterizes dynamic tumor vascular networks. *Science Advances*, 8(23), 2022.

The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2015.

Tinarrage, R. Recovering the homology of immersed manifolds. *Discrete & Computational Geometry*, pp. 1–86, 2023.

tom Dieck, T. *Algebraic topology*, volume 8. European Mathematical Society, Zürich, 2008.

von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, Dec 2007. doi: 10.1007/s11222-007-9033-z.

Wang, Y., Jiang, Y., Wu, Y., and Zhou, Z.-H. Spectral clustering on multiple manifolds. *IEEE Transactions on Neural Networks*, 22(7):1149–1161, 2011.

Wells, C. blendplot, 2017.

Zhou, L., Liu, H., Zhao, Q., Wu, J., and Yan, Z. Architecture of the human nalcn channelosome. *Cell Discovery*, 8(1): 33, Apr 2022. doi: 10.1038/s41421-022-00392-4.

Zografos, V., Ellis, L., and Mester, R. Discriminative subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2107–2114, 2013.

A. Implementation

To construct the simplicial complex used in TPCC from a point cloud, we first computed a persistence diagram. Then we selected the parameter ε in the range of the most persistent homology features. Hence, we connected all points p_1 and p_2 with $\|p_1 - p_2\|_2 < \varepsilon$. We also chose until which dimension we build the simplicial complex by looking at the topological features of the underlying point cloud. In practice, on all considered data sets the maximum dimension of topological features was 2. Hence building the simplicial complex up to dimension 3 suffices: We note that computing information on the k-th homology group and the k-th Betti number requires the simplices of dimension up to $k+1$. This is reflected in the shape of the k-th Hodge Laplacian $L_k := \mathcal{B}_{k-1}^\top \mathcal{B}_{k-1} + \mathcal{B}_k \mathcal{B}_k^\top$ featuring $\mathcal{B}_k$. The k-th boundary matrix $\mathcal{B}_k$ maps $(k+1)$-simplices to k-simplices.

Computational Complexity Persistent homology and the calculation of the zero eigenvectors of the Hodge Laplacian are computationally expensive and thus TPCC in its pure form does not scale well for large data sets. The complexity of random sparse eigensolvers is approximately $O(kT + k^2 n)$ for $n \times n$ matrices where k is the number of desired eigenvectors and T is the number of flops required for one sparse matrix vector multiplication (Halko et al., 2011). The number of non-zero values in the k-th Hodge-Laplacian is bounded by the number of ordered pairs of upper-adjacent and of lower-adjacent k-simplices. For a fixed point density, fixed ε, and fixed k, the number of k-simplices n is linear in the number of points.

However, we believe that the main advantage of TPCC is that it can do something no other existing point cloud clustering algorithm can do or was designed for, namely clustering points according to higher order topological features.

		2spheres	6spheres
	Number of points	4600	33 600
TPCC	Landmark sampling	0.7 s	48.7 s
	Persistent homology	2.0 s	4.1 s
	Eigenvector computation	3.6 s	31.4 s
	Sum of times	6.3 s	**84.2 s**
	Adjusted Rand Index	0.93	0.94
TPCC+witness	Landmark sampling	0.7 s	48.7 s
	Witness complex	0.2 s	615.5 s
	Persistent homology	0.5 s	4.7 s
	Eigenvector computation	5.4 s	19.7 s
	Sum of times	6.8 s	688.6 s
	Adjusted Rand Index	**0.95**	**0.97**
SpC	Time	**1.7 s**	346.3 s
	Adjusted Rand Index	0.71	0.47

Table 2. We test the scalability of clustering approaches using TPCC. We compare the accuracy the running time and the Adjusted Rand Index (ARI) of TPCC and Spectral Clustering on the data set of Figure 5 and on a version with more points, spheres and circles. We also compare two different versions of constructing the SC in step 1 of TPCC: We used a naive python implementation of min-max landmark sampling to select respectively 400 or 1200 landmarks. For the first version we constructed the Vietoris-Rips complex directly on this point cloud. For TPCC+witness we constructed the witness complex based on the landmarks and the entire data set. After running TPCC, we cluster the remaining points using a 1-nearest neighbour approach. Both the TPCC approaches achieved a significantly higher ARI than Spectral Clustering. On the large data set, TPCC using landmark sampling and a VR construction had a significantly smaller running time then basic Spectral Clustering.

