# NODE-LEVEL TOPOLOGICAL REPRESENTATION LEARNING ON POINT CLOUDS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Topological Data Analysis (TDA) allows us to extract powerful topological, and higher-order information on the global shape of a data set or point cloud. Tools like Persistent Homology or the Euler Transform give a *single* complex description of the *global structure* of the point cloud. However, common machine learning applications like classification require *point-level* information and features to be available. In this paper, we bridge this gap and propose a novel method to extract node-level topological features from complex point clouds using discrete variants of concepts from algebraic topology and differential geometry. We verify the effectiveness of these topological point features (TOPF) on both synthetic and real-world data and study their robustness under noise.

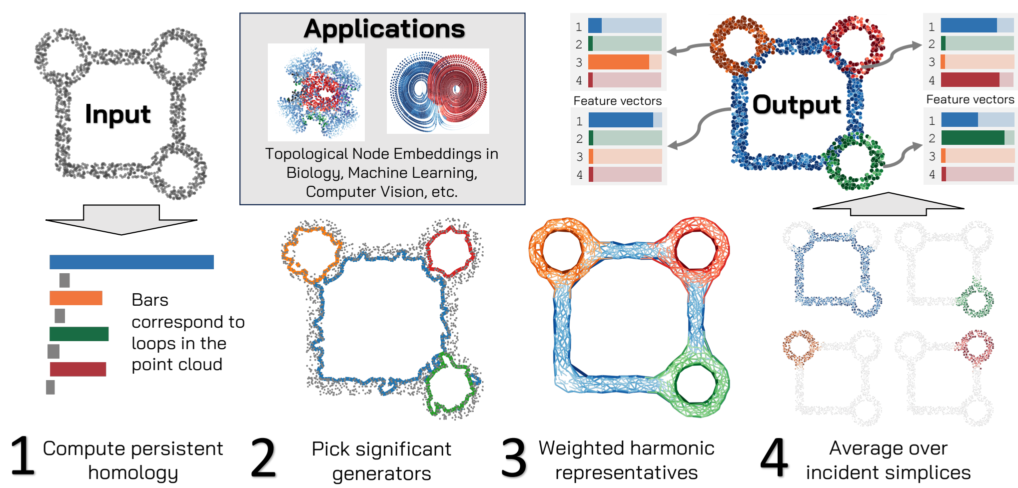


Figure 1: **Schematic of Computing Topological Point Features (TOPF)**. **Input.** A point cloud $X$ in $n$-dimensional space. **Step 1.** To extract global topological information, the persistent homology is computed on an $\alpha$/VR-filtration. The most significant topological features $\mathcal{F}$ across all specified dimensions are selected. **Step 2.** $k$-homology generators associated to all features $f_{i,k} \in \mathcal{F}$ are computed. For every feature, a simplicial complex is built at a step of the filtration where $f_{i,k}$ is alive. **Step 3.** The homology generators are projected to the harmonic space of the simplices. **Step 4.** The vectors are normalised to obtain vectors $\mathbf{e}_\mathbf{k}^\mathbf{i}$ indexed over the $k$-simplices. For every point $x$ and feature $f \in \mathcal{F}$, we compute the mean of the entries of $\mathbf{e}_\mathbf{k}^\mathbf{i}$ corresponding to simplices containing $x$. The output is a $|X| \times |\mathcal{F}|$ matrix which can be used for downstream ML tasks. **Optional.** We weigh the simplicial complexes resulting in a topologically more faithful harmonic representative in **Step 3**.

## 1 INTRODUCTION

In modern machine learning (Murphy, 2022), objects are described by feature vectors within a high-dimensional space. However, the coordinates of a single vector can often only be understood in relation to the entire data set: if the value $x$ is small, average, large, or even an outlier depends on the remaining data. In a 1-dimensional (or low-dimensional) case this issue can be addressed simply by normalising the data points according to the global mean and standard deviation or similar procedures. We can interpret this as the most straight-forward way to construct *local* features informed by the *global* structure of the data set.

In the case where not all data dimensions are equally relevant, or contain correlated and redundant information, we can apply (sparse) PCA to project the data points to a lower-dimensional space using information about the *global structure* of the point cloud (Zou et al., 2006). For even more complex data, we may first have to learn the encoded structure itself: indeed, a typical assumption underpinning many unsupervised learning methods is the so-called "manifold hypothesis" which posits that real world data can be described well via submanifolds of $n$-dimensional space (Ma & Fu, 2012; Fefferman et al., 2016). Using eigenvectors of some Laplacian, we can then obtain a coordinate system intrinsic to the point cloud (see e.g. Shi & Malik (2000); Belkin & Niyogi (2003); Coifman & Lafon (2006)). Common to all these above examples is the goal is to construct locally interpretable point-level features that encode *globally meaningful positional information* robust to local perturbations of the data. However, none of these approaches is able to represent higher-order topological information (as captured by homology in degrees above 0), making point clouds with these kind of structure inaccessible to point-level machine learning algorithms.

Instead of focussing on the interpretation of individual points, topological data analysis (TDA), Carlsson & Vejdemo-Johansson (2021), follows a different approach. TDA extracts a global description of the shape of data, which is typically considered in the form of a high-dimensional point cloud. This is done measuring topological features like persistent homology, which counts the number of generalised "holes" in the point cloud on multiple scales. Due to their flexibility and robustness these global topological features have been shown to contain relevant information in a broad range of application scenarios: In medicine, TDA has provided methods to analyse cancer progression (Lawson et al., 2019). In biology, persistent homology has been used to analyse knotted protein structures (Benjamin et al., 2023), and the spectrum of the Hodge Laplacian has been used for predicting protein behaviour (Wee et al., 2024).

This success of topological data analysis is a testament to the fact that relevant information is encoded in the global topological structure of point cloud data. Such higher-order topological information is however invisible to standard tools of data analysis like PCA or $k$-means clustering, and can also not be captured by graph models of the point cloud. We are now faced by the problem that **(i)** important parts of the global structure of a complex point cloud can only be described by the language of applied topology, however **(ii)** most standard methods to obtain positional point-level information are not sensitive to the higher-order topology of the point cloud. The goal of our paper is to solve the above-mentioned problem.

**Contributions** We introduce TOPF (Figure 1), a novel method to compute **node-level topological features** relating individual points to global topological structures of point clouds. TOPF **(i)** *outperforms* other methods and embeddings for clustering downstream tasks on topologically structured data, returns **(ii)** *provably meaningful representations*, and is **(iii)** *robust to noise*. Finally, we introduce the topological clustering benchmark suite, the first benchmark for topological clustering.

**Related Work** The intersection of topological data analysis, topological signal processing and geometry processing has many interesting related developments in the past few years. On the side of homology and TDA, the authors in De Silva & Vejdemo-Johansson (2009) and Perea (2020) use harmonic *co*homology representatives to reparametrise point clouds based on circular coordinates. This implicitly assumes that the underlying structure of the point cloud is amenable to such a characterization.

In Basu & Cox (2022); Gurnari et al. (2023), the authors develop and use harmonic persistent homology for data analysis. However, among other differences their focus is not on providing robust topological point features. Carrière et al. (2015) construct point features using Topology as well. However, their signatures only summarise the local topology of the neighbourhood of the point rather than the relation between the point and the *global* topological features. Grande & Schaub (2023a) uses the harmonic space of the Hodge Laplacians to cluster point clouds respecting topology, but is unstable against some form of noise, has no possibility for features selection across scales and is computationally far more expensive than TOPF. An overview over the thriving field of topological data analysis can be found in (Wasserman, 2018; Munch, 2017). For a more in-depth review of related work, see Appendix A. Because there are different views on what constitutes *representation learning*, we note that the learnt representations of TOPF are the result of *homology and linear algebra computations*, rather than trained features of an autoencoder or other neural network.

**Organisation of the paper** In Section 2, we give an overview over the main ideas and concepts behind of TOPF. In Section 3, we describe how to compute TOPF. Finally, we will apply TOPF on synthetic and real-world data in Section 4. Furthermore, Appendix A contains a brief history of topology and a detailed discussion of related work. Appendix B contains additional theoretical considerations. In Appendix C, we give a theoretical result guaranteeing the correctness of TOPF. Appendix D describes the novel topological clustering benchmark suite, Appendix E contains details on the implementation and the choice of hyperparameters, Appendix G gives a detailed treatment of feature selection, Appendix H discusses simplicial weights, and Appendix I discusses limitations in detail.

**Code** We provide TOPF as an easy-to-use python package with example Jupyter Notebooks in the supplementary material.

## 2 Main Ideas of TOPF

A main goal of algebraic topology is to capture the shape of spaces. Techniques from topology describe globally meaningful structures that are indifferent to local perturbations and deformations. This robustness of topological features to local perturbations is particularly useful for the analysis of large-scale noisy datasets. To apply the ideas of algebraic topology in our TOPF pipeline, we need to formalise and explain the notion of *topological features*. An important observation for this is that high-dimensional point clouds and data may be seen as being sampled from topological spaces — most of the time, even low-dimensional submanifolds of $\mathbb{R}^n$ (Fefferman et al., 2016).

In this section we provide a broad overview over the most important concepts of topology and TDA for our context, prioritising intuition over technical formalities. The interested reader is referred to Bredon et al. (1993); Hatcher (2002); tom Dieck (2008) for a complete technical account of topology and Munch (2017) for an overview over TDA.

**Simplicial Complexes** Spaces in topology are *continuous* (*connected*), consist of *infinitely* many points, and often live in *abstract space*. Our input data sets however consist of *finitely* many points embedded in *real space* $\mathbb{R}^n$. In order to bridge this gap and open up topology to computational methods, we need a notion of discretised topological spaces consisting of finitely many base points with finite description length. A *Simplicial Complex* is the simplest discrete model that can still approximate any topological space occuring in practice (Quillen, 1967):

**Definition 2.1** (Simplicial complexes)**.** A *simplicial complex* (SC) $\mathcal{S}$ consists of a set of vertices $V$ and a set of finite non-empty subsets (simplices, $S$) of $V$ closed under taking non-empty subsets, such that the union over all simplices $\bigcup_{\sigma\in S}\sigma$ is $V$. In the following, we will often identify $\mathcal{S}$ with its set of simplicies $S$ and denote by $\mathcal{S}_k$ the set of simplices $\sigma \in S$ with $|\sigma| = k+1$, called *$k$-simplices*. We say that $\mathcal{S}$ is $n$-dimensional, where $n$ is the largest $k$ such that the set of $k$-simplices $\mathcal{S}_k$ is non-empty. The *$k$-skeleton* of SC contains the simplices of dimension at most $k$. If the vertices $V$ lie in real space $\mathbb{R}^n$, we call the convex hull in $\mathbb{R}^n$ of a simplex $\sigma$ its *geometric realisation* $|\sigma|$. When doing this for every simplex of $\mathcal{S}$, we call this the *geometric realisation of* $\mathcal{S}$, $|\mathcal{S}| \subset \mathbb{R}^n$.

Concretely, we can construct an $n$-dimensional SC $\mathcal{S}$ in $n+1$ steps: First, we start with a set of vertices $V$ which we can identify with the 0-simplices $\mathcal{S}_0$. Second, we connect certain pairs of vertices with edges, which constitute the set of 1-simplices. We can then choose to fill in some triples of vertices which are fully connected by 1-simplices with triangles, i.e. 2-simplices. More generally, in the $k^{\text{th}}$ step, we can add a $k$-simplex for every set $\sigma_k$ of $k+1$ vertices such that every $k$-element subset $\sigma_{k-1}$ of $\sigma_k$ is already a $(k-1)$-simplex.

**Vietoris–Rips and $\alpha$-complexes** We now need a way to construct a *simplicial complex* that approximates the *topological structure* inherent in our data set $X \subset \mathbb{R}^n$. Such a construction will always depend on the scale of the structures we are interested in. When looking from a very large distance, the point cloud will appear as a singular connected blob in the otherwise empty and infinite real space, on the other hand when we continue to zoom in, the point cloud will at some point appear as a collection of individual points separated by empty continuous space; all interesting information can be found in-between these two extreme scales where some vertices are joined by simplices and others are not. Instead of having to pick a single scale, the *Vietoris–Rips (VR) filtration* and the *$\alpha$-filtration* take as input a point cloud and return a nested sequence of simplicial complexes indexed by a scale parameter $\varepsilon$ approximating the topology of the data across all possible scales.

**Definition 2.2** (VR complex)**.** Given a finite point cloud $X$ in a metric space $(\mathcal{M}, d)$ and a non-negative real number $\varepsilon \in \mathbb{R}_{\geq 0}$, the associated VR complex $VR_\varepsilon(X)$ is given by the vertex set $X$ and the set of simplices $S = \{\sigma \subset X \mid \sigma \neq \emptyset, \forall x, y \in \sigma : d(x, y) \leq \varepsilon\}$.

Intuitively, a VR complex with parameter $\varepsilon$ consists of all simplices $\sigma$ where all vertices $x \in \sigma$ have a pair-wise distance of at most $\varepsilon$. For $r \leq r'$, we obtain the canonical inclusions $i_{r,r'}(X) \colon VR_r(X) \hookrightarrow VR_{r'}(X)$. For a simplex $\sigma$ and a filtration $F$, we denote by its *filtration value* $F(\sigma)$ the smallest $\varepsilon$ such that $\sigma \in F_\varepsilon$. The set of VR complexes on $X$ for all possible $r \in \mathbb{R}_{\geq 0}$ together with the inclusions then form the *VR filtration* on $X$. For large point clouds, using the VR complex for computations becomes expensive due to its large number of simplices. In contrast, the more sophisticated $\alpha$-complex approximates the topology of a point cloud using far fewer simplices. Thus we will make use of $\alpha$-complexes in settings of ambient dimension lower than 4, where their construction is computationally feasible. For a complete account and definition of $\alpha$-complexes and our reason to use them, see Appendix B. We note that the filtration value of a $k$-simplex in the $\alpha$-filtration is related to the radius of its circum-$k$-sphere, which differs from its filtration value in the VR filtration.

**Boundary matrices** So far, we have discussed a discretised version of topological spaces in the form of SCs and a way to turn point clouds into a sequence of SCs indexed by a scale parameter. However, we still need an *algebraic representation* of simplicial complexes that is capable of encoding the structure of the SC and enables extraction of the *topological features*: The *boundary matrices* $\mathcal{B}_k$ associated to an SC $\mathcal{S}$ store all structural information of the SC. The rows of $\mathcal{B}_k$ are indexed by the $k$-simplices of $\mathcal{S}$ and the columns are indexed by the $(k+1)$-simplices.

**Definition 2.3** (Boundary matrices)**.** Let $\mathcal{S}$ be a simplicial complex and $\preceq$ a total order on its vertices $V$. Then, the $i$-th face map in dimension $n$ $f_i^n \colon \mathcal{S}_n \to \mathcal{S}_{n-1}$ is given by

$$f_i^n \colon \{v_0, v_1, \ldots, v_n\} \mapsto \{v_0, v_1, \ldots, \widehat{v}_i, \ldots, v_n\}$$

with $v_0 \preceq v_1 \preceq \cdots \preceq v_n$ and $\widehat{v}_i$ denoting the omission of $v_i$. Now, the $n$-th *boundary operator* $\mathcal{B}_n \colon \mathbb{R}[\mathcal{S}_{n+1}] \to \mathbb{R}[\mathcal{S}_n]$ with $\mathbb{R}[\mathcal{S}_n]$ being the real vector space over the basis $\mathcal{S}_n$ is given by

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

When lexicographically ordering the simplex basis, we can view $\mathcal{B}_n$ as a *matrix*. We call $\mathbb{R}[\mathcal{S}_n]$ the space of $n$-chains. Now, $\mathcal{B}_0$ is the vertex-edge incidence matrix of the associated graph consisting of the 0- and 1-simplices of $\mathcal{S}$ and $\mathcal{B}_1$ is the edge-triangle incidence matrix of $\mathcal{S}$

**Betti Numbers and Persistent Homology** We now turn to the notion of *topological features* and how to extract them. *Homology* is one of the main algebraic invariants to capture the shape of topological spaces and SC. From a technical point of view, the $k$-th homology module $H_k(\mathcal{S})$ of an SC $\mathcal{S}$ with boundary operators $\mathcal{B}_k$ is defined as $H_k(\mathcal{S}) \coloneqq \ker \mathcal{B}_{k-1} / \operatorname{Im} \mathcal{B}_k$. The *generator* or representative of a homology class is an element of the kernel $\ker \mathcal{B}_{k-1}$. In dimension 1, these are given by formal sums of 1-simplices forming closed loops in the SC. Importantly, the rank $\operatorname{rk} H_k(\mathcal{S})$ is called the $k$-th *Betti number* $B_k$ of $\mathcal{S}$. In dimension 0, $B_0$ counts the number of connected components, $B_1$ counts the number of loops around 'holes' of the space, $B_2$ counts the number of 3-dimensional voids with 2-dimensional boundary, and so on.

