On the Effectiveness of Persistent Homology

Anonymous Author(s)
Affiliation
Address
email

Abstract

Persistent homology (PH) is one of the most popular methods in Topological Data Analysis. Even though PH has been used in many different types of applications, the reasons behind its success remain elusive; in particular, it is not known for which classes of problems it is most effective, or to what extent it can detect geometric or topological features. The goal of this work is to identify some types of problems where PH performs well or even better than other state-of-the-art methods in data analysis. We consider three fundamental shape analysis tasks: the detection of the number of holes, curvature and convexity from 2D and 3D point clouds sampled from shapes. Experiments demonstrate that PH is successful in these tasks, outperforming several baselines, including PointNet, an architecture inspired precisely by the properties of point clouds. In addition, we observe that PH remains effective for limited computational resources and limited training data, as well as out-of-distribution test data, including various data transformations and noise.

1 Introduction

Persistent homology (PH) is an extension of the idea of homology, which is a way to capture topological information about connectivity and holes in a geometric object. PH can be regarded as a framework to compute representations of raw data that can be used for further processing or as inputs to learning algorithms. There are numerous successful applications of PH in the last decade, from face, gait and activity recognition [92, 48, 52, 44] or digital forensics [6], to discriminating breast cancer subtypes [74], quantifying the porosity of nanoporous materials [51], classifying fingerprints [39], or studying the morphology of leaves [53][1]. At the same time, the reasons behind these successes are not well understood. The goal of this work is to shed some light on this issue, by identifying fundamental classes of problems suitable for PH, and what type of topological or geometric information PH is able to detect.

Here we investigate *what* is seen by persistent homology: Given a data set, i.e., a point cloud, which underlying topological and geometric features can we detect with PH? This question is related to manifold learning and, specifically, topological and geometric inference: Given a finite point cloud X of (noisy) samples from an unknown manifold M, how can one infer properties of M [20, 23, 12, 11]? Obtaining a representation of a shape that can be used in statistical models is an important task in data analysis and many approaches to modeling surfaces and shapes [80].

Since PH is inspired by homology, which provides a measure for the number of connected components, holes, voids, and higher-dimensional holes of a space — to which we collectively refer as "topological features" —, we start with the obvious question of whether PH applied to a point cloud sampled from a geometric object can detect the ***number of (1-dimensional) holes*** of the underlying object. Unlike homology, however, PH registers also the persistence of topological features across scales,

[1] A database of applications of persistent homology is being maintained at [40].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

and can thereby capture geometric information, such as size or position of holes. We therefore also investigate how well PH can detect fundamental geometric notions as are ***curvature*** and ***convexity***. To gauge the performance of PH, we compare it against several baselines. As a first machine learning (ML) baseline we take a SVM trained on the distance matrices of point clouds. We further consider fully-connected neural networks (NN) with a single or multiple hidden layers also trained on distance matrices. As a stronger baseline we consider a PointNet trained on the point clouds directly. PointNet [66, 1] is designed specifically for point cloud data. Similar architectures with convolutional layers (and fully-connected layers and pooling) have been applied for Betti-number and curvature estimation [61, 41].

Related works In spite of the growing interest in PH, so far there is only limited work in the direction that we pursue here. The literature about the performance of PH on the three basic problems we pursue here is sparse. There is indeed theoretical evidence that the number of holes of the underlying space can be detected from PH , under some conditions about the target space, and the sample density and closeness to the space [47, 24, 57]. There is a lot of interest in investigating how well this works in practice, however there are few available results in this direction[2].

Some insights about PH and curvature have been obtained in the literature, starting with an illustrative example in [31, Figure 12] which shows that PH on the filtered tangent complex can distinguish between letters (C and I) that have the same topology, since their curvature is different. Recently, [17] demonstrated that PH can predict curvature (with the computational experiments being replicated in [83]), which inspired us to investigate this problem in more detail.

Regarding the important geometric problem of classification between convex and concave shapes, we were not able to identify any previous works investigating the applicability of PH to this task.

Some further recent work investigating the topological and geometric features seen by PH are the following. Bubenik [16] shows that using PH of points sampled from spheres one can determine the dimension of the underlying spheres. A connection has also been established between PH and the magnitude (an isometric invariant) of a metric space [59]. There have been several efforts in using PH to estimate fractal dimensions, such as [72] in which Schweinhart proves that the fractal dimension of some metric spaces can be recovered from the PH of random samples.

Main contributions Our contributions can be summarized as follows.