Because TPCC is agnostic to the type of simplicial complex constructed, its computational scalability can easily be improved by using a more efficient construction than Vietoris-Rips. Usually, the topological information of a data set is already contained in a small subset of the points. It is thus possible to use a witness complex construction (Silva & Carlsson, 2004), or to sample landmark points representing the topological structure, doing TPCC on them, and then clustering the remaining points using k-nearest neighbours.

In Table 2, we show that using TPCC in conjunction with min-max landmark sampling and a k-nearest neighbour approach to classify the remaining points scales well to larger data sets while maintaining a high accuracy.

Min-Max landmark sampling Min-max landmark sampling provides a way to approximate the topology of a point cloud using a set of landmarks L. For a point cloud X, a distance function $d \colon X \times X \to \mathbb{R}_{\geq 0}$, and a desired number of landmarks $1 \leq k \leq |X|$ we first sample a point $x \in X$ uniformly at random and add it to the set of landmarks L. Then we iteratively add the point $x \in X$ maximising the expression

$$\min_{l \in L} d(l, x)$$

to L until $|L| = k$. (For example, compare (Perea, 2020).)

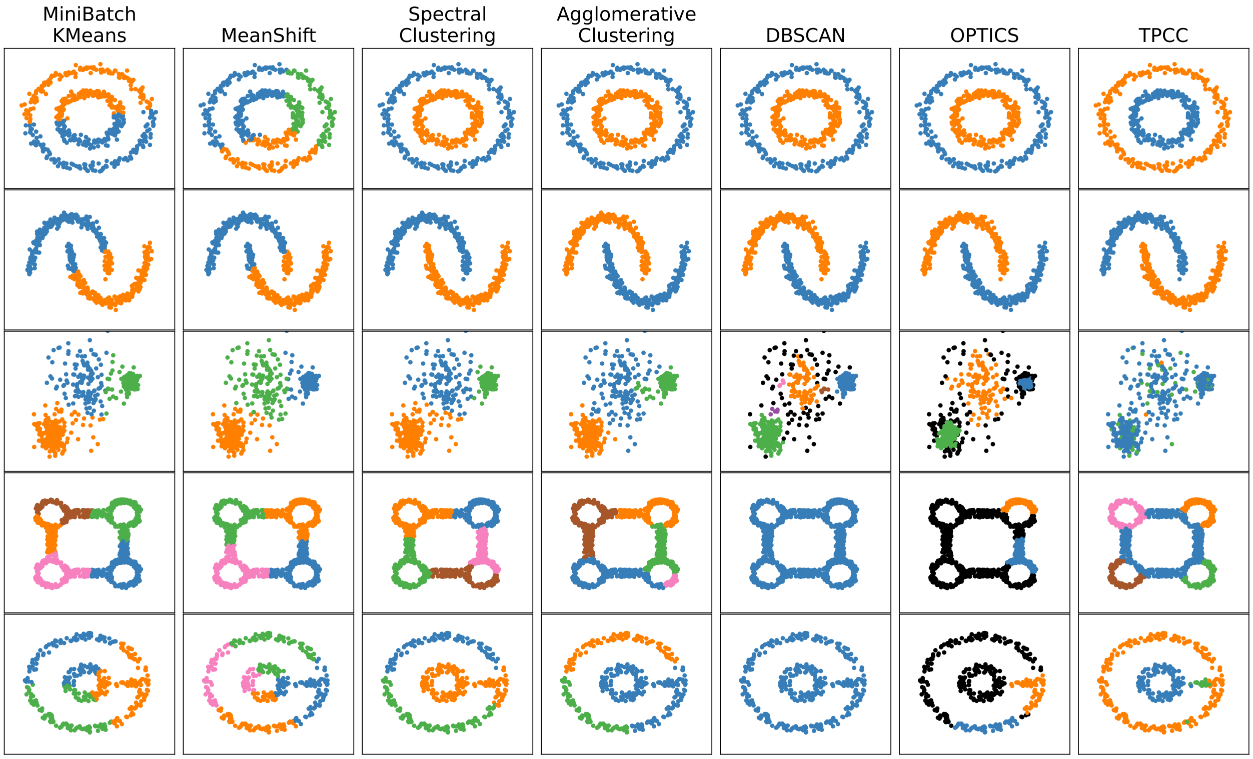

Figure 9. **Comparison of clusterings produced by TPCC and other clustering algorithms on existing and new datasets.** While TPCC is not able to find any structure in the third dataset, it identifies the topological substructures in the fourth and fifth dataset. (Cf. `scikit-learn`, (Pedregosa et al., 2011)) While TPCC is the only algorithm clustering the four connected circles respecting the topology, it still is inconsistent in whether to assign the shared parts of circles and rectangle to the cluster of the circle or the connecting lines. Theoretically, these shared parts would belong to 4 additional clusters. However, this would require the subspace clustering to identify 9 different linear subspaces, which is not feasible with the implementation of subspace clustering we were using.

Witness complex The (weak) witness complex, as introduced in (Silva & Carlsson, 2004), provides a way to approximate the topology of a large point cloud by a simplicial complex with significantly fewer vertices. It takes as input the original point cloud X, a set of landmarks L in an ambient Euclidean space, and a parameter R determining the length of edges. It then constructs a simplicial complex on L, where the simplices are added based on whether they are ”witnessed” by points in X. While witness complexes are very robust topological approximators, their construction is computationally demanding for large point clouds X.