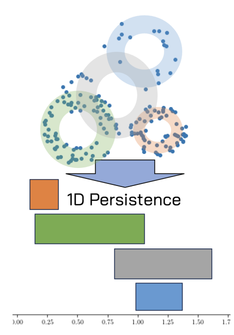


Figure 2: Sketch of Persistent Homology, Grande & Schaub (2023b)

If we are now given a filtration of simplicial complexes instead of a single SC, we can track how the homology modules *evolve* as the simplicial complex *grows*. The mathematical formalisation, *persistent homology*,

**Algorithm 1** Topological Point Features (TOPF)

**Input:** Point cloud $X \in \mathbb{R}^n$, maximum homology dimension $d \in \mathbb{N}$, interpolation coeff. $\lambda$.
**1.** Compute persistent homology with generators in dimension $k \leq d$.
**2.** Select set of significant features $(b_i, d_i, g_i)$ with birth, death, and generator in $\mathbb{F}_3$ coordinates.
**3.** Embed $g_i$ into real space and project into harmonic subspace of SC at step $t = d_i^\lambda b_i^1 - \lambda$ or $t = \lambda d_i + (1-\lambda) b_i$.
**4.** Normalise projections to $\mathbf{e}_i^k$ and compute $F_k^i(x) \coloneqq \operatorname{avg}_{x\in\sigma}(\mathbf{e}_i^k l(\sigma))$ for all points $x \in X$.
**Output:** Features of $x \in X$

thus turns a point cloud via a simplicial filtration into an algebraic object summarising the topological features of the point cloud (Edelsbrunner & Harer, 2008). For better computational performance, the computations are usually done in one of the small finite fields $\mathbb{Z}/p\mathbb{Z}$. Because we will later be interested in the sign of numbers to distinguish different simplex orientations, we will use $\mathbb{Z}/3\mathbb{Z}$-coefficients, with $\mathbb{Z}/3\mathbb{Z}$ being the smallest field being able to distinguish 1 and $-1$.

**The Hodge Laplacian and the Harmonic Space** In the previous part, we have introduced a language to characterise the global shape of spaces and point clouds. However, we still need to find a way to relate these *global characterisations* back to *local properties* of the point cloud. We will do so by using ideas and concepts from differential geometry and topology: The simplicial Hodge Laplacian is a discretisation of the Hodge–Laplace operator acting on differential forms of manifolds:

**Definition 2.4** (Hodge Laplacian)**.** Given a simplicial complex $\mathcal{S}$ with boundary operators $\mathcal{B}_k$, we define the $n$-th Hodge Laplacian $L_n \colon \mathbb{R}[\mathcal{S}_n] \to \mathbb{R}[\mathcal{S}_n]$ by setting

$$L_n \coloneqq \mathcal{B}_{n-1}^\top \mathcal{B}_{n-1} + \mathcal{B}_n \mathcal{B}_n^\top .$$

The Hodge Laplacian gives rise to the Hodge decomposition theorem:

**Theorem 2.5** (Hodge Decomposition (Lim, 2020; Schaub et al., 2021; Roddenberry et al., 2021))**.** *For an SC $\mathcal{S}$ with boundary matrices $(\mathcal{B}_i)$ and Hodge Laplacians $(L_i)$, we have in every dimension $k$*

$$\mathbb{R}[\mathcal{S}_k] = \underbrace{\operatorname{Im} \mathcal{B}_{k-1}^\top}_{\textit{gradient space}} \oplus \underbrace{\ker L_k}_{\textit{harmonic space}} \oplus \underbrace{\operatorname{Im} \mathcal{B}_k}_{\textit{curl space}} .$$

This, together with the fact that the $k$-th harmonic space is isomorphic to the $k$-th real-valued homology group $\ker L_k \cong H_k(\mathbb{R})$ means that we can associate a *unique harmonic representative* to every homology class. The harmonic space encodes higher-order generalisations of smooth flow around the holes of the simplicial complex. Intuitively, this means that for every *abstract global homology class* of persistent homology at filtration step $t$ from above we can now compute one *unique harmonic representative* in $\ker L_k$ that assigns every simplex a value based on how much it contributes to the homology class. Thus, the Hodge Laplacian is a gateway between the *global topological features* and the *local properties* of our SC. It is easy to show that the kernel of the Hodge Laplacian is the intersection of the kernel of the boundary and the coboundary map $\ker L_k = \ker \mathcal{B}_{n-1} \cap \ker \mathcal{B}_n^\top$. Because we have finite SCs we can identify the spaces of chains and cochains. This leads to another characterisation of the harmonic space: The space of chains that are simultaneously homology and cohomology representatives. We discuss how this relates to the theory of differential forms and Hodge theory in the continuous case in Appendix B.2.

## 3 How to Compute Topological Point Features

In this section, we will combine the ideas and insights of the previous section to give a complete account of how to compute topological point features (TOPF). A pseudo-code version can be found in Algorithm 1 and an overview in Figure 1. We start with a finite point cloud $X \subset \mathbb{R}^n$.

**Step 1: Computing the persistent homology** First, we need to determine the *most significant persistent homology classes* which determine the shape of the point cloud. By doing this, we can also extract the "interesting" scales of the data set. We will later use this to construct SCs to derive

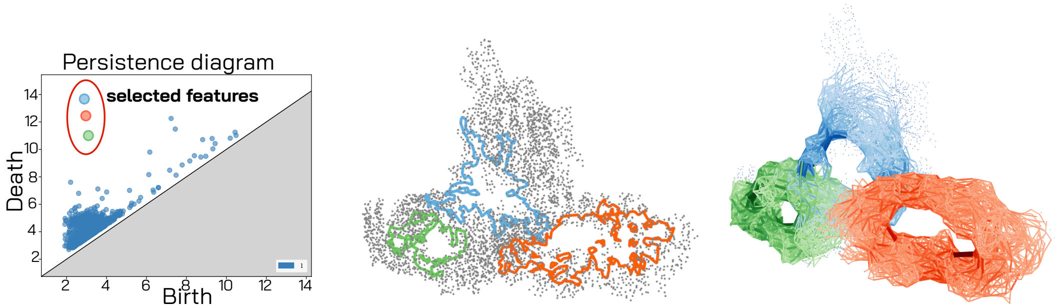


Figure 3: **TOPF pipeline applied to NALCN channelosome, a membrane protein (Kschonsak et al., 2022).** *Left:* Steps **1&2a**, when computing persistent 1-homology, three classes are more prominent than the rest. *Centre:* Step **2b**: The selected homology generators. *Right:* Step **3**: The projections of the generators into harmonic space are now each supported on one of the rings.

local variants of the global homology features. Thus we first compute the persistent $k$-homology modules $P_k$ including a set of homology representatives $R_k$ of $X$ using an $\alpha$-filtration for $n \leq 3$ and a VR filtration for $n > 3$. We use $\mathbb{Z}/3\mathbb{Z}$ coefficients to be sensitive to simplex orientations. In case we have prior knowledge on the data set, we can choose a real number $R \in \mathbb{R}_{>0}$ and only compute the filtration and persistent homology connecting points up to a distance of at most $R$. In data sets like protein atom coordinates, this might be useful as we have prior knowledge on what constitutes the "interesting" scale, reducing computational complexity. See Figure 3 *left* for a PH diagram.

**Step 2: Selecting the relevant topological features** We now need to select the relevant *homology classes* which carry the most important *global information*. The persistent homology $P_k$ module in dimension $k$ is given to us as a list of pairs of birth and death times $(b_i^k, d_i^k)$. We can assume these pairs are ordered in non-increasing order of the durations $l_i^k = d_i^k - b_i^k$. This list is typically very long and consists to a large part of noisy homological features which vanish right after they appear. In contrast, we are interested in connected components, loops, cavities, etc. that *persist* over a long time, indicating that they are important for the shape of the point cloud. Distinguishing between the relevant and the irrelevant features is in general difficult and may depend on additional insights on the domain of application. In order to provide a heuristic which does not depend on any a-priori assumptions on the number of relevant features we pick the smallest quotient $q_i^k := l_{i+1}^k / l_i^k > 0$ as the point of cut-off $N_k := \arg\min_i q_i^k$. The only underlying assumption of this approach is that the band of "relevant" features is separated from the "noisy" homological features by a drop in persistence. If this assumption is violated, the only possible way to do meaningful feature selection depends on application-specific domain knowledge. We found that our proposed heuristics work well across a large scale of applications. See Figure 3 *left* and *centre* for an illustration and Appendix G for more technical details and ways to improve and adapt the feature selection module of TOPF. We call the chosen $k$-homology classes including $k$-homology generators $f_k^i$.

**Step 3: Projecting the features into harmonic space and normalising** In this step, we need to relate the *global topology* extracted in the previous step to the simplices which we will use to compute the *local* topological point feature. Every selected feature $f_k^i$ of the previous step comes with a birth time $b_{i,k}$ and a death time $d_{i,k}$. This means that the homology class $f_k^i$ is present in every SC of the filtration between step $\varepsilon = b_{i,k}$ and $\varepsilon = d_{i,k}$ and we could choose any of the SCs for the next step. Picking a *small* $\varepsilon$ will lead to *fewer* simplices in the SC and thus to a very *localised* harmonic representative. Picking a *large* $\varepsilon$ will lead to *many* simplices in the SC and thus to a very *smooth* and "blurry" harmonic representative with large support. Finding a middle ground between these regimes returns optimal results. For the interpolation parameter $\gamma \in (0, 1)$, we will thus consider the simplicial complex $\mathcal{S}^{t_{i,k}}(X)$ at step $t_{i,k} := b_{i,k}^{1-\gamma} d_{i,k}^{\gamma}$ for $k > 0$ and at step $t_{i,k} := \gamma d_{i,k}$ for $k = 0$ of the simplicial filtration. At this point, the homology class $f_k^i$ is still alive. We then consider the real vector space $\mathbb{R}[\mathcal{S}_k^{t_{i,k}}(X)]$ with formal basis consisting of the $k$-simplices of the SC $\mathcal{S}^{t_{i,k}}$. From the persistent homology computation of the first step, we also obtain a generator of the feature $f_k^i$, consisting of a list $\Sigma_k^i$ of simplices $\hat{\sigma}_j \in \mathcal{S}_k^{b_{i,k}}$ and coefficients $c_j \in \mathbb{Z}/3\mathbb{Z}$. We need to turn this formal sum of simplices with $\mathbb{Z}/3\mathbb{Z}$-coefficients into a vector in the real vector space $\mathbb{R}[\mathcal{S}_k^{t_{i,k}}(X)]$: Let $\iota\colon \mathbb{Z}/3\mathbb{Z}$ be the map induced by the canonical inclusion of $\{-1, 0, 1\} \hookrightarrow \mathbb{R}$. We can now define an indicator vector $e_k^i \in \mathbb{R}[\mathcal{S}_k^{t_{i,k}}(X)]$ associated to the feature $f_k^i$ (Cf. De Silva & Vejdemo-Johansson

(2009)).

$$e_k^i(\sigma) \coloneqq \begin{cases} \iota(c_j) & \exists \hat{\sigma}_j \in \Sigma_k^i : \sigma = \hat{\sigma}_j \\ 0 & \text{else} \end{cases}.$$

Empirically, $e_k^i$ is a homology generator for all real coefficients as well, although our construction only guarantees for $\mathcal{B}_{k-1} e_k^i \equiv 0 \mod 3$. We discuss how to fix the rare case where this does not work in Appendix B.3 and now assume to work with a real homology representative $e_k^i$. While this homology representative lives in a real vector space, it is not unique, has a small support, and its value can differ largely even between close simplices. All of these problems can be solved by projecting the homology representative to the harmonic subspace $\ker L_k$ of $\mathbb{R}[\mathcal{S}_k^{t_{i,k}}(X)]$. Rather than directly projecting $e_k^i$ to the harmonic subspace, we make use of the Hodge decomposition theorem (Theorem 2.5) which allows us to solve computationally efficient least square problems:

$$e_{k,\text{grad}}^i \coloneqq \mathcal{B}_{k-1}^\top \operatorname*{arg\,min}_{x \in \mathbb{R}[\mathcal{S}_{k-1}]} \left\| e_k^i - \mathcal{B}_{k-1}^\top x \right\|_2^2 \quad \text{and} \quad e_{k,\text{curl}}^i \coloneqq \mathcal{B}_k \operatorname*{arg\,min}_{x \in \mathbb{R}[\mathcal{S}_{k+1}]} \left\| e_k^i - e_{k,\text{grad}}^i - \mathcal{B}_k x \right\|_2^2$$

and then obtain the **harmonic representative** $\hat{e}_k^i \coloneqq e_k^i - e_{k,\text{grad}}^i - e_{k,\text{curl}}^i$. (Cf. Figure 3 *right* for a visualisation.) Because homology representatives are gradient-free, we only need to consider the projection of $e_k^i$ into the curl space.

**Step 4: Processing and aggregation at a point level** In the previous step, we have computed a set of harmonic representatives of homology classes. Each harmonic representative assigns each simplex in the correspondings SCs a value. However, these simplices likely have no real-world meaning and the underlying simplicical complexes differ depending on the birth and death times of the homology classes. Hence in this step, we will collect the features on the point-level after performing some necessary preprocessing. Given a vector $\hat{e}_k^i$ indexed over simplices and a hyperparameter $\delta$, we now construct $\mathbf{e}_k^i \colon \mathcal{S}_k^{t_{i,k}}(X) \to [0,1]$ by setting $\mathbf{e}_k^i \colon \sigma \mapsto \in \{|\hat{e}_k^i(\sigma)|/(\delta \max_{\sigma' \in \mathcal{S}_k^{t_{i,k}}(X)} |\hat{e}_k^i(\sigma')|), 1\}$ such that $\hat{e}_k^i$ is normalised to $[0,1]$, then the values of $[0,\delta]$ are mapped linearly to $[0,1]$ and everything above is sent to 1. We found empirically that a thresholding parameter of $\delta = 0.07$ works best across at the range of applications considered below. However, TOPF is not sensitive to small changes to $\delta$ because entries of $\hat{e}_k^i$ are concentrated around 0 (Cf. Appendix E.1).

For every feature $f_k^i$ in dimension $k$ with processed simplicial feature vector $\mathbf{e}_k^i$ and simplicial complex $\mathcal{S}^{t_{i,k}}$, we define the point-level feature map $F_i^k \colon X \to \mathbb{R}$ mapping from the initial point cloud $X$ to $\mathbb{R}$ by setting

$$F_i^k \colon v \mapsto \frac{\sum_{\sigma_k \in \mathcal{S}_k^{t_{i,k}} \colon v \in \sigma_k} \mathbf{e}_k^i(\sigma_k)}{\max(1, |\{\sigma_k \in \mathcal{S}_k^t : v \in \sigma_k\}|)}.$$

For every point $v$, we can thus view the vector $(F_i^k(v) \colon f_i^k \in \mathcal{F})$ as a feature vector for $v$. We call this collection of features *Topological Point Features* (TOPF). (Cf. Figure 4 for an example).

**Choosing Simplicial Weights** By default, the simplicial complexes of $\alpha$- and VR filtrations are unweighted. However, the weights determine the entries of the harmonic representatives, increasing and decreasing the influence of certain simplices and parts of the simplicial complex. We can use this observation to increase the robustness of TOPF against the influence of heterogeneous point cloud structure, which is present in virtually all real-world data sets. For a complete technical account of how and why we do this, see Appendix H.

## 4 Experiments

In this section, we conduct experiments on real world and synthetic data, compare the clustering results with clustering by TPCC, other classical clustering algorithms, and other point features, and demonstrate the robustness of TOPF against noise.