- We demonstrate experimentally that PH can detect the number of holes (Section 3), curvature (Section 4), and convexity (Section 5) from point clouds in $\mathbb{R}^2$ or $\mathbb{R}^3$, outperforming SVMs and fully-connected networks trained on distance matrices, and PointNet trained on point clouds.
- We demonstrate experimentally that PH features allow to solve the above tasks even in the case of limited training data (Section 3), noisy (Section 3) and out-of-distribution (Section 5) test data, and limited computational resources (Section 3, Section 4, Section 5).
- We provide insights about the topological and geometric features that are captured with long and short persistence intervals (Section 6), and formulate guidelines for applications that are suitable for PH (Section 7).
- We provide data sets that can be directly used as a benchmark for our tasks or other related point-cloud-analysis or classification problems. We provide computer code to construct more data and replicate our experiments.

2 Background on persistent homology

Homology is a topological concept that attempts to distinguish between topological spaces by constructing algebraic invariants that reflect their connectivity properties [68], i.e., k-dimensional cycles (connected components, holes, voids, ...). The number of independent k-dimensional cycles is called kth Betti number and denoted by β_k. For example, the circle has Betti numbers $\beta_0 = 1$, $\beta_1 = 1$, $\beta_2 = 0$, and for a torus, we have $\beta_0 = 1$, $\beta_1 = 2$, $\beta_2 = 1$.

[2] Some works demonstrate that PH can be used to detect the number of holes, but only on individual toy examples (e.g. [20, Figure 19], [23, Figures 2, 3, 6, 11, 12], [68]), without looking into the statistical significance between different classes of data, or the accuracy of some classification algorithms on a comprehensive data set.

Persistent homology is an extension of this idea that has found success in applications to data [93]. To calculate PH from some data X, we must first build a filtration, i.e., a family of topological spaces $K_1 \subseteq K_2 \subseteq \ldots K_r \subseteq \ldots K_k$ which, in a suitable sense, approximate X at different scales $r \in \mathbb{R}$. Typically, X is a point cloud in $\mathbb{R}^d$, and K_r is a simplicial complex, a set of simplices σ (which we can think of as vertices, edges, triangles, $\ldots$) such that if $\sigma \in K_r$ and $\tau \subseteq \sigma$, then $\tau \in K_r$ [31]. A common choice is $K_r = VR(X, r)$, where $VR(X, r)$ is the Vietoris-Rips simplicial complex, in which $\sigma = \{x_0, \ldots, x_n\} \in VR(X, r)$ when $d(x_i, x_j) \leq r$ for all $1 \leq i, j \leq n$. PH can then be summarized with a persistence diagram (PD), a scatter plot with the x and y axes respectively depicting the scale $r \in \mathbb{R}$ at which each cycle is born and dies or is identified (i.e., merges) with another cycle within a filtration. The length $l = d - b$ of a persistence interval (b, d) measures the lifespan — the so-called "persistence" — of the corresponding cycle in the filtration.

Instead of working directly with persistence diagrams, these are often represented by other signatures that are better suited for machine learning frameworks. A common choice is a persistence image (PI) [2] (discretized sum of Gaussian kernels centered at the PD points), or a persistence landscape (PL) (functions obtained by "stacking isosceles triangles" above persistence intervals, with height reflecting their lifespan) [15]. The steps for extracting PH features are visualized in Figure 1. For good choices of filtration and signature [79], there are theoretical results that guarantee that PH is stable under small perturbations [21, 75]. After the PH signature is calculated, statistical hypothesis testing [15, 10], or some machine learning techniques such as SVM or k-NN [2, 37, 58] can be used on these features to study the differences within the dataset of interest.

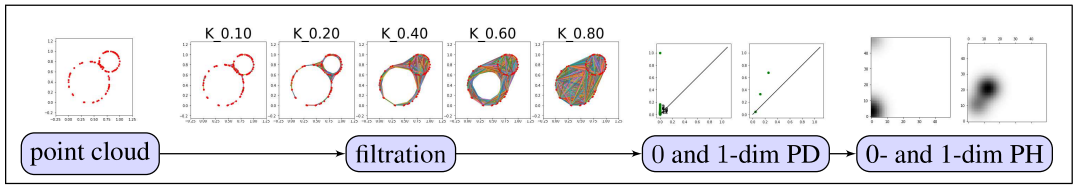


Figure 1: Persistent homology features. For the given example point cloud, 0-dimensional PD has one persistent cycle, reflecting the single connected component, and a number of short cycles that correspond to the individual point-cloud points that are connected to other ones early in the filtration. We have that the 1-dimensional PD summarizes the two holes, whose birth and death values respectively reflect the sparsity along the hole and the size of the hole, as these are the scales $r \in \mathbb{R}$ at which the hole appears and when it is filled in within the filtration. PDs are then represented by PIs, which are vector summaries that can be used in statistical learning frameworks.