Supplementary material Code of our implementation to reproduce the experimental results will be made available in the supplementary material.

Software used We implemented the algorithm in python. We use the Gudhi library (The GUDHI Project, 2015) for all topology-related computations and operations. For general arithmetic and clustering purposes we use NumPy (Harris et al., 2020), scikit-learn (Pedregosa et al., 2011), and ARPACK (Lehoucq et al., 1998). For subspace clustering, we use DiSC (Zografos et al., 2013). For 3d visualisation, we use blender with blendplot (Wells, 2017).

B. Running Example

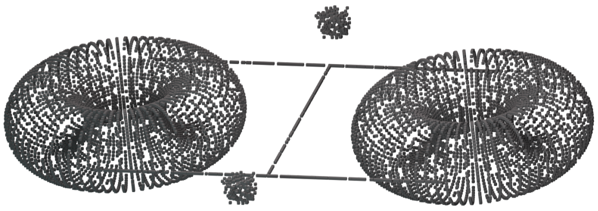

Figure 10. The point cloud of our toy example projected to 3d space. Every point is represented by a small cube.

Our toy example consists of two tori. A torus can be seen as a doughnut where we removed the filling. Topologically, it consists of a single connected component. Hence its 0-th Betti number B_0 is 1, counting the number of connected components. There are two main directions a 1-dimensional loop can wrap around a torus. First, there is the large loop going around the entire circle spanned by the torus. Second, a loop can just wrap around a side of a torus. All other loops can be generated by concatenation of the previous two types of loops. Hence the 1st Betti number B_1 is 2, counting the number of generating loops. Finally, there is a single 2-dimensional cavity in a torus, representing the void left behind by removing the filling of the doughnut. Thus, the 2nd Betti number B_2 of a torus is 2. We can embed a torus in 4-dimensional space by taking it to be the product of two 1-dimensional spheres. Note that we project the tori to 3-dimensional space only for better readability in our plots. We sample the point cloud by first taking 5000 points in a grid on each of the tori. We then randomly forget 20% of the points in order to simulate noise. The tori are connected by two straight lines, from which we each sample 300 points uniformly at random. We connect the two lines by another straight line with 300 randomly sampled points. The three lines add two more loops to the topological space Finally, we sample 200 points uniformly at random from two cubes not connected with the rest of the topological space. Our point cloud has 11 topological features across 3 dimensions. In terms of Betti numbers, we have $B_0 = 3$, $B_1 = 6$, and $B_2 = 2$.