**Topological Point Cloud Clustering Benchmark** We introduce the topological clustering benchmark suite (Appendix D) and report running times and the accuracies of clustering based on TOPF and other methods and point embeddings, see Table 1. We see that spectral clustering on TOPF vectors

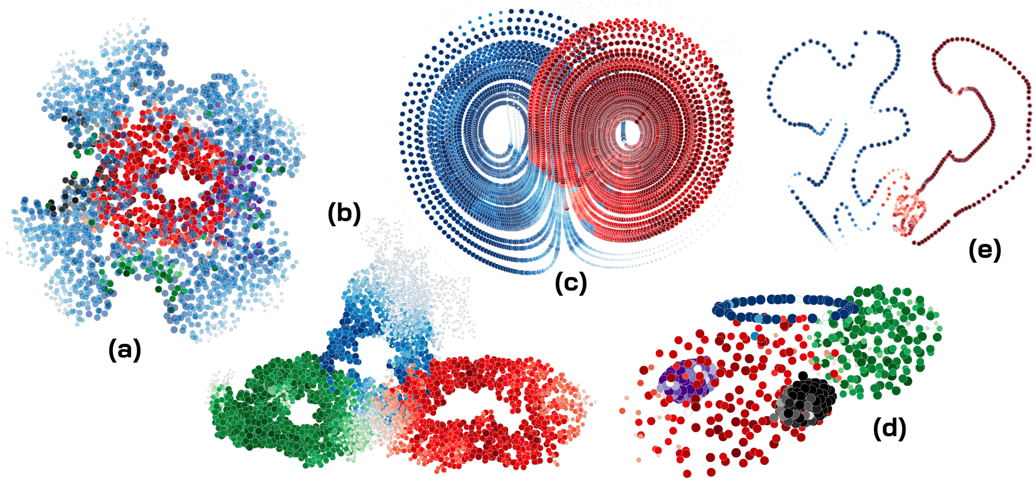


Figure 4: **TOPF on $3D$ real-world and synthetic point clouds.** For every point, we highlight the largest corresponding topological feature, where colour stands for the different features and saturation for the value of the feature. *(a):* Atoms of mutated Cys123 of E. coli (Hidber et al., 2007). We added auxiliary points on the convex hull and considered 2-homology, to detect the protein pockets which are crucial for protein-environment interactions (Cf. Oda et al. (2024)) (*Green, black, purple:* Detected side pockets, *Red:* Central hole). *(b):* Atoms of NALCN Channelosome (Kschonsak et al., 2022) display three distinct loops. *(c):* Points sampled in the state space of a Lorentz attractor. The two features correspond to the two lobes of the attractor. *(d):* Point cloud `spaceship` of our newly introduced topological clustering benchmark suite (See Appendix D). *(e):* Latent space of a VAE trained on image patches showing topological structure (See Figure 13 for details).

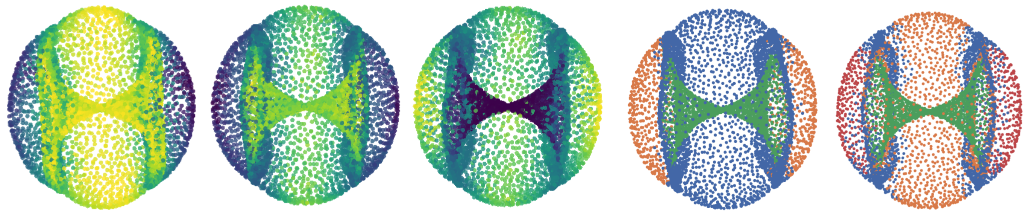

Figure 5: **TOPF on a high-dimensional point cloud.** We used TOPF on 6500 points sampled from the 24-dimensional conformation space of cyclooctane Martin & Watson (2011). Due to the dimension constraints of papers, we have to show the ISOMAP projection from 24 dimensions down to 3 dimensions. *Left:* Three (one 1-dimensional and two 2-dimensional) features automatically selected by TOPF. *Right:* Result of clustering for three (automatically chosen by TOPF) and four clusters.

(listed as TOPF) *outperforms* all classical clustering algorithms on all but one dataset by a wide margin. We also see that TOPF closely matches the performance of the only other higher-order topological clustering algorithm, TPCC on two datasets with clear topological features, whereas TOPF *outperforms* TPCC on datasets with more complex structure. In addition, TOPF has a consistently lower running time with better scaling for the more complex datasets, while not requiring prior knowledge on the best topological scale. As for the other point embeddings, Node2Vec is not able to capture any meaningful topological information, whereas the performance of clustering using geometric features depends on the data set. Furthermore, TOPF outperforms PointNet (Qi et al., 2017) pretrained on the ShapeNetPart data set and WSDesc (Li et al., 2022) pretrained on the 3DMatch dataset (Zeng et al., 2017). This does not highlight the fact that TOPF outperforms PointNet or WSDesc in general but rather that neural network architectures like PointNet need specific application-specific training data for good performance that is simply not available in all cases. We note that PointNet is not the current SOTA on ShapenetPart Segmentation (Chang et al., 2015).

**Feature Generation** In Figure 4, we show qualitatively that TOPF constructs meaningful topological features on data sets from Biology and Physics, and synthetic data, corresponding to for example rings and pockets in proteins or trajectories around different attractors in dynamical systems, (for

Table 1: **Quantitative performance comparison of clustering with TOPF and other features/clustering algorithms.** Four $2D$ and three $3D$ data sets of the topological clustering benchmark suite (Appendix D, cf. Figure 9 for ground truth and Figure 10 for labels by TOPF). We ran each algorithm 20 times and list the mean adjusted rand index (ARI) with standard deviation $\sigma$ and mean running time. We omit $\sigma$ for algorithms with $\sigma = 0$. Spectral clustering on TOPF vectors consistently outperforms or almost matches the other algorithms while having significantly better run time than the second best performing algorithm TPCC. Spectral Clustering (SC), DBSCAN, and Agglomerative Clustering (AgC) (applied on the points) are standard clustering algorithms, ToMATo is a topological clustering algorithm (Chazal et al., 2013), Geo clusters using 12-dimensional point geometric features extracted by `pgeof`, whereas node2vec (Grover & Leskovec, 2016) produces node embeddings on a $k$-nearest neighbour graph b.We pretrained PointNet (PN) (Qi et al., 2017) on the ShapeNetPart point cloud segmentation data set (Chang et al., 2015). “Weakly Supervised 3D Local Descriptor Learning for Point Cloud Registration” (WSD (Li et al., 2022)) is pretrained on 3DMatch data and produces 32-dimensional feature vectors.We highlight all ARI scores within $\pm 0.05$ of the best ARI score.

| | | **TOPF** | TPCC | SC | DBSCAN | AgC | ToMATo | Geo | node2vec | PN | WSD |
|---|---|---|---|---|---|---|---|---|---|---|---|
| `4spheres` | ARI | **0.81** | 0.52±0.17 | 0.37 | 0.00 | 0.45 | 0.32 | 0.20 | 0.00±0.00 | 0.30 | 0.13 |
| | time (s) | 14.5 | 23.3 | 0.2 | <0.1 | <0.1 | <0.1 | 0.2 | 48.4 | <0.1 | <0.1 |
| `Ellipses` | ARI | **0.95** | 0.47±0.04 | 0.25 | 0.19 | 0.52 | 0.29 | 0.81 | 0.02±0.00 | 0.50 | 0.35 |
| | time (s) | 12.7 | 14.4 | 0.1 | <0.1 | <0.1 | <0.1 | 0.1 | 11.2 | <0.1 | <0.1 |
| `4`$\bigcirc$`+Grid` | ARI | 0.70 | 0.39±0.04 | **0.90** | **0.92** | **0.89** | 0.82 | 0.41 | 0.01±0.00 | 0.55 | 0.06 |
| | time (s) | 13.0 | 28.5 | 0.5 | <0.1 | <0.1 | <0.1 | 0.3 | 63.8 | <0.1 | <0.1 |
| `Halved` $\bigcirc$ | ARI | **0.71** | 0.18±0.12 | 0.24 | 0.00 | 0.20 | 0.16 | 0.08 | 0.00±0.01 | 0.36 | 0.00 |
| | time (s) | 12.2 | 14.3 | 0.1 | <0.1 | <0.1 | <0.1 | 0.1 | 18.2 | <0.1 | <0.1 |
| `2Spheres2`$\bigcirc$ | ARI | **0.94** | **0.97**±0.01 | 0.70 | 0.00 | 0.51 | 0.87 | 0.12 | 0.00±0.00 | 0.55 | **0.95** |
| | time (s) | 38.9 | 1662.2 | 1.6 | <0.1 | 0.3 | <0.1 | 0.9 | 348.6 | <0.1 | <0.1 |
| `Spherein`$\bigcirc$ | ARI | **0.97** | **0.98**±<0.1 | 0.34 | 0.00 | 0.29 | 0.06 | 0.69 | 0.13±0.03 | 0.39 | 0.46 |
| | time (s) | 14.5 | 8.0 | <0.1 | <0.1 | <0.1 | <0.1 | 0.08 | 20.1 | <0.1 | <0.1 |
| `Spaceship` | ARI | **0.92** | 0.56±0.03 | 0.28 | 0.26 | 0.47 | 0.30 | **0.87** | 0.07±0.00 | 0.41 | 0.76 |
| | time (s) | 16.3 | 341.8 | 16.7 | <0.1 | <0.1 | <0.1 | 0.2 | 49.8 | <0.1 | <0.1 |
| **mean** | ARI | **0.86** | 0.58 | 0.44 | 0.16 | 0.48 | 0.40 | 0.45 | 0.03 | 0.44 | 0.39 |
| | time (s) | 17.5 | 298.9 | 0.4 | <0.1 | <0.1 | <0.1 | 0.3 | 80.0 | <0.1 | <0.1 |

individual heatmaps see Figure 14). In Figure 5, we show that TOPF works for the *high-dimensional* conformation space of cyclooctane.

**Robustness Against Noise, Downsampling, and Sampling Heterogeneity** We have evaluated the robustness of TOPF against Gaussian noise on the dataset introduced in (Grande & Schaub, 2023a) and compared the results against TPCC, Spectral Clustering, Graph Spectral Clustering on the graph constructed by TPCC, and against $k$-means in Figure 7 *Left*. We have also analysed the robustness of TOPF against the addition of outliers in Figure 7 *Right*. We see that TOPF performs well in both cases, underlining our claim of robustness. Finally, in Figure 16 and Figure 15, we show that TOPF performs well under *sampling heterogeneity* and under *downsampling to sparse point clouds*, while comparing TOPF to the other baselines.

**Embedding Space of Variational Autoencoders** Variational autoencoders (VAE) are unsupervised neural networks that learn to extract a low-dimensional embedding of a high-dimensional data set. We have trained a VAE on $518$ $33 \times 33$ image patches which were sampled along a topological structure on a larger image with a latent space dimension of 3. We show that running TOPF on the latent space can recover the *topological structure* underlying the capturing process of the training set. We visualise the feature vectors on the latent space in Figure 4 (e) and Figure 6 and provide additional details in Figure 13 and Appendix F.

**High-dimensional Point Clouds** In Figure 5, we use TOPF on a high-dimensional input point cloud representing the conformation space of cyclooctane Martin & Watson (2011). We see in the ISOMAP projections of Figure 5 that TOPF extracts reasonable features representing the topology of the conformation space. In Figure 6, we apply TOPF to the original 8748-dimensional space and showcase the result on the centre of the corresponding sampled image patches. This shows the feasability of TOPF on very high-dimensional spaces.

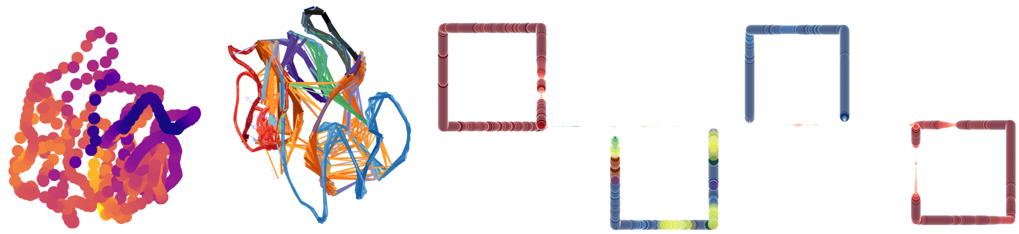

Figure 6: **VAE experiment described in Figure 13 for a** $16$**-dimensional latent space of the VAE and the original** $8748$**-dimensional image space.** *Left:* The first three of the $16$ dimension of the latent space embeddings, colour represents the fourth dimension. *Centre left:* The weighted harmonic representatives of the selected topological features. *Centre right:* The features produced by TOPF overlayed on the centre pixels of the associated image patches, as done in Figure 13 **4.**, just without the picture of the cow. *Right:* TOPF features obtained from the original $8748$-dimensional image space. This shows that TOPF can help analyse topological structure in high-dimensional data.

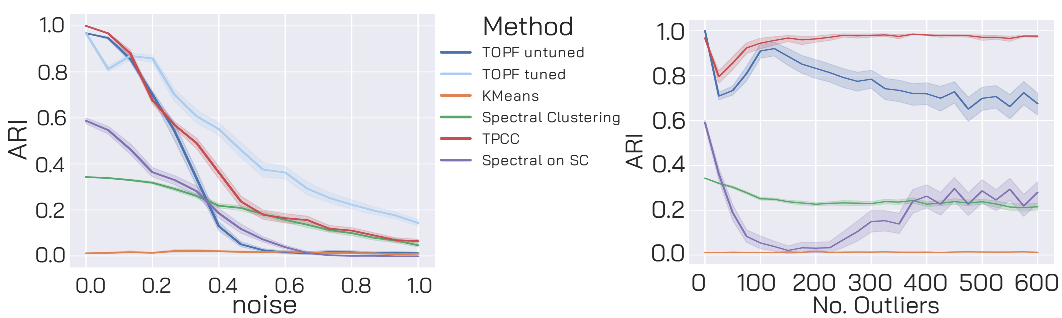


Figure 7: **Performance of TOPF Clustering with noise/outliers on `Spherein◯`, 95% CI.** The radii are $3$ and $1$, the mean distance to the closest neighbour is $0.22$. *Left:* We add i.i.d. Gaussian noise to every point with standard deviation indicated by the `noise` parameter. Even when compared with TPCC on a data set specifically crafted for TPCC, TOPF requires significantly less information and delivers almost equal performance. Tuned for datasets with a high noise, the TOPF outperform TPCC. *Right:* We add outliers with same standard deviation as the point cloud. We then measure ARI restricted to original points. Compared with TPCC on a data set specifically crafted for TPCC, TOPF requires significantly less information and delivers matching to superior performance.

## 5 DISCUSSION

**Limitations** TOPF can — by design — only produce meaningful output on point clouds with a *topological structure* quantifiable by persistent homology. In practice it is thus desirable to combine TOPF with some geometric or other point-level feature extractor. As TOPF relies on the computation of persistent homology, its runtime increases on very large point clouds, especially in higher dimensions where $\alpha$-filtrations are computationally infeasible. However, subsampling, either randomly or using landmarks, usually preserves relevant topological features while improving run time (Perea, 2020). Finally, selection of the relevant features is a very hard problem. While our proposed heuristics work well across a variety of domains and application scenarios, only domain- and problem-specific knowledge makes correct feature selection feasible.

**Future Work** The integration of higher-order TOPF features into ML pipelines that require point-level features potentially leads to many new interesting insights across the domains of biology, drug design, graph learning and computer vision. Furthermore, efficient computation of simplicial weights leading to the provably most faithful topological point features is an exciting open problem.

**Conclusion** We introduced point-level features TOPF founded on algebraic topology relating global structural features to local information. We gave theoretical guarantees for the correctness of their construction and evaluated them quantitatively and qualitatively on synthetic and real-world data sets. Finally, we introduced the novel topological clustering benchmark suite and showed that clustering using TOPF outperforms other available clustering methods and features extractors.

# REFERENCES


Naheed Anjum Arafat, Debabrota Basu, Yulia Gel, and Yuzhou Chen. When witnesses defend: A witness graph topological layer for adversarial graph learning. *arXiv preprint arXiv:2409.14161*, 2024.

Michael Atiyah. *K-theory*. CRC press, 1989.

Saugata Basu and Nathanael Cox. Harmonic persistent homology. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 1112–1123. IEEE, 2022.

Ulrich Bauer. Ripser: efficient computation of vietoris–rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 2021.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Katherine Benjamin, Lamisah Mukta, Gabriel Moryoussef, Christopher Uren, Heather A Harrington, Ulrike Tillmann, and Agnese Barbensi. Homology of homologous knotted proteins. *Journal of the Royal Society Interface*, 20(201):20220727, 2023.

A. K. Bousfield. The localization of spaces with respect to homology. *Topology*, 14(2):133–150, 1975. doi: https://doi.org/10.1016/0040-9383(75)90023-3.

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

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

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

Mathieu Carrière, Steve Y Oudot, and Maks Ovsjanikov. Stable topological signatures for points on 3d shapes. In *Computer graphics forum*, volume 34, pp. 1–12. Wiley Online Library, 2015.

Mathieu Carriere, Frédéric Chazal, Marc Glisse, Yuichi Ike, Hariprasad Kannan, and Yuhei Umeda. Optimizing persistent homology based functions. In *International conference on machine learning*, pp. 1294–1303. PMLR, 2021.

Mathieu Carriere, Marc Theveneau, and Théo Lacombe. Diffeomorphic interpolation for efficient persistence-based topological optimization. *arXiv preprint arXiv:2405.18820*, 2024.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Charu Chaudhry, Arthur L Horwich, Axel T Brunger, and Paul D Adams. Exploring the structural dynamics of the e. coli chaperonin groel using translation-libration-screw crystallographic refinement of intermediate states. *Journal of molecular biology*, 342(1):229–245, 2004.

Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in artificial intelligence*, 4:108, 2021.

Frédéric Chazal, Leonidas J. Guibas, Steve Y. Oudot, and Primoz Skraba. Persistence-based clustering in riemannian manifolds. *J. ACM*, 60(6), nov 2013.

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

Yu-Chia Chen, Marina Meilă, and Ioannis G Kevrekidis. Helmholtzian eigenmap: Topological feature discovery & edge flow learning from point cloud data. *arXiv preprint arXiv:2103.07626*, 2021.

Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

Matija Čufar and Žiga Virk. Fast computation of persistent homology representatives with involuted persistent homology. *Foundations of Data Science*, 5(4):466–479, 2023.

Vin De Silva and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pp. 227–236, 2009.

Vin De Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co) homology. *Inverse Problems*, 27(12):124003, 2011.

Richard Dedekind. *Was sind und was sollen die Zahlen?* Verlag Friedrich Vieweg und Sohn, Braunschweig, 1888.

Boris Delaunay et al. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934.

Jozef Dodziuk. Finite-difference approach to the hodge theory of harmonic forms. *American Journal of Mathematics*, 98(1):79–104, 1976.

Stefania Ebli and Gard Spreemann. 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.

Herbert Edelsbrunner. Alpha shapes-a survey. In *Tessellations in the sciences: Virtues, techniques and applications of geometric tilings*. 2011.

Herbert Edelsbrunner and John Harer. Persistent homology—a survey. *Contemporary mathematics*, 453(26):257–282, 2008.

Samuel Eilenberg and Saunders MacLane. General theory of natural equivalences. *Transactions of the American Mathematical Society*, 58:231–294, 1945.

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

David Chin-Lung Fong and Michael Saunders. Lsmr: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, 33(5):2950–2971, 2011.

Vincent P. Grande and Michael T. Schaub. Topological point cloud clustering. In *Proceedings of the 40th International Coference on Machine Learning*, ICML'23, 2023a.

Vincent P. Grande and Michael T Schaub. Non-isotropic persistent homology: Leveraging the metric dependency of ph. In *Learning on Graphs Conference*, pp. 17–1. PMLR, 2023b.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.

Davide Gurnari, Aldo Guzmán-Sáenz, Filippo Utro, Aritra Bose, Saugata Basu, and Laxmi Parida. Probing omics data via harmonic persistent homology. *arXiv preprint arXiv:2311.06357*, 2023.

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

Felix Hausdorff. Grundzüge einer theorie der geordneten mengen. *Mathematische Annalen*, 65: 435–505, 1908.

Esther Hidber, Edward R Brownie, Koto Hayakawa, and Marie E Fraser. Participation of cys123$\alpha$ of escherichia coli succinyl-coa synthetase in catalysis. *Acta Crystallographica Section D: Biological Crystallography*, 63(8):876–884, 2007.

David Hilbert. *Grundlagen der Geometrie*. Wissenschaft und Hypothese. B. G. Teubner, Leipzig, 1899.

Yasuaki Hiraoka, Yusuke Imoto, Théo Lacombe, Killian Meehan, and Toshiaki Yachimura. Topological node2vec: Enhanced graph embedding via persistent homology. *Journal of Machine Learning Research*, 25(134):1–26, 2024.

Sze-tsen Hu. *Homotopy theory*. Academic press, 1959.

Michael Kerber and Herbert Edelsbrunner. 3d kinetic alpha complexes and their implementation. In *2013 Proceedings of the Fifteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pp. 70–77. SIAM, 2013.

Marc Kschonsak, Han Chow Chua, Claudia Weidling, Nourdine Chakouri, Cameron L. Noland, Katharina Schott, Timothy Chang, Christine Tam, Nidhi Patel, Christopher P. Arthur, Alexander Leitner, Manu Ben-Johny, Claudio Ciferri, Stephan Alexander Pless, and Jian Payandeh. Structural architecture of the human nalcn channelosome. *Nature*, 603(7899):180–186, Mar 2022. doi: 10.1038/s41586-021-04313-5.

Peter Lawson, Andrew B Sholl, J Quincy Brown, Brittany Terese Fasy, and Carola Wenk. Persistent homology for the quantitative evaluation of architectural features in prostate cancer histology. *Scientific reports*, 9(1):1139, 2019.

Lei Li, Hongbo Fu, and Maks Ovsjanikov. WSDesc: Weakly supervised 3d local descriptor learning for point cloud registration. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2022. doi: 10.1109/TVCG.2022.3160005.

Lek-Heng Lim. Hodge laplacians on graphs. *SIAM Review*, 62(3):685–715, 2020.

Jacob Lurie. Stable infinity categories. *arXiv preprint math/0608228*, 2006.

Yunqian Ma and Yun Fu. *Manifold learning theory and applications*, volume 434. CRC press Boca Raton, 2012.

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

Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11:417–487, 2011.

Facundo Mémoli, Zhengchao Wan, and Yusu Wang. Persistent laplacians: Properties, algorithms and implications. *SIAM Journal on Mathematics of Data Science*, 4(2):858–884, 2022.

Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *International conference on machine learning*, pp. 7045–7054. PMLR, 2020.

Elizabeth Munch. A user's guide to topological data analysis. *Journal of Learning Analytics*, 4(2): 47–61, 2017.

Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL `probml.ai`.

Haruhisa Oda, Mayuko Kida, Yoichi Nakata, and Hiroki Kurihara. Novel definition and quantitative analysis of branch structure with topological data analysis. *arXiv preprint arXiv:2402.07436*, 2024.

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

Jose A. Perea, Luis Scoccola, and Christopher J. Tralie. Dreimac: Dimensionality reduction with eilenberg-maclane coordinates. *Journal of Open Source Software*, 8(91):5791, 2023. doi: 10.21105/joss.05791. URL `https://doi.org/10.21105/joss.05791`.

Henri Poincaré. Analysis situs. *J. de l'Ecole Poly.*, 1, 1895.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

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

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

Michael T Schaub, Austin R Benson, Paul Horn, Gabor Lippner, and Ali Jadbabaie. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Review*, 62(2):353–391, 2020.

Michael T. Schaub, Yu Zhu, Jean-Baptiste Seby, T. Mitchell Roddenberry, and Santiago Segarra. 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.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

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

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

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

Larry Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, 5(1): 501–532, 2018.

JunJie Wee, Jiahui Chen, Kelin Xia, and Guo-Wei Wei. Integration of persistent laplacian and pre-trained transformer for protein solubility changes upon mutation. *Computers in Biology and Medicine*, pp. 107918, 2024.

Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017.

Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

Matija Čufar. Ripserer.jl: flexible and efficient persistent homology computation in julia. *Journal of Open Source Software*, 5(54):2614, 2020. doi: 10.21105/joss.02614. URL `https://doi.org/10.21105/joss.02614`.

## A Extended Background

**A brief history of topology and machine learning** Algebraic topology is a discipline of Mathematics dating back roughly to the late 19th century (Poincaré, 1895). Starting with Henri Poincaré and continuing in the early 20th century, the mathematical community became interested in developing a framework to capture the global shapes of manifolds and topological spaces in concise algebraic terms. This development was partly made possible by the push towards a formalisation of mathematics and analysis, in particular, which took place inside the mathematical community in the 1800's and early 1900's (e.g. Dedekind (1888); Hilbert (1899); Hausdorff (1908)). The axiomatisation of analysis in the early 20th century is an important result of this process. These abstract ideas made it possible for Topologists to talk about the now common notions of Euler characteristics, Betti number, simplicial homology of manifolds, topological spaces, and simplicial and CW complexes. Over the course of the last 100 years, branching into many sub-areas like low-dimensional topology, differential topology, K-theory or homotopy theory (Atiyah, 1989; Hu, 1959), algebraic topology has resolved many of the important questions and provides a comprehensive tool-box for the study of topological spaces. These achievements were tied to an abstraction and generalisation of concepts: topological spaces turned into spectra, diffeomorphism to homotopy equvialences and later weak equivalences, and

Topologists turned to category theory (Eilenberg & MacLane, 1945), model categories (Bousfield, 1975) and recently $\infty$-categories (Lurie, 2006) as the language of choice.

The 21st century saw the advent and rise of topological data analysis (TDA, (Bubenik et al., 2015; Chazal & Michel, 2021)). In short, mathematicians realised that the same notions of shape and topology that their predecessors carefully defined a century earlier were now characterising the difference between healthy and unhealthy tissue, between normal and abnormal behaviour protein behaviour, or more general between different categories in their complex data sets.

**Related Work** The intersection of topological data analysis, topological signal processing and geometry processing has many interesting related developments in the past few years. On the side of homology and TDA, the authors in De Silva & Vejdemo-Johansson (2009) and Perea (2020) use harmonic *co*homology representatives to reparametrise point clouds based on circular coordinates. This implicitly assumes that the underlying structure of the point cloud is amenable to such a characterization. Although circular coordinates are orthogonal to the core goal of TOPF, the approaches share many key ideas and insights. In Basu & Cox (2022); Gurnari et al. (2023), the authors develop and use harmonic persistent homology and provide a way to pool features to the point-level. However, their focus is not on providing robust topological point features and their approach includes no tunable homology feature selection across dimensions, no support for weighted simplicial complexes, and they only construct the simplicial complex at birth. In their paper on topological mode analysis, Chazal et al. (2013) use persistent homology to cluster point clouds. However, they only consider 0-dimensional homology to base the clustering on densities and there is no clear way to generalise this to higher dimensions.

On the more geometric-centred side, Ebli & Spreemann (2019) already provide a notion of harmonic clustering on simplices, Chen & Meilă (2021); Chen et al. (2021) analyse the notion of geometry and topology encoded in the Hodge Laplacian and its relation to homology decompositions, Schaub et al. (2020) study the normalised and weighted Hodge Laplacian in the context of random walks, and Grande & Schaub (2023a) use the harmonic space of the Hodge Laplacians to cluster point clouds respecting topology. Finally, a persistent variant of the Hodge Laplacian is used to study filtrations of simplicial complexes (Mémoli et al., 2022).

There are other constructions of isometry-invariant local shape descriptors, as for example proposed in Mémoli (2011). However, their aim is not relate the global topology back to the local features.

In Moor et al. (2020), the authors basically set up a topological loss functions that ensures that topological features of the original point cloud are preserved in the latent space of the autoencoder. This is different from the approach TOPF takes, as TOPF tries to detect and extract topological features. Similar things can be said about topological node-2-vec Hiraoka et al. (2024) and some experiments in Carriere et al. (2021) or some experiments of Carriere et al. (2024): They all try to come up with ways to preserve global topological structure while embedding a point cloud, while we on the other hand construct the embedding based on the relationship between the points and the global topological structure.

On graphs, Arafat et al. (2024) construct topological representations of graphs to study adversarial graph learning. They construct a local topological features for nodes and global topological features for the entire graph. In contrast to our approach, they do not relate the global topological features back to the local level, but rather consider the local topology on the point level and the single global topological summary separately. Furthermore when constructing their features, they use vision transformers on the persistence images of their witness filtrations, which does not allow for the same amount of interpretability as TOPFdoes.

In Grande & Schaub (2023a), the authors have introduced TPCC, the first method to cluster a point cloud based on the higher-order topological features encoded in the data set. However, TPCC is **(i)** computationally expensive due to extensive eigenvector computations, **(ii)** depending on high-dimensional subspace clustering algorithms, which are prone to instabilities and errors, **(iii)** sensitive to the correct choice of hyperparameters, **(iv)** requiring the topological true features and noise to occur in different steps of the simplicial filtration, and it **(v)** solely focussed on clustering the points rather than extracting relevant node-level features. This paper solves all the above by completely revamping the TPCC pipeline, introducing several new ideas from applied algebraic topology and differential geometry. The core insight is: When you have the time to compute persistent homology with generators on a data set, you get the topological node features with similar computational effort.

# B THEORETICAL CONSIDERATIONS

## B.1 MORE DETAILS ON VR AND $\alpha$-FILTRATIONS

Vietoris–Rips complexes are easy to define, approximate the topological properties of a point cloud across all scales and computationally easy to implement. However for moderately large $r$, the associated VR complex contains a large number of simplices — up to $\binom{|X|}{n+1}$ $n$-simplices for large enough $r$ — leading to poor computational performance for any downstream task on some large point clouds. One way to see this is the following: After adding the first edge that connects two components or the final simplex that fills a hole in the simplicial complex the VR complex keeps adding more and more simplices in the same area that keep the topology unchanged. One way to mitigate this problem is to pre-compute a set of simplices that are able to express the entire topology of the point cloud. For a point cloud $X \subset \mathbb{R}^n$, the $\alpha$-filtration consists of the intersection of the simplicial complexes of the VR filtration on $X$ with the (higher-dimensional) Delaunay triangulation of $X$ in $\mathbb{R}$. Due to algorithmic reasons, the filtration value of a simplex is then related to the radius of the circumscribed sphere instead of the maximum pair-wise distance of vertices, where the filtration value $\alpha^X(\sigma)$ of a simplex $\sigma$ is given by the minimum $\varepsilon$ such that $\sigma$ is contained in $\alpha_\varepsilon(X)$, i.e. $\alpha^X(\sigma) := \inf\{\varepsilon \in \mathbb{R} : \sigma \in \alpha_\varepsilon(V)\}$. This reduces the number of required simplices across all dimensions to $O(|X|^{\lceil n/2 \rceil})$. However, the Delaunay triangulation becomes computationally infeasible for larger $n$.

**Definition B.1** ($n$-dimensional Delaunay triangulation)**.** Given a set of vertices $V \subset \mathbb{R}^n$, a Delaunay triangulation $DT(V)$ is a triangulation of $V$ such that for any $n$-simplex $\sigma_n \in DT(V)$ the interior of the circum-hypersphere of $\sigma_n$ contains no point of $DT(V)$. A triangulation of $V$ is a SC $\mathcal{S}$ with vertex set $V$ such that its geometric realisation covers the convex hull of $V$ $\mathrm{hull}(V) = |\mathcal{S}|$ and we have for any two simplices $\sigma$, $\sigma'$ that the intersection of geometric realisations $|\sigma| \cap |\sigma'|$ is either empty or the geometric realisation $|\hat{\sigma}|$ of a common sub-simplex $\hat{\sigma} \subset \sigma, \sigma'$.

If $V$ is in general position, the Delaunay triangulation is unique and guaranteed to exist (Delaunay et al., 1934). We will now first introduce a slightly simpler version of the $\alpha$-filtration, the $\alpha^*$-filtration.

**Definition B.2** ($\alpha^*$-complex of a point cloud)**.** Given a finite point cloud $X$ in real space $\mathbb{R}^n$, the $\alpha^*$-complex $\alpha^*_\varepsilon(X)$ is the subset of the $n$-dimensional Delaunay triangulation $DT(X)$ consisting of all $k$-simplices $\sigma_k \in DT(X)$ with a radius $r$ of its circumscribed $(k-1)$-sphere with $r \leq \varepsilon$ for all $k \leq n$.

The $\alpha^*$ and $\alpha$-filtration share the same set of simplices and agree on the filtration values of the simplices on all top-dimensional simplices and all other simplices which are *Gabriel* simplices.

**Definition B.3** (Gabriel Simplex)**.** Given a set of vertices $V \subset \mathbb{R}^n$ and its Delaunay triangulation $DT(V)$. A simplex $\sigma \in DT(V)$ is called *Gabriel* if there exists no point $v \in V$ such that $v$ is contained in the interior of the circumsphere of $\sigma$.

**Definition B.4** ($\alpha$-complex of a point cloud, (Kerber & Edelsbrunner, 2013; Edelsbrunner, 2011))**.** Given a finite point cloud $X$ in real space $\mathbb{R}^n$, the $\alpha$-complex $\alpha_\varepsilon(X)$ is the $\alpha^*$-complex of $V$, except that the filtration value $\alpha(\sigma)$ of all non-Gabriel simplices $\sigma$ with associated points $X$ in the interior of the circumsphere of $\sigma$ is given by $\min_{x \in X} \alpha(\sigma \cup \{x\})$.

We note that due to the definition of the Delaunay triangulation, all $n$-simplices are Gabriel simplices and hence this is well-defined. This is an equivalent formulation of the original definition, which was for example shown in Kerber & Edelsbrunner (2013). We chose to go with the above formulation, as it is the form used in implementations of $\alpha$-filtrations.

We note two things: *(i)* The $\alpha^*$-filtration and $\alpha$-filtration have fewer simplices than the VR filtration and *(ii)* the filtration values of individual simplices differ between the $\alpha^*$, the $\alpha$ and the VR filtration. In particular, the order of the filtration values can be different across the two types of filtration. For the 1-skeleton, the $\alpha^*$ and VR filtrations are equivalent: The VR filtration value of an edge is its length $l$, whereas its $\alpha^*$-filtration value is the radius $r = l/2$ of the associated 0-sphere consisting only of the two vertices.