C. Technical considerations

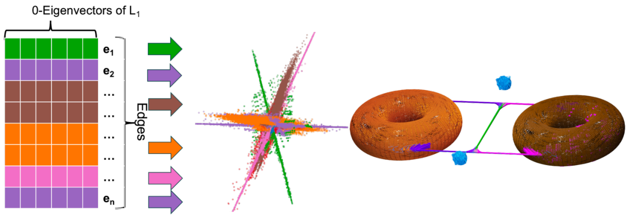

Figure 11. We cluster the edges of the simplicial complex $\mathcal{S}$ depicted in Figure 1, our toy example. Its first Betti number $B_1(\mathcal{S})$ is 6, corresponding to a 6-dimensional zero-eigenspace of L_1. We show a projection of the 6-dimensional feature space $\mathcal{X}_1$ to 3-dimensional space. There are six different subspace clusters: three 1-dimensional lines in purple, green, and pink corresponding to ordinary loops in the point cloud. Furthermore, there are two 2-dimensional subspaces marked in brown and orange. They represent the edges in the two tori of our data set. Finally, there is one 0-dimensional cluster corresponding to the edges without a contribution to homology in the two cubes marked in blue.

Multi-dimensional subspaces of the feature spaces $\mathcal{X}_k$ In practice, most of the subspaces of the feature space $\mathcal{X}_k$ used for subspace clustering are 1-dimensional[1]. However, sometimes more complex substructures arise: Recall that our toy point cloud (Figure 1, Input) consisted of two tori. The 1-homology of each of the tori is generated by two loops, one of which follows the larger circle of the associated filled doughnut, the other one encircling a slice of the doughnut. Now imagine an edge e starting in one of the outermost points of the torus. If the edge faces in a left or right direction, it will only contribute to the first loop. If it faces up or down, it only contributes o the second loop. However, an edge can point in an arbitrary superposition of the two directions. Thus also its homological contribution will be an arbitrary superposition of the two generating loops of the tori. In other words, the embeddings into the feature space $\iota(e) \in \mathcal{X} = \mathbb{R}^{B_k}$ of the edges e running along the generating loops of the tori correspond to points on two orthogonal lines. The embeddings of all other edges on the surface of the torus lie on the 2-dimensional subspace of $\mathcal{X}_1$ spanned by the two lines. Because the angles of the edges can vary continuously, the edges correspond to arbitrary points on the 2-dimensional subspace. Thus, we propose clustering the edges based on membership in arbitrary-dimensional subspaces. In the toy point cloud example, we can hence measure to which torus an edge belongs by identifying the 2-dimensional subspace its eigenvector coordinates lie on. This is illustrated in Figure 11. By allowing for detection of arbitrary dimensional subspaces of $\mathcal{X}_1$ our approach is able to detect significantly more topological features than the precursor approach in (Ebli & Spreemann, 2019).

Extracting the dimensionality of topological features TPCC not only distinguishes different topological features, it is also capable of extracting additional information on the features. In particular, there are two ways the framework can distinguish between different dimensionalities of the features:

I. A 1-dimensional loop will appear in the zero-eigenspace of the 1-Hodge Laplacian, whereas a 2-dimensional boundary of a void will appear in the 0-eigenspace of the 2-Hodge Laplacian. This information can easily be relayed back to the points.

II. Topological features will manifest as linear subspaces of different dimensions of the zero-eigenspaces of the corresponding Hodge Laplace operators. Usually, these subspaces will be 1-dimensional. The subspace corresponding to the first homology group of the torus is however 2-dimensional. (Cf. the previous paragraph.) This is because edges on the torus can point in arbitrary superpositions of the two ”generating homological dimensions”. (This, by Hurewicz’s thm., corresponds to the respective generators of the fundamental group commuting with each other.) We can view this as the feature being another notion of 2-dimensional and relay the information back to the points.