### B.2 DIFFERENTIAL FORMS AND A CONTINUOUS ANALOGUE OF TOPF

In this section, we will discuss how a continuous analogue of TOPF on Riemannian manifolds would look like. Simply considering the simplicial homology of the manifold, as we did in the case of a discrete simplicial complex, is however not sufficient: Harmonic cycles and cocycles don't exist in the simplicial chain complex of the manifold. This is because the space of chains in simplicial homology on manifolds is infinite-dimensional. Thus, we cannot canonically identify the space of chains with its dual, the space of cochains. From an intuitive point of view, a harmonic chain would need to take infinitesimal small values on every simplex which is not allowed. Rather, the right language to formalise harmonic representatives is *Differential Geometry, Hodge Theory, and harmonic forms*. Dodziuk proved in Dodziuk (1976) that the discrete Hodge Laplacian as used in this paper converges to the Hodge Laplacian on differential forms which are arbitrary-dimensional integrands over manifolds.

There is a beautiful connection between differential geometry and algebraic topology: A theorem by de Rham states that given a Riemannian manifold $M$, the real-valued homology $H_k(M;\mathbb{R})$ with coefficients in $\mathbb{R}$ is isomorphic to the de Rham cohomology $H^k_{\text{dR}}(M;\mathbb{R})$ on differential forms on $M$ for arbitrary $k \in \mathbb{Z}_{\geq 0}$, some form of cohomology defined via the chain complex of vector spaces of differential $k$-forms on $M$. We can define an analogue of the discrete Hodge Laplacian on differential forms by the differential induced via exterior derivates $d$ and its adjoint $\sigma$:

$$\Delta \coloneqq \delta d + d\delta$$

The kernel $\ker \Delta$ of the Hodge Laplacian is called the space of harmonic forms. Similar to the discrete case, the Hodge theorem now gives us a natural vector space isomorphism between the space of harmonic $k$-forms on $M$, $\mathcal{H}_k(M)$, and the $k^{\text{th}}$ real-valued homology group

$$H_k(M;R) \cong \mathcal{H}_k(M).$$

We can rephrase this as follows: the harmonic forms can be seen as unique and natural homology representatives.

Let us now get some intuition for what this means for continuous TOPF features: In dimension $0$, 0-forms are functions $f\colon M \to R$. In this case, we can simply take their value $f(x)$ at $x \in M$ as the corresponding point feature at $x$. In dimension 1, there is a correspondence between 1-forms and vector fields on $M$. In this case, we can take the norm of the vector field that corresponds to a given harmonic form at a point $x$ via the map $x \mapsto |v(x)|$.

In general in dimension $k$, this is more complicated. Luckily, as we are interested in point-features, we can do all computations point-wise. We consider a point $x \in M$ and a harmonic form $\omega$. At point $x$, $\omega$ determines an element in the exterior algebra on the dual of the tangent space of $M$ at $x$, i.e $\omega_x \in \bigwedge^k T^*_x M$ and we need to define a norm on this space. An orthonormal basis on $T^*_x M$, $e_1(x), \ldots, e_n(x)$ gives rise to a basis consisting of elements $e_{i_1}(x) \wedge \ldots \wedge e_{i_k}(x)$ of the exterior algebra. We can now evaluate $\omega_x$ on the elements of this basis $\omega_x(e_{i,1}(x) \wedge \ldots \wedge e_{i,k}(x))$ and take the square root of the sum of squares of the individual results

$$\texttt{TOPF}_\omega\colon x \mapsto \sqrt{\sum_{1 \leq i_1 < \cdots < i_k \leq n} \omega_x\left(e_{i_1}(x) \wedge \cdots \wedge e_{i_k}(x)\right)^2}.$$

As one can show, this result is independent of the choice of ONB of $T^*_x M$ and gives a point-wise norm of the differential forms.

In the discrete case discussed in this paper, however, we don't evaluate on all basis elements as described above, but rather on all $k$-simplices adjacent to $x$. These $k$-simplices can thus be seen as samples from the exterior algebra of $x$. This sampling induces some differences to the continuous case, which we address with the post-processing procedure as described in the paper.

### B.3 FIXING REAL LIFTS OF HOMOLOGY GENERATORS IN $\mathbb{Z}/3\mathbb{Z}$-COEFFICIENTS

In step **3.** of the TOPF algorithm in Section 3, we attempt to lift a homology generator with simplices $\sigma_j \in \Sigma_k$ and coefficients $c_j$ in $\mathbb{Z}/3\mathbb{Z}$-coefficients to a homology generator in $\mathbb{R}$-coefficients using the

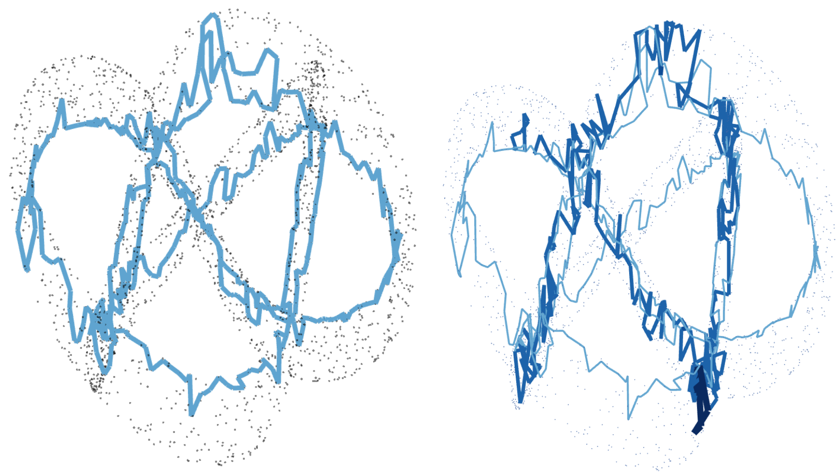

Figure 8: $\mathbb{Z}/3\mathbb{Z}$**-homology generator with "faulty" lift to** $\mathbb{R}$**-coefficients.** *Left:* Initial lift produced in $\mathbb{Z}/3\mathbb{Z}$-coefficients. *Right:* Fixed lift with *light blue* edges representing the original lift with coefficients in $\{-1, 1\}$ and *dark blue* edges representing the fix with coefficients in $3\mathbb{Z}$. The underlying data is the trefoil knot embedded in $\mathbb{R}^3$, as generated by Perea et al. (2023).

embedding

$$e_k^i(\sigma) := \begin{cases} \iota(c_j) & \exists \hat{\sigma}_j \in \Sigma_k^i : \sigma = \hat{\sigma}_j \\ 0 & \text{else} \end{cases}.$$

where $\iota\colon \mathbb{Z}/3\mathbb{Z}$ is the map induced by the canonical inclusion of $\{-1, 0, 1\} \hookrightarrow \mathbb{R}$. Thus, we know that $e_k^i$ is a cycle modulo 3: $\mathcal{B}_{k-1} e_k^i \equiv 0 \mod 3$. A-prior however, there is no reason why this equality should hold in $\mathbb{R}$ as well. As it turns out, in virtually all examples in practice, we already have $e_k^i \in \ker \mathcal{B}_{k-1}$ which is very convenient. The same phenomenon occurs for cocycles as well, see De Silva & Vejdemo-Johansson (2009). We will now assume to be in a rare case of $e_k^i \notin \ker \mathcal{B}_{k-1}$. We can now attempt to fix this using a linear integer program finding $y \in \mathbb{Z}^{|\mathcal{S}_k|}$ such that

$$3\mathcal{B}_{k-1} y = \mathcal{B}_{k-1} e_k^i.$$

And then we set $\text{fix}(e_k^i) := e_k^i - 3y$. If this linear program is not feasible, $e_k^i$ has no integer/real lift and TOPF will skip the associated topological feature. In Figure 8, we give a visual example of a faulty lift and a fix obtained by the above procedure.

## C Theoretical Guarantees

In this section, we want to investigate possible theoretical guarantees for TOPF computed on idealised datasets. Instead of directly working with the $\alpha$-complexes, we will first prove guarantees for the closely related construction of $\alpha^*$-complexes. $\alpha^*$ filtrations (Definition B.2) are closely related to $\alpha$-filtrations (Definition B.4) used in the implementation and agree on the filtration values on most of the simplices, have however nicer properties. Finally, we give conditions on when this result applies to the $\alpha$-filtrations, which is relevant to the actual implementation of TOPF.

**Theorem C.1** (Topological Point Features of Spheres)**.** *Let $X$ consist of at least $(n+2)$ points (denoted by $S$) sampled uniformly at random from a unit $n$-sphere in $\mathbb{R}^{n+1}$ and an arbitrary number of points with distance of at least 2 to $S$. When we now consider the $\alpha^*$-filtration on this point cloud, with probability 1 we have that* ***(i)*** *there exists an $n$-th persistent homology class generated by the $n$-simplices on the convex hull of $S$, and* ***(ii)*** *the support of the associated topological point feature (TOPF) $\mathcal{F}_n^*$ is precisely $S$: $\text{supp}(\mathcal{F}_n^*) = S$.* ***(iii)*** *The same holds true for point clouds sampled from multiple $n_i$-spheres if the above conditions are met on each individual sphere.* ***(iv)*** *If there are no other points contained in or on the circumspheres of the $n$-simplices of the convex hull of $S$, then the same above holds true for TOPF computed using the $\alpha$-filtration.* ***(v)*** *More generally, for any contractible subset $M$ of the $(n+1)$-simplices such that all $n$-simplices on the boundary fulfill the* *above condition, this holds when using the $\alpha$-filtration, the discussed $n$-simplices (i) and the subset of points contained in $M$ (ii).*

We will give a proof of this after the following remark.

*Remark* C.2. The key idea of the proof is to use that $\alpha^*$-filtrations and partially $\alpha$-filtrations assign the filtration value based on the radius of the circumsphere of the $k$-simplex. Because all points in $S$ lie on the same $n$-sphere, with probability 1 we can write down the filtration value of the $(n+1)$-simplices explicitly. This is of course a very idealised setting. However in practice, datasets with topological structure consist in a majority of cases of points sampled with noise from deformed $n$-spheres. The theorem thus guarantees that TOPF will recover these structural information in an idealised setting. Experimental evidence suggests that this holds under the addition of noise as well which is plausible as harmonic persistent homology is robust against some noise (Basu & Cox, 2022). For the $2D$ setting, condition **(iv)** is equivalent to assuming that there is no line through the centre of the circle such that one half plane does not contain any points of $S$.

We will now give the proof of the theorem that guarantees that TOPF works. We discuss the assumptions of the theorem in Remark C.3.

*Proof.* Assume that we are in the scenario of the theorem. Now because the $n$-volume of $(n-1)$-submanifolds is zero, we have that with probability 1 the points of $S$ don't lie on a single $(n-1)$ sphere inside the $n$-sphere. Let us now look at the $\alpha$-filtration of the simplices in $S$: Recall that the filtration values of a $k$-simplex is given by the radius of the $(k-1)$-sphere determined by its vertices. Because all of the $(n+1)$-simplices $\sigma_{n+1}$ with vertices $V \subset S$ in $S$ lie on the same unit $n$-sphere $S_n$, they all share the filtration value of $\alpha(\sigma_{n+1}) = 1$. By the same argument as above, with probability 1 there are no $(n+1)$ points in $S$ that lie on an *unit* $(n-1)$-sphere. Thus all of the $n$-simplices $\sigma_n$ lie on $(n-1)$-spheres $S_n$ with a radius $r < 1$ smaller than 1 and hence have a filtration value $\alpha(\sigma_n)$ smaller than 1. Let

$$b \coloneqq \max\left(\{\alpha(\sigma_n) : \sigma_n \subset \partial\, \mathrm{hull}(S)\}\right)$$

be the maximum filtration value of an $n$-simplex on the boundary of the convex hull of $S$. Then, a linear combination $g$ of the $n$-simplices of the boundary of the convex hull of $S$ with coefficients in $\pm 1$ is a generator of a persistent homology class with life time $(b, 1)$ (this follows from the fact that $n$-spheres and their triangulations are orientable). This proves claim **(i)**.

Because of the assumption that all points not contained in $S$ have a distance of at least 2 to the points in $S$, all $(n+1)$-simplices $\sigma_{n+1}$ with vertices both in $S$ and its complement in $X$ will have a filtration value $\alpha(\sigma_{n+1}) \geq 1$ of at least 1. Recall that all $(n+1)$-simplices $\sigma_{n+1} \subset S$ with vertices inside $S$ have a filtration value of $\alpha(\sigma_{n+1}) = 1$. Thus the adjoint of the $n$-th boundary operator $\mathcal{B}_n^\top$ is trivial on the homology generator $g$ for a simplicial complex constructed during $(b, 1)$. Thus, we have that for the $n$-th Hodge Laplacian

$$L_n g = \mathcal{B}_{n-1}^\top \mathcal{B}_{n-1} g + \mathcal{B}_n \mathcal{B}_n^\top g = 0 + 0 = 0$$

and hence $g$ is a harmonic generator for the entire filtration range of $(b, 1)$. Claim **(ii)** and **(iii)** then follow from the construction of the TOPF values.

For claim **(iv)**, it suffices to notice that the $n$-simplices $\sigma$ on the convex hull are precisely non-Gabriel simplices iff there exist a point $x \in S$ lying inside the circumsphere of $\sigma$. Because the assumption of part **(iv)** excludes this case, all $n$-simplices are Gabriel simplices and the $\alpha$-filtration values (Definition B.4) and $\alpha^*$-filtration values (Definition B.2) agree on the $n$-simplices and $(n+1)$-simplices of the Delaunay triangulation. Because these are the only considered filtration values for the proof, the claim follows.

Finally for part **(v)**, we see that the same argument now applies to the simplices on the boundary, consisting of $n$-simplices of $M$, consisting of $(n+1)$-simplices. Because $M$ is contractible, it is connected and has trivial homology. Thus, its boundary is homotopy equivalent to an $n$-sphere. Because in this case only the points $X'$ on the boundary of $M$ are contained in the generating $n$-simplices of the considered homology class, the associated TOPF feature will have support on $X'$. This concludes the proof. □

*Remark* C.3. The setting of the theorem is a very special setting. In general on data, harmonic chains have entries with absolute values somewhere in $[0, 1]$. In the setting of the theorem however, the

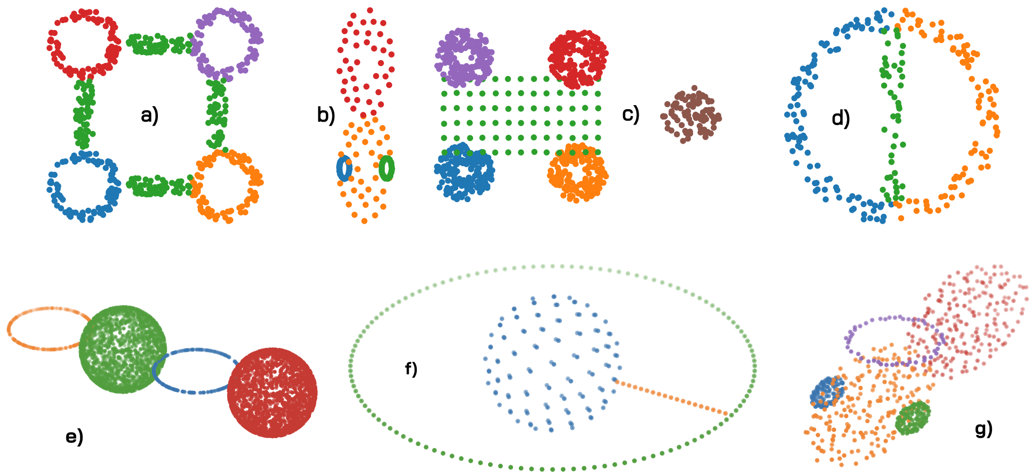


Figure 9: **Data sets of the Topological Clustering Benchmark Suite (TCBS) with true labels.** *Top:* 2*D* data sets. *From left to right: a):* `4Spheres` (656 points), *b):* `Ellipses` (158 points), *c):* `Spheres+Grid` (866 points), *d):* `Halved Circle` (249 points). *Bottom:* 3*D* data sets. *From left to right: e):* `2Spheres2Circles` (4600 points), *f):* `SphereinCircle` (267 points), *g):* `spaceship` (650 points).

harmonic representative will take values in $-1, 0, 1$. We will briefly explain why this is the case: Consider a $k$-homology representative $r$ with simplex-values in $\{-1, 0, 1\}$. We consider the harmonic projection $h$ of $r$ into the harmonic space. Then, for a $k$-simplex $\sigma$, we have that $h(\sigma) = \pm 1$ iff $r(\sigma) = 1$ **and** $\sigma$ is not the face of any $(k+1)$-simplices.

This can for example be seen by representing $h$ as the difference of $r$ and its gradient and curl parts $h = r - r_{\text{grad}} - r_{\text{curl}}$. Because $r$ is already a cycle, $\mathcal{B}_k r = 0$ and thus $r_{\text{grad}} = 0$. Now, the curl part can be written as stemming from a signal on the the $k+1$-simplices, $r_{\text{curl}} = B_{k+1} x_{\text{curl}}$. However, because $\sigma$ is not the face of a $k+1$-simplex, $r_{\text{curl}}(\sigma) = 0$ and thus $h(\sigma) = r(\sigma)$.

The setting of the theorem precisely constructs a case where we have $k$-simplices and no $(k+1)$-simplices in the relevant part of the filtration. This is the case because in an $\alpha$-filtration, the filtration value of a $k$-simplex is the radius of its circumscribed $(k-1)$-sphere. All relevant $(k+1)$-simplices in the theorem lie on the $k$-unit sphere, and thus have filtration value 1. This is however an idealised setting. Because we construct the simplicial complex to compute the harmonic representative somewhere in the middle (determined by the interpolation coefficient) between the birth and death time of the homology class, empirically the majority of the $k$- simplices of the $k$-homology generators are faces of $(k+1)$-simplices. In this case, the harmonic representative smooths out the feature.

## D TOPOLOGICAL CLUSTERING BENCHMARK SUITE

We introduce seven point clouds for topological point cloud clustering in the topological clustering benchmark suite (TCBS). Some of these point clouds have been adapted from Grande & Schaub (2023a). The ground truth and the point clouds are depicted in Figure 9. The point clouds represent a mix between 0-, 1- and 2-dimensional topological structures in noiseless and noisy settings in ambient 2-dimensional and 3-dimensional space. The results of clustering according to TOPF can be found in Figure 10.

We constructed the benchmark by sampling from topologically different shapes with varying sampling density in different ambient dimensions. For example for the point cloud `2Spheres2Circles`, we combined points sampled from two spheres and two circles. We then divided the point cloud into four ground truth clusters, one for each of the two spheres and two circles and assigned every point the cluster corresponding to the object it was sampled from. Thus, the ground truth labels of a point corresponds to the topological structure it was sampled from.

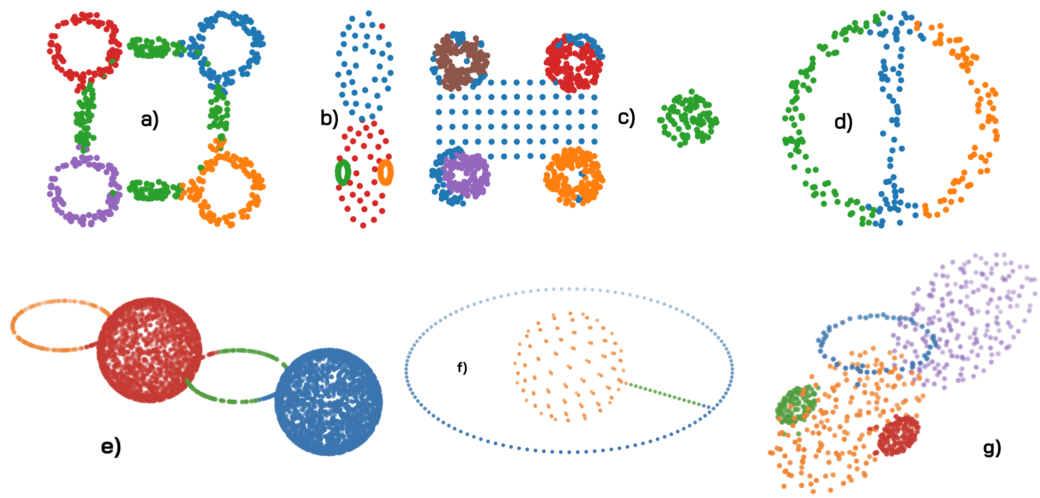


Figure 10: **Data sets of the Topological Clustering Benchmark Suite (TCBS) with labels generated by TOPF.** *Top:* $2D$ data sets. *From left to right: a):* `4Spheres` (0.81 ARI), *b):* `Ellipses` (0.95 ARI), *c):* `Spheres+Grid` (0.70 ARI), *d):* `Halved Circle` (0.71 ARI). *Bottom:* $3D$ data sets. *From left to right: e):* `2Spheres2Circles` (0.94 ARI), *f):* `SphereinCircle` (0.97 ARI), *g):* `spaceship` (0.92 ARI).

# E Implementation

We have created an easy-to-use python package TOPF which can be found in the supplementary material. The package currently works under macOS and Linux. The package contains both the code to generate the topological point features, as well as the code to reproduce the various visualisation steps in this paper.

All experiments were run on a Apple M1 Pro chipset with 10 cores and 32 GB memory. TOPF and the experiments are implemented in Python and Julia. For persistent homology computations, we used GUDHI (The GUDHI Project, 2015) (© The GUDHI developers, MIT license) and Ripserer (Čufar, 2020) (© mtsch, MIT license), which is a modified Julia implementation of (Bauer, 2021). For the least square problems, we used the LSMR implementation of SciPy (Fong & Saunders, 2011). We used the Node2Vec python implementation `https://github.com/eliorc/node2vec` (© Elior Cohen, MIT License) based on the Node2Vec Paper (Grover & Leskovec, 2016). We used the `pgeof` Python package for computation of geometric features `https://github.com/drprojects/point_geometric_features` (© Damien Robert, Loic Landrieu, Romain Janvier, MIT license). We use parts of the implementation of TPCC `https://git.rwth-aachen.de/netsci/publication-2023-topological-point-cloud-clustering` (© Computational Network Science Group, RWTH Aachen University, MIT license). We use the implementation of WSDesc Li et al. (2022), `https://github.com/craigleili/WSDesc` (Lei Li, Hongbo Fu, Maks Ovsjanikov. CC BY-NC 4.0 License). The idea for the implementation of the fix of the lift of the $\mathbb{Z}/3\mathbb{Z}$-representative to $\mathbb{R}$-coefficients was inspired by DREiMac, Perea et al. (2023).

## E.1 Hyperparameters

All the relevant hyperparameters are already mentioned in their respective sections. However, for convenience we gather and briefly discuss them in this section. We note that TOPF is robust and applicable in most scenarios when using the default parameters without tuning hyperparameters. The hyperparameters should more be thought of as an additional way where detailed domain-knowledge can enter the TOPF pipeline. We have performed additional experiments to validate the robustness of TOPF to moderate changes in parameters, see Figure 11.

**Maximum Homology Dimension** $d$ The maximum homology dimension determines the dimensions of persistent homology the algorithm computes.

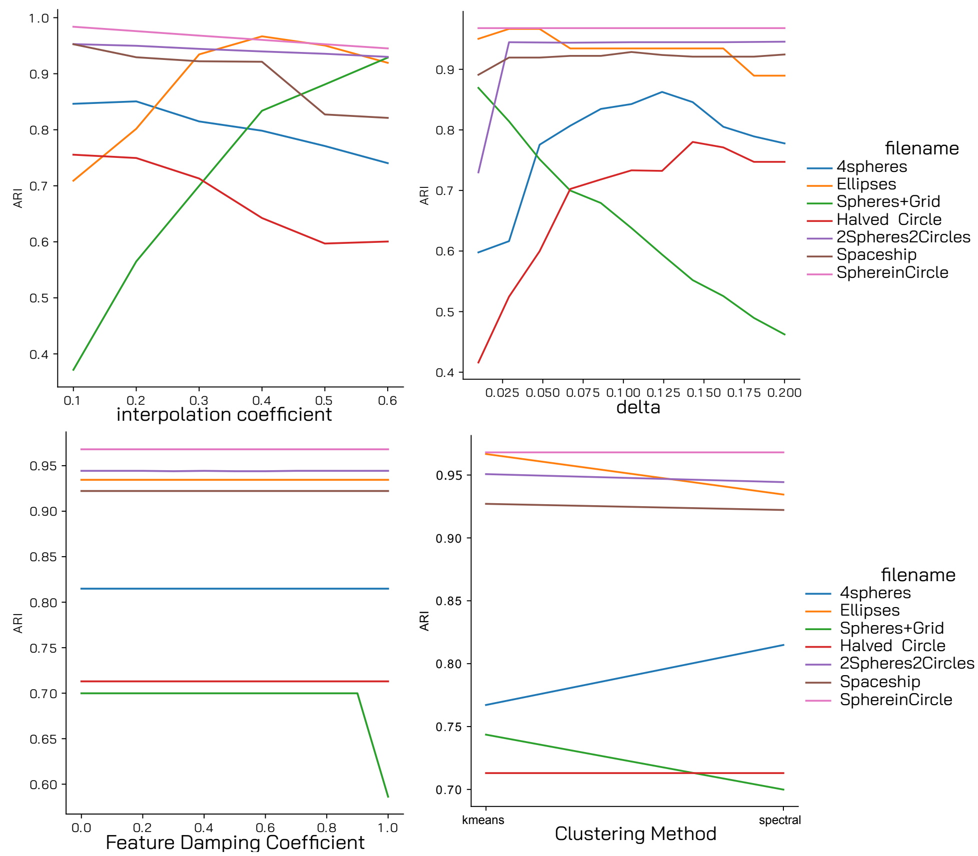


Figure 11: **Hyperparameter robustness of TOPF on the TCBS**. TOPF performs reasonably well across a large range around the default parameters (Interpolation coefficient: 0.3, $\delta$: 0.07, damping coefficient (non-negative): 0, Clustering method: `spectral`). The clustering method is used on the TOPF vectors to determine the cluster-ids.

For the choice of the maximum homology degree $d$ to be considered there are mainly three heuristics which we will list in decreasing importance (Cf. Grande & Schaub (2023a)):

I. In applications, we usually know which kind of topological features we are interested in, which will then determine $d$. This means that 1-dimensional homology and $d = 1$ suffices when we are looking at loops of protein chains. On the other hand, if we are working with voids and cavities in 3d histological data, we need $d = 2$ and thus compute 2-dimensional homology.

II. Algebraic topology tells us that there are no closed $n$-dimensional submanifolds of $\mathbb{R}^n$. Hence their top-homology will always vanish and all interesting homological activity will appear for $d < n$.

III. In the vast majority of cases, the choice will be between $d = 1$ or $d = 2$ because empirically there are virtually no higher-dimensional topological features in practice.

In our quantitative experiments, we have always chosen $d = n - 1$.

**Thresholding parameter $\delta$** In step 4 of the algorithm, we normalise and threshold the harmonic representatives. After normalising, the entries of the vectors lie in the interval of $[0, 1]$. The thresholding parameter $\delta$ now essentially determines an interval of $[0, \delta]$ which we will linearly map to $[0, 1]$, while mapping all entries above $\delta$ to 1 as well. This is necessary as most of the entries in the vector $e_k^i$ are very close to 0 with a very small number of entries being close to 1. Without this thresholding, TOPF would now be almost entirely determined by these few large values. Thus this step limits the maximum possible influence of a single entry. However, because most of the entries of $e_k^i$ are concentrated around 0, small changes in $\delta$ will not have a large effect and we chose $\delta = 0.07$ in all our experiments.

Table 2: **Runtime of individual TOPF steps on TCBS.** We ran TOPF on each of the point clouds of the TCBS and recorded the runtime of the most important individual steps. Note that there is a difference between the total time and the sum of the individual steps, which is due to initialisation time for the julia kernel and other various minor steps. We notice that for smaller and lower-dimensional point clouds, a major part of the runtime is due to steps performed in julia and ripserer calculations, over which we have no influence over. For large 2-dimensional point clouds like `2Spheres2Circles`, we see that the homology computation time increases a bit, but the most significant increase is the time needed to compute the projections to the harmonic representations. *Julia imports* is the time needed by the julia kernel to import the relevant libraries, *Julia reading* is the time Julia needs to read the data from the input stream, *Julia complex* is the time Ripserer needs to build an $\alpha$-complex, *Julia homology* is the time needed for the persistent homology computation by Ripserer, *Reps extraction* is the time needed for selecting the relevant homology representatives, *Harmonic projection* denotes the time needed to build the $\alpha$-complexes by gudhi and to compute the weighted harmonic representatives, *Clustering and dict* is the time needed to perform clustering on the extracted features and condense the results into a dictionary, *total* total is the total runtime of a Jupyter Notebook cell executing the iteration of TOPF.

| dataset | Julia imports | Julia reading | Julia complex | Julia homology | Julia output | Reps extraction | Harmonic projection | Clustering and dict | total |
|---|---|---|---|---|---|---|---|---|---|
| `4spheres` | 3.3s | 2.9s | 2.9s | 4.4s | 0.9s | 0.1s | 0.4s | 0.3s | 15.8s |
| `Ellipses` | 2.8s | 2.7s | 2.7s | 4.2s | 1.0s | <0.1s | 0.1s | 0.1s | 14.7s |
| `4◯+Grid` | 3.3s | 2.7s | 2.7s | 4.1s | 0.9s | 0.1s | 0.5s | 0.3s | 15.1s |
| `Halved ◯` | 3.0s | 2.6s | 2.7s | 4.1s | 0.9s | <0.1s | 0.1s | 0.1s | 14.4s |
| `2Spheres2◯` | 2.7s | 2.7s | 3.3s | 5.5s | 1.3s | 0.7s | 15s | 2.0s | 34.5s |
| `Spherein◯` | 3.0s | 3.0s | 2.9s | 4.7s | 1.1s | <0.1s | 0.3s | 0.3s | 16.1s |
| `Spaceship` | 2.9s | 2.9s | 2.8s | 5.2s | 1.2s | 0.1s | 2.6s | 0.3s | 19.0s |

**Interpolation coefficient $\lambda$** The interpolation coefficient $\lambda \in (0, 1)$ determines whether we build our simplicial complexes close to the birth or the death of the relevant homological features at time $t = b^{1-\lambda}d$. This then in turns controls how localised or smooth the harmonic representative will be. In general, the noisier the ground data is the higher we should choose $\lambda$. However, TOPF is not sensitive to small changes in $\lambda$. We have picked $\lambda = 0.3$ for all the quantitative experiments, which empirically represents a good choice for a broad range of applications.

**Feature selection factor $\beta$** Increasing $\beta$ leads to TOPF preferring to pick a larger number of relevant topological features. Without specific domain-knowledge, $\beta = 0$ represents a good choice.

**Feature selection quotients `max_total_quot`, `min_rel_quot`, and `min_0_ratio`** These are technical hyperparameters controlling the feature selection module of TOPF. For a technical account of them, see Appendix G. In most of the cases without domain knowledge, they do not have an effect on the performance of TOPF and should be kept at their default values.

**Simplicial Complex Weights** Although the simplicial weights are not technically a hyperparameter, there are many potential ways to weigh the considers SCs that can highlight or suppress different topological and geometric properties. In all our experiments, we use $w_\Delta$ weights discussed in Appendix H.

**Ablation study** In Table 3, we performed an ablation study and benchmarked TOPF against TOPF while skipping the harmonic projection step with the Hodge Laplacian (Step **3.**). The result show that in general, skipping the step with the Hodge Laplacian decreases the performance of TOPF by a wide margin. The exception to this are `SphereInCircle` or `2Spheres2Circles`, where the two methods have similar performance. In these two last examples, points are directly sampled from a manifold, and thus the loops and spheres are very "thin". Thus, the homology generator is already a good approximation of all the simplices responsible for a topological feature. In general however, this is not the case, necessating the use of Differential Geometry and the Hodge Laplacian.

### E.2 Runtime

In Table 2, we analyse the runtime of different components of the TOPF pipeline on the point clouds of the TCBS. In Figure 12, we analyse the runtime of the current TOPF implementation on two point

Table 3: **Ablation study.**

| dataset | ARI TOPF | ARI TOPF without harmonic projection |
|---|---|---|
| `4spheres` | **0.81** | 0.08 |
| `Ellipses` | **0.95** | 0.53 |
| `4◯+Grid` | **0.70** | 0.11 |
| `Halved ◯` | **0.71** | 0.15 |
| `2Spheres2◯` | 0.94 | **0.95** |
| `Spherein◯` | 0.97 | **1.00** |
| `Spaceship` | **0.92** | 0.49 |
| mean | **0.86** | 0.47 |

clouds from the topological clustering benchmark suite. We increase the point density while keeping the structure of the point cloud intact. The runtime of our implementation with `sparsification = 'off'` in the regime of Figure 12 appears to scale linearly with the total number of points with a significant constant term. The significant constant term is probably due to our implementation starting a new julia kernel and communicating between python and julia. A python-only or julia-only implementation would thus speed up computations significantly.