[1] Although we could regard the simplices without homological contribution as lying in a 0-dimensional subspace.

D. Proof of Theorem 4.1

Theorem. *Let $\mathbb{P} \subset \mathbb{R}^n$ be a finite point cloud in $\mathbb{R}^n$ that is sampled from a space X. Furthermore, let $X = \bigvee_{i \in \mathcal{I}} \mathbb{S}_i^{d_i}$ with finite indexing set $\mathcal{I}$ with $|\mathcal{I}| > 1$ and $0 < d \in \mathbb{N}$ be a bouquet of spheres . We assume that the geometric realisation of the simplicial approximation $\mathcal{S}$ is homotopy-equivalent to X, and furthermore that the simplicial subcomplexes for the $\mathbb{S}^{d_i}$ only overlap in the base-point, and divide $\mathbb{S}^{d_i}$ into d_i-simplices.*

Then topological point cloud clustering recovers the different spheres and the base point accurately.

Proof. The k-th Betti number of $\mathcal{S}$ is equal to the number of $i \in \mathcal{I}$ with $d_i = k$ (Cor. 2.25 (Hatcher, 2002)). Because spheres are orientable, we can simply assume that the d_i-simplices in $\mathbb{S}_i^{d_i}$ are oriented such that each two adjacent d_i-simplices induce opposite orientations on the shared (d_i)-simplex. We now claim that for each $i \in \mathcal{I}$ the indicator vector e_i on the d_i-simplices in $\mathbb{S}_i^{d_i}$ is an eigenvector of the d_i-th Hodge Laplacian L_i of $\mathcal{S}$. Because of our assumption on $\mathcal{S}$, there are no $(d_i + 1)$-simplices upper-adjacent to the d_i-simplices of $\mathbb{S}_i^{d_i}$. Hence, we obtain the first half of our claim, namely that $\mathcal{B}_{d_i}\mathcal{B}_{d_i}^\top e_i = 0$ holds. We have assumed that $\mathcal{S}$ was constructed in such a way that each $(d_i - 1)$-simplex $\sigma_{d_i - 1}$ of $\mathbb{S}_i^{d_i}$ has exactly two upper-adjacent neighbours $\sigma_{d_i}^1$ and $\sigma_{d_i}^2$. Because $\sigma_{d_i}^1$ and $\sigma_{d_i}^2$ induce the opposite orientation on $\sigma_{d_i - 1}$, the corresponding entries of the $(d_i - 1)$-th boundary matrix $\mathcal{B}_{d_i - 1}$ of $\mathcal{S}$ are 1 and -1. Thus we also have $\mathcal{B}_{d_i - 1} e_i = 0$ and finally $L_{d_i} e_i = \mathcal{B}_{d_i}\mathcal{B}_{d_i}^\top e_i + \mathcal{B}_{d_i - 1}^\top \mathcal{B}_{d_i - 1} e_i = 0$. This proves the claim.

The eigenvectors e_i of the same dimension are orthogonal and match in number with the corresponding Betti number of $\mathcal{S}$. Hence the e_i span the eigenspaces of the Hodge Laplace operators of $\mathcal{S}$. For all $i \in \mathcal{I}$ the entries of the d_i-simplices in $\mathbb{S}_i^{d_i}$ in the matching zero eigenvectors e_j are 1 for $j = i$, and 0 else. All other d-simplices for $d > 0$ have trivial eigenvector entries. Thus, subspace clustering recovers the top-level simplices in each of the spheres and assigns every other simplex to the trivial homology cluster. The topological signature of the points in the sphere $\mathbb{S}_i^{d_i}$ in dimension d_i will then feature a characteristic cluster of (d_i)-simplices and a trivial signature across the other dimensions. Finally, the topological signatures of the base point will feature all characteristic clusters. Hence k-means on the topological signatures can distinguish the points on the different spheres and the base point. □