**Theoretical runtime complexity**

1. **a). Constructing the complex:** The Vietoris–Rips complex will have $O(n^{k+1})$ simplices, where $n$ is the number of points and $k$ the maximum homology dimension. Apart from this, there is no significant computational effort needed. Note that we can decrease this if we set a maximum VR-radius, as done in many works from the literature. Computing the alpha-complex in ambient dimension $d$ is equivalent to computing the convex hull in ambient dimension $d+1$, which has a complexity of $\sim O(n^{d/2})$. For a fixed dimension $d$, the number of simplices is linear in $n$. **b). Computing persistent homology and generators:** As we need the homology generators, we use the recently introduced involuted persistent homology Čufar & Virk (2023). While the authors discuss the runtime performance, they do not give a concrete complexity bound in Čufar & Virk (2023). However, they deduce that it has a similar run-time to usual persistent cohomology computations, whose performance is for example discussed in De Silva et al. (2011).
2. **Picking the significant generators** has negligible impact.
3. **Computing the harmonic projections.** Given a homology representative $r$ in dimension $k$, computing the harmonic representative amounts to computing a sparse matrix vector product of $B_{k+1}x$ and a sparse least squares problem $\mathrm{lsmr}(B_{k+1}, r)$, i.e. solving
$$\min_{x\in\mathbb{R}^{S_{k+1}}} \|r - B_{k+1}x\|_2$$
where $S_{k+1}$ is the number of $(k+1)$-simplices in the simplicial complex and $B_{k+1}$ has $(k+2)S_{k+1}$ non-zero entries. This is a sparse least-squares problem, which we solve using the iterative sparse solver `lsmr` (Fong & Saunders, 2011). Because this is an iterative solver, the authors do not give runtime complexity, but discuss the computational requirements in Fong & Saunders (2011).
4. **Pooling and averaging** over the simplicial neighbours has negligible runtime constraints.

# F More Details on the Experiments

**High-dimensional Point Clouds** In Figure 5, we use TOPF on a high-dimensional input point cloud representing the conformation space of cyclooctane Martin & Watson (2011). Because the ambient dimension is too large for $\alpha$-filtrations, a Vietoris–Rips filtration with more simplices is used. The VR filtration depends on the ambient dimension only for computing the distance matrix, which is a negligible part of the runtime. Hence, a similar performance can be expected for higher dimensions than 24 as well. To counteract the increase in computational complexity, TOPF automatically downsamples the point cloud using a mixture of landmark and random sampling. We see in the ISOMAP projections of Figure 5 that TOPF extracts reasonable features representing the topology of

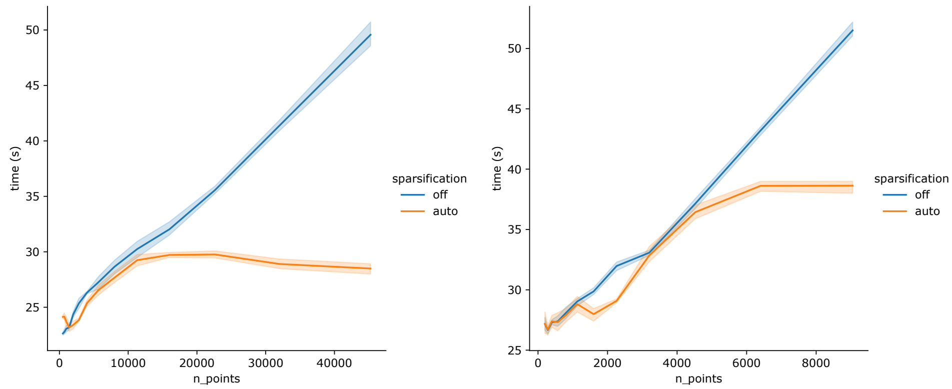


Figure 12: **Runtime on two examples from Topological Clustering Benchmark Suite while increasing point density.** *Left:* `Halved Circle`, *Right:* `2 Spheres 2 Circles`. Experiments kept the topological structure intact while increasing the number of points sampled. Both the runtime of the vanilla TOPF algorithm, as well as of TOPF using automatic subsampling heuristic as implemented in the python package, are listed.

the conformation space. Setting $n_{\text{clusters}}$ to $4$ even reveals the manifold anomalies in a separate cluster (*blue*), similar to the results of Stolz et al. (2020). The runtime for this high-dimensional experiment was $189.8$ seconds.

**Variational Autoencoders** We have trained a VAE on image patches sampled with a topological structure, see Figure 13. We have sampled the image patches around two loops and a connecting line. Using TOPF, we could recover this structure in the latent space of the VAE highlighting how topological point features can help in explainable AI. We note, of course, that many latent spaces do not carry any higher-order topological information, making characterisation by TOPF infeasible. Rather we showed that when we expect a topological structure to exist in the *data set* due to some specifics of the data, we can recover this topological structure even in the latent space of the VAE using TOPF. We have repeated the experiments for a higher-dimensional latent space and received similar results, see Figure 6.

**Additional heatmaps on proteins** In Figure 14, we provide additional experiments of TOPF on protein data. This time, we report every single feature vector produced by TOPF in a separate plot. In the experiment with Cys123, we wanted to show that TOPF can detect so-called protein pockets. Because pockets do not form holes detectable by TDA straightforward, the literature suggests adding (and later removing) these points on the convex hull, which turns the pockets into holes, see for example Oda et al. (2024).

**Performance with decreasing sampling density** In Figure 15, we analyse the robustness of TOPF with respect to a decrease in sampling density. We have selected `2Spheres2Circles` as the dataset with the highest original sampling density for this experiment. It shows that TOPF can still produce meaningful results even for significantly downsampled point clouds. We note that the considered point clouds are in particular very sparse when compared to the point clouds usually considered by neural 3D point cloud architectures. We interpret the results as basically repeating the good performance of TOPF, except for the datasets `EllipsesinEllipses` and `spaceship`, where the performance drops rapidly (at least in a logarithmic plot). Our interpretation of this is that both datasets consist of submanifolds with different sampling density, which intersect/are contained in one another. Thus, the point density is a key to distinguishing the classes, and a random change in these densities due to downsampling can make the encoded structure hard to detect or even causes them to vanish in a sense of topology.

In Figure 16, we analyse the robustness of TOPF with respect to heterogeneous sampling. The results show that TOPF performs well even in the presence of inhomogeneous sampling, and the degrading

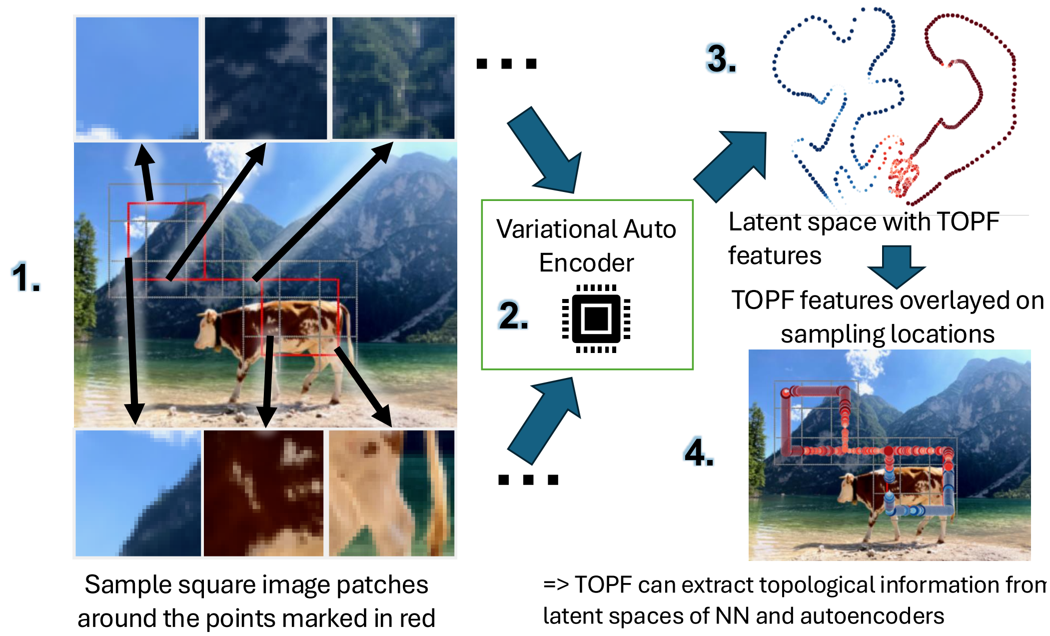


Figure 13: **Using TOPF to explore the topological structure in latent/embedding spaces of Variational Auto Encoders (VAE) 1.** Given a picture (of a cow), we sample $578$ square patches around the centre points marked in red in the image. The centre points are taken from the sides of two squares and a line connecting the two squares. The assumption is that this topological structure (with two holes) is present in the sample space as well. **2.** We train a VAE on the set of $578$ square patches with latent space dimension of $3$ (down from $33 \cdot 33 \cdot 3$.) **3.** We run TOPF with `fixed_num_features` set to $[0, 2]$ on the latent space to extract the two most significant point-wise topological features. **4.** We overlay the topological features from the latent space over the centre points of the corresponding image patches. We see that TOPF has roughly *recovered the topological structure inherent in the sample space* as described in step 1. We note that the image patches with centre on the middle horizontal line are almost identical, making it virtually impossible to distinguish them. Note that although TOPF thinks they contribute to the left loop, their corresponding feature entries are a lot smaller than those of the image patches (shown by lighter colours and smaller dots in the plot.)

of performance can largely be attributed to a decrease in minimum sampling rate, rather than the inhomogeneous nature.

From a theoretical point of view, this should not be surprising: The birth of the topological feature will now depend on the minimum sampling density, while the death time still depends on the size of the feature. The heuristic employed by TOPF will then choose a maximal triangulation radius which lies between these two radii. In the dense part, the triangulation will be denser then in the sparse parts. However, due to the the character of the alpha filtration, the weighting of the simplices discussed in Appendix H and the thresholding this does not cause an issue in the features generated by TOPF.

**Details on experiments with WSDesc** WSDesc (Weakly Supervised 3D Local Descriptor Learning for Point Cloud Registration), Li et al. (2022), uses voxel-based representations of point clouds to extract robust 3D local descriptors for point cloud registration. As showcased in the original paper, WSDesc showcases a good generalisation for point clouds outside of the training set. We used WSDesc pretrained on the 3DMatch data. WSDesc was used with 512 keypoints, and tested with a varying of keypoints up to $\sim 5000$. We thus tested WSDesc with the maximum number of keypoints possible per input point cloud.

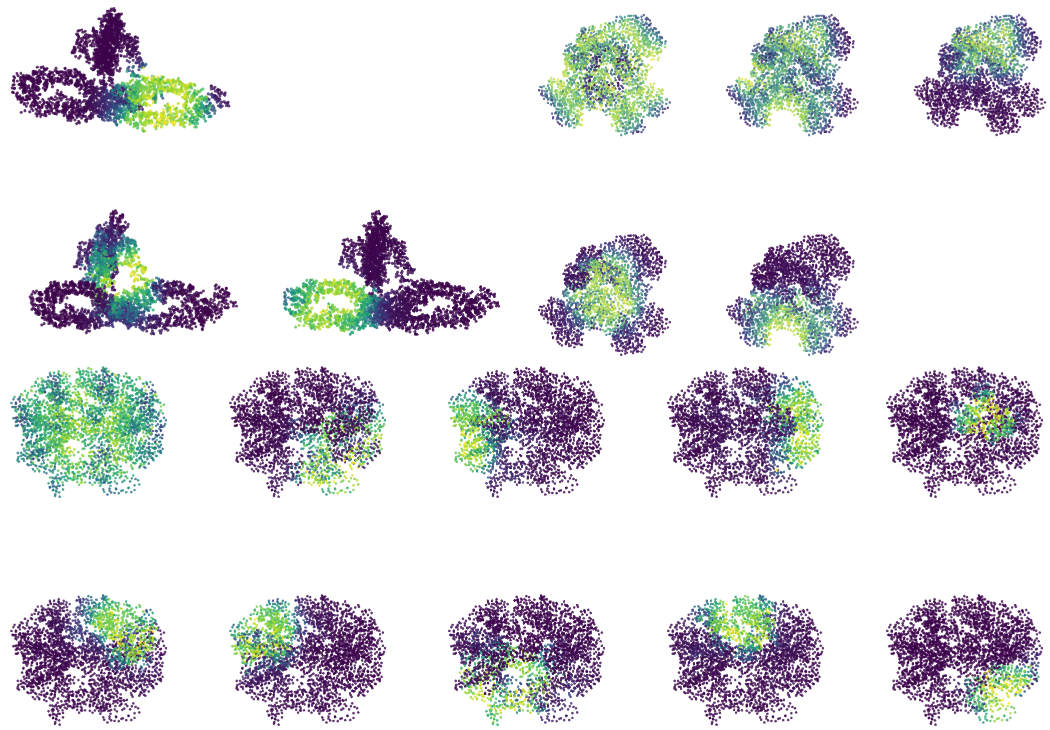

Figure 14: **TOPF heatmaps for three proteins.** *Top left:* NALCN channelosome (Kschonsak et al., 2022) *Top right:* Mutated Cys123 of E. coli (Hidber et al., 2007), with convex hull added during computation, only 2-dimensional homology features *Bottom:* GroEL of E. coli (Chaudhry et al., 2004) (Selected features).

## G How to pick the most relevant topological features

**Simplified heuristic** The persistent homology $P_k$ module in dimension $k$ is given to us as a list of pairs of birth and death times $(b_i^k, d_i^k)$. We can assume these pairs are ordered in non-increasing order of the durations $l_i^k = d_i^k - b_i^k$. This list is typically very long and consists to a large part of noisy homological features which vanish right after they appear. In contrast, we are interested in connected components, loops, cavities, etc. that *persist* over a long time, indicating that they are important for the shape of the point cloud. Distinguishing between the relevant and the irrelevant features is in general difficult and may depend on additional insights on the domain of application. In order to provide a heuristic which does not depend on any a-priori assumptions on the number of relevant features we pick the smallest quotient $q_i^k := l_{i+1}^k / l_i^k > 0$ as the point of cut-off $N_k := \arg\min_i q_i^k$. The only underlying assumption of this approach is that the band of "relevant" features is separated from the "noisy" homological features by a drop in persistence.

**Advanced Heuristic** However, certain applications have a single very prominent feature, followed by a range of still relevant features with significantly smaller life times, that are then followed by the noisy features after another drop-off. This then could potentially lead the heuristic to find the wrong drop-off. We propose to mitigate this issue by introducing a hyperparameter $\beta \in \mathbb{R}_{>0}$. We then define the $i$-th importance-drop-off quotient $q_i^k$ by

$$q_i^k := {}^{l_{i+1}^k}/{}_{l_i^k} \left(1 + {}^{\beta}/{}_{i}\right).$$

The basic idea is now to consider the most significant $N_k$ homology classes in dimension $k$ when setting $N_k$ to be

$$N_k := \arg\min_i q_i^k.$$

Increasing $\beta$ leads the heuristic to prefer selections with more features than with fewer features. Empirically, we still found $\beta = 0$ to work well in a broad range of application scenarios and used

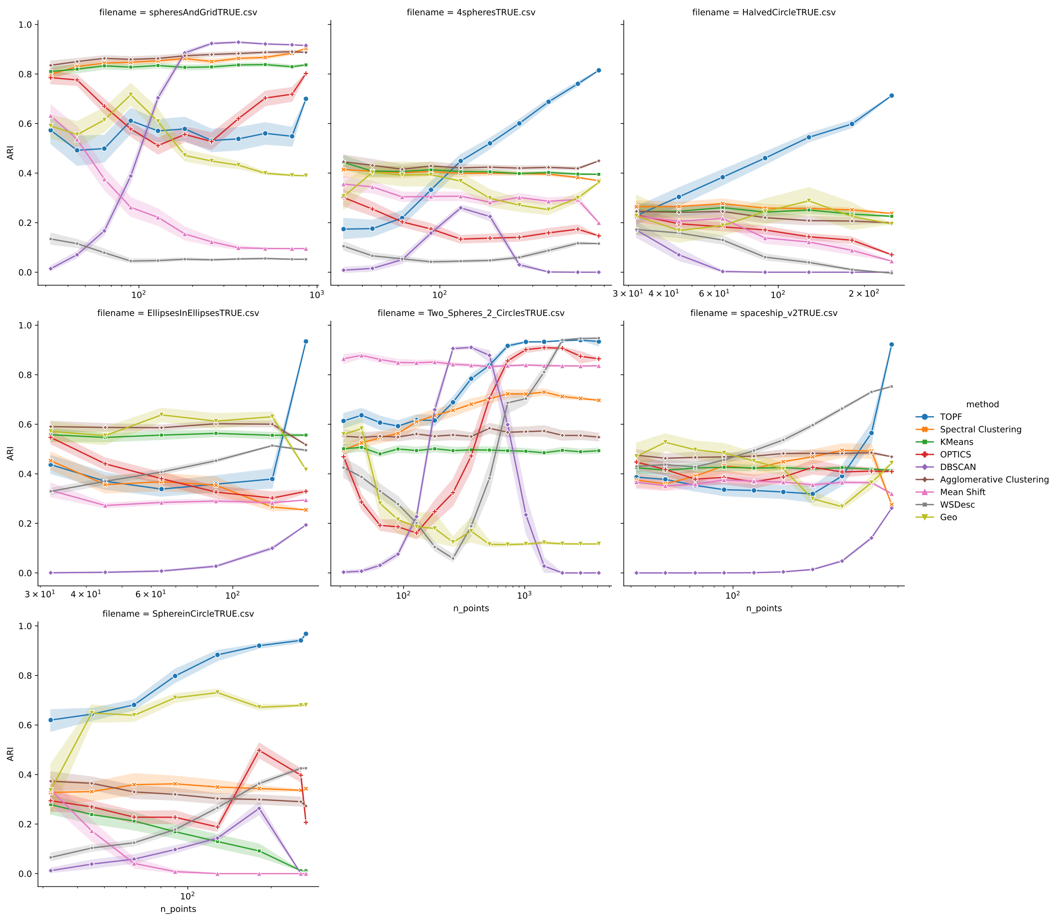


Figure 15: **Performance of TOPF while decreasing sampling density.** The $x$-axis is scaled logarithmically and TOPF achieves an adjsuted rand index (ARI, random algorithms achieve an ARI of over 0.9 for $\sim$ 700 samples, down from 4600 original points on the `2Spheres2CirclesDataset`. The smallest considered sample size was 16 points. The points were sampled at random. We ran each experiment 100 times and report the standard deviation.

it throughout all experiments. There are only a few cases where domain-specific knowledge could suggest picking a larger $\beta$.

To catch edge cases with multiple steep drops or a continuous transition between real features and noise, we introduce two more checks: We allow a minimal $q_i^k$ of $\texttt{min\_rel\_quot} = 0.1$ and a maximal quotient $q_1^h/q_i^k$ of $\texttt{max\_total\_quot} = 10$ between any homology dimensions. Because features in 0-dimensional homology are often more noisy than features in higher dimensions, we add a minimum zero-dimensional homology ratio of $\texttt{min\_0\_ratio} = 5$, i.e. every chosen 0-dimensional feature needs to be at least `min_0_ratio` more persistent then the minimum persistence of the higher-dimensional features. Because these hyperparameters only deal with the edge cases of feature selection, TOPF is not very sensitive to them. For all our experiments, we used the above hyperparameters. We advise to change them only in cases where one has in-depth domain knowledge about the nature of relevant topological features.

**Fixed number of topological features** Alternatively, it is possible to specify a fixed desired number $N_d$ of topological features per dimension $d$. TOPF then automatically returns the $N_d$ most relevant features in dimension $d$. For practical purposes, we can then weigh these features by a function $w(-)$ in the life time $d_i - b_i$ or the life quotient $d_i/b_i$ of the features. Possible picks for $w$ include an exponential function $w(x) = e^x$, a quadratic function $w(x) = x^2$, a linear function $w(x) = x$ or a scaled sigmoid function. Our intuition suggests that when picking functions like a linear function,

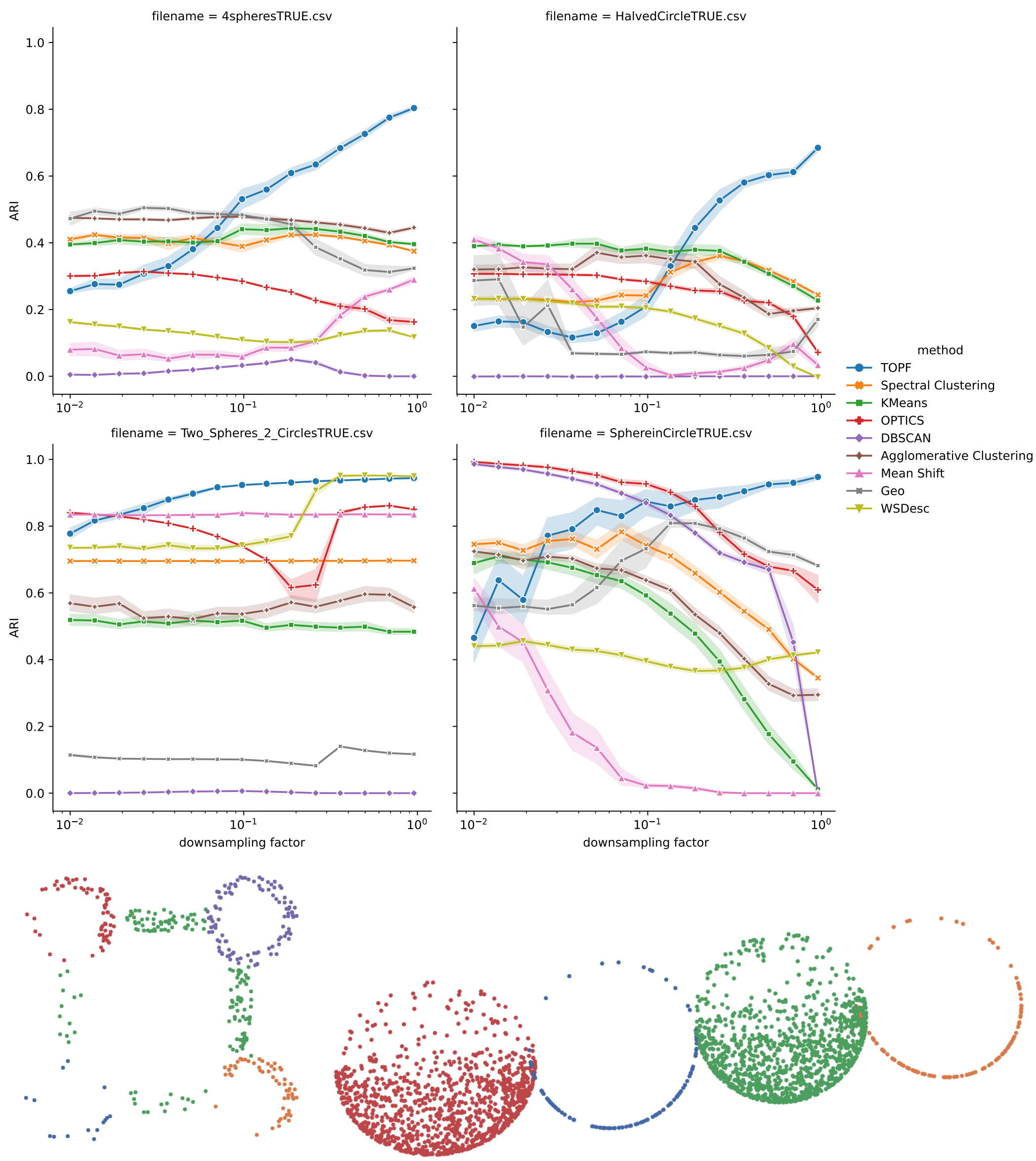


Figure 16: **Performance of TOPF under heterogeneous sampling.** *Top*We divide four of the point clouds with a symmetry into two halves. We downsample only one of the halves, creating sampling irregularities within the individual topological features. We repeat the experiment $100$ times and report the achieved ARI. Note that the lowest value on the $x$-axis means downsampling by a factor of $100$, meaning that the smaller point clouds will then almost only consist of one half. WE compare the performance of TOPF with the other baselines, noting that TOPF outperforms them for reasonable heterogeneities. *Bottom Left:* `4spheres` with downsampling factor $0.2$ with true labels. *Bottom Right:* `2spheres2circles` with downsampling factor $0.1$ with true labels.

the many short-lived features will be given too much weight in comparison to the few more relevant features. Selection the best weight function is an interesting open problem for future work.

# H Simplicial Weights

In an ideal world, the harmonic eigenvectors in dimension $k$ would be vectors assigning $\pm 1$ to all $k$-simplices contributing to $k$-dimensional homological feature, a $0$ to all $k$-simplices not contributing or orthogonal to the feature, and a value in $(-1, 1)$ for all simplices based on the alignment of the simplex with the boundary of the void. However, this is not the case: In dimension $1$, we can for

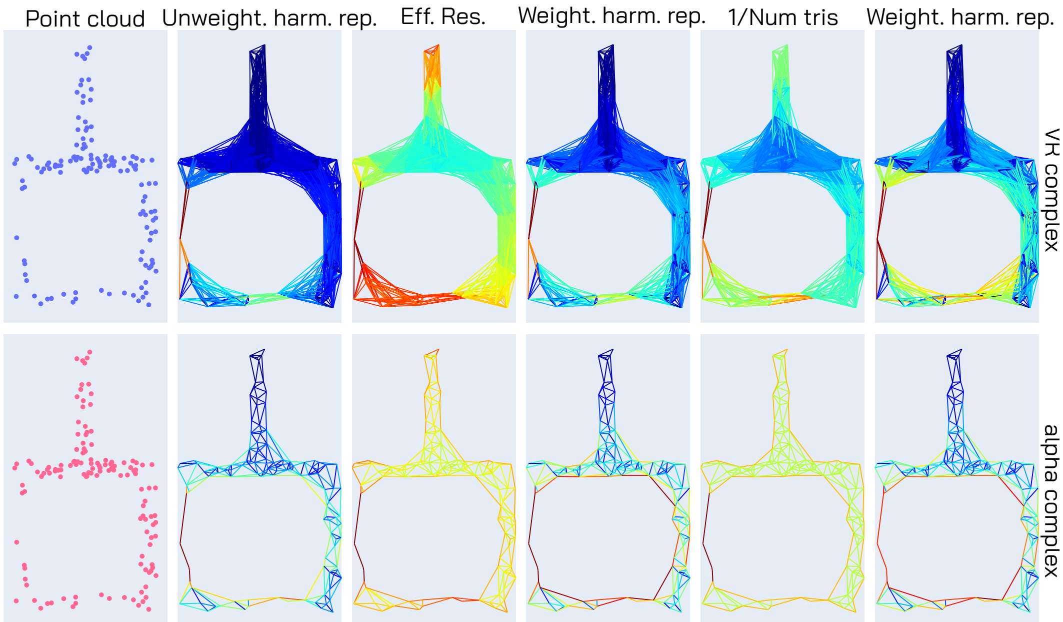


Figure 17: **Effect of weighing a simplicial complex on harmonic representatives.** *Top:* VR complex. *Bottom:* $\alpha$-complex *Left:* The base point cloud with different densities. *2nd Left:* Unweighted harmonic homology representative of the large loop. *3rd Right:* Effective resistance of the 1-simplices. *3rd Right:* Harmonic homology representative of the complex weighted by effective resistance. *2nd Right:* Inverse of number of incident triangles (Definition H.1). *Right:* Harmonic homology representative of the complex weighted by number of incident triangles. Up to a small threshold, the standard harmonic representative in the VR complex is almost exclusively supported in the low-density regions of the simplicial complex. This leads to poor and unpredictable classification performance in downstream tasks. In contrast, the harmonic homology representative of the weighted VR complex has a more homogenous support along the loop, while still being able to discriminate the edges not contributing to the loop. The $\alpha$-complex suffers less from this phenomenon (at least in dimension 2), and hence reweighing is not necessarily required.

example imagine a total flow of 1 circling around the hole. This flow is then split up between all parallel edges which means *two* things: **I** Edges where the loop has a *larger diameter* have *smaller harmonic values* than edges in thin areas and **II** in VR complexes, which are the most frequently used simplicial complexes in TDA, edges in areas with a *high point density* have *smaller harmonic values* than edges in low-density areas. Point **II** is another advantage of $\alpha$-complexes: The expected number of simplices per point does not scale with the point density in the same way as it does in the VR complex, because only the simplices of the Delaunay triangulation can appear in the complex.

We address this problem by weighing the $k$-simplices of the simplicial complex. The idea behind this is to weigh the simplicial complex in such a way that it increases and decreases the harmonic values of some simplices in an effort to make the harmonic eigenvectors more homogeneous. For weights $w \in \mathbb{R}^{\mathcal{S}_k}$, $W = \operatorname{diag}(w)$, the symmetric weighted Hodge Laplacian (Schaub et al., 2020) takes the form of

$$L_k^w = W^{1/2}\mathcal{B}_{k-1}\mathcal{B}_{k-1}^\top W^{1/2} + W^{-1/2}\mathcal{B}_k\mathcal{B}_k^\top W^{-1/2}.$$

Because we want the homology representative to lie in the weighted gradient space, we have to scale its entries with the weight and set $e_{k,w}^i := W^{-1/2}e_k^i$. With this, we have that

$$\mathcal{B}_{k-1}^\top W^{1/2}e_{k,w}^i = \mathcal{B}_{k-1}^\top W^{1/2}W^{-1/2}e_k^i = \mathcal{B}_{k-1}^\top e_k^i = 0$$

We propose two options to weigh the simplicial complex. The first option is to weigh a $k$-simplex by the square of the number of $k+1$-simplices the simplex is contained in:

$$w_\Delta(\sigma_k) = 1/(|\{\sigma_{k+1} \in \mathcal{S}_{k+1}^t : \sigma_k \subset \sigma_{k+1}\}| + 1)^2$$

where the $+1$ is to enforce good behaviour at simplices that are not contained in any higher-order simplices. One of the advantages of the $\alpha$-complex is that we don't have large concentrations of simplices in well-connected areas. The proposed weighting $w_\Delta$ is computationally straightforward, as it can be obtained as the column sums of the absolute value of the boundary matrix $|\mathcal{B}_k|$. The weights also deal with the previously mentioned problem **II**: As the homology representative is scaled inversely to the weight vector $w$, the simplices in high-density regions will be assigned a low weight and thus their weighted homology representative will have a larger entry. By the projection to the orthogonal complement of the curl space, this large entry is then diffused among the high-density region of the SC with many simplices, whereas the lower entries of the simplices in low-density regions are only diffused among fewer adjacent simplices.

However, the first weight is not able to incorporate the number of parallel simplices into the weighting. This is why we propose a second simplicial weight function based on generalised effective resistance.

**Definition H.1** (Effective Hodge resistance weights)**.** For a simplicial complex $\mathcal{S}$ with boundary matrices $(\mathcal{B}_k)$, we define the effective Hodge resistance weights $w_R$ on $k$-simplices to be:

$$w_R \coloneqq \operatorname{diag}\left(\mathcal{B}_{k-1}^{+}\mathcal{B}_{k-1}\right)^2$$

where $\operatorname{diag}(-)$ denotes the vector of diagonal entries and $(-)^+$ denotes taking the Moore–Penrose inverse.

Intuitively for $k = 1$, we can assume that every edge has a resistance of $1$ and then the effective resistance coincides with the notion from Physics. Thus simplices with many parallel simplices are assigned a small effective resistance, whereas simplices with few parallel simplices are assigned an effective resistance close to $1$. However, computing the Moore–Penrose inverse is computationally expensive and only feasible for small simplicial complexes.

In Figure 17, we show that the weights $w_\Delta$ are a good approximation of the effective resistance in terms of the resulting harmonic representative. The standard form of TOPF used in all experiments uses $w_\Delta$-weights.

# I Limitations

**Topological features are not everywhere** The proposed topological point features take relevant persistent homology generators and turn these into point-level features. As such, applying TOPF only produces meaningful results on point clouds that have a topological structure. On these point clouds, TOPF can extract structural information unobtainable by non-topological methods. Although TDA has been successful in a wide range of applications, a large number of data sets does not possess a meaningful topological structure. Applying TOPF in these cases will produce no additional information. Other data sets require pre-processing before containing topological features. In Figure 4 *left*, the $2d$ topological features characterising protein pockets of Cys123 only appear after artificially adding points sampled on the convex hull of the point cloud (Cf. Oda et al. (2024)).

**Computing persistent homology can be computationally expensive** As TOPF relies on the computation of persistent homology including homology generators, its runtime increases on very large point clouds. This is especially true when using VR instead of $\alpha$-filtrations, which become computationally infeasible for higher-dimensional point clouds. Persistent homology computations for dimensions above 2 are only feasible for very small point clouds. Because virtually all discovered relevant homological features in applications appear in dimension $0$, $1$, or $2$, this does not present a large problem. Despite these computational challenges, subsampling, either randomly or using landmarks, usually preserves relevant topological features and thus extends the applicability of TDA in general and TOPF even to very large point clouds.

**Automatic feature selection is difficult without domain knowledge** While the proposed heuristics works well across a variety of domains and application scenarios, only domain- and problem-specific knowledge makes truthful feature selection feasible.

**Experimental Evaluation** There are no benchmark sets for topological point features in the literature, which makes benchmarking TOPF not straightforward. On the level of clustering, we introduced the topological clustering benchmark suite to make quantitative comparisons of TOPF possible, and benchmarked TOPF on some of the point clouds of Grande & Schaub (2023a). On both the level of point features and real-world data sets, it is however hard to establish what a *ground truth* of topological features would mean. Instead we chose to qualitatively report the results of TOPF on proteins and real-world data, see Figure 4.