It is important to note that PH is very flexible, as different choices can be made in every step of the pipeline, regarding the input and output of PH, as detailed in the remainder of this section.

2.1 Approximation of a space at scale $r \in \mathbb{R}$

Instead of the Vietoris-Rips complex, other types of complexes can be used to approximate the data X at the given scale $r \in \mathbb{R}$. For instance, since Vietoris-Rips simplicial complex is computationally expensive [60], one might rather choose the alpha complex [35], which is closely related to the Vietoris-Rips complex [47], but consists of significantly less simplices and is faster to construct [60, Table 1] (see Figure 10) when the dimension of the ambient space is 2 or 3. If data X is an image rather than a point cloud, or if the point cloud can be seen as an image without losing important information, cubical complexes [45] might be a more suitable choice, where vertices, edges and triangles are replaced by vertices, edges and squares (for details, see Appendix D).

2.2 Filtration

The underlying filtration function in the common PH pipeline introduced above is the distance function $\delta_X : \mathbb{R}^n \to \mathbb{R}$, where $\delta_X(\mathbf{y}) = \min\{d(\mathbf{y}, \mathbf{x}) \mid \mathbf{x} \in X\}$ is the distance to point cloud X. Indeed, the Vietoris-Rips simplicial complex $VR(X, r)$ approximates the sublevel set

$$K_r = \delta_X^{-1}((-\infty, r]) = \{\mathbf{y} \in \mathbb{R}^n \mid \delta_X(\mathbf{y}) \leq r\} = \cup_{\mathbf{x} \in X} B(\mathbf{x}, r),$$

where $B(\mathbf{x}, r)$ is a ball with radius r centered around $\mathbf{x} \in X$ [22].

However, PH on such a filtration is very sensitive to outliers, since even a single outlier changes δ_X significantly. If the data is noisy, it is better to replace the distance function with the Distance-to-Measure (DTM) $\delta_{X,m} : \mathbb{R}^n \to \mathbb{R}$, where $\delta_{X,m}(x)$ is the average distance from a number of neighbors on the point cloud [22, 5] (see Figure 10). However, depending on the task, there are many other filtration functions one could choose, such as rank [63], height, radial, erosion, dilation [37], and the resulting PH captures completely different information about the cycles [79]. For example, whereas PH with respect to the Vietoris-Rips filtration encodes the size of the hole (Figure 1), PH on the height filtration informs about the position of the hole (Figure 16). For a more detailed discussion about the influence of filtration, see Appendix E.

2.3 Persistence signature

Next to PIs and PLs, a plethora of persistence signatures has been introduced in the literature, e.g., Betti numbers [43, 82] or Euler characteristic [54] (across scales), or even scalar summaries such as amplitude [37], entropy [27, 70], or algebraic functions of the birth and death values [4, 46]. Some of these signatures summarize the same information, but lie in different metric spaces [81]. Others, however, such as the scalar summaries listed above, discard information compared to PDs, as it might sometimes be useful to, e.g., only capture the (total or maximum) persistence of intervals, but not all the detailed information about all birth and death values (see Appendix E).

3 Number of holes

In this section, we focus on the task of (ordinal) classification of point clouds by the number of 1-dimensional holes. Research in psychology shows that global properties often dominate perception, and in particular, that topological invariants such as number of holes, inside versus outside, and connectivity can be effective primitives for recognizing shapes [64]. Extracting such topological information can therefore prove useful for many computer vision tasks.

Data We consider 20 different shapes in $\mathbb{R}^2$ and $\mathbb{R}^3$, with four different shapes having the same number of holes ($0, 1, 2, 4$ or 9). For each shape, we construct 50 point clouds each with $1\,000$ points sampled from a uniform distribution over the shape, resulting in a balanced dataset of $1\,000 = 20 \times 50$ point clouds. A few examples of these point clouds are shown in Figure 2. The label of a point cloud is the number of holes in the underlying shape.

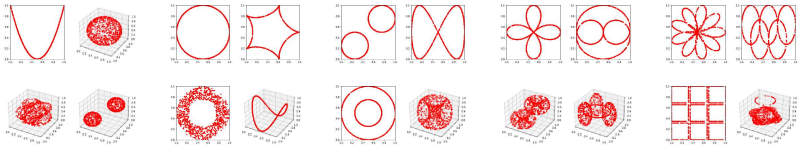
shapes					
number of holes	0	1	2	4	9

Figure 2: Number of holes dataset.

PH pipeline For each point cloud X and scale $r \in \mathbb{R}$, we consider its alpha complex. We will look into scenarios in which data contains noise, and therefore, instead of the standard distance function, we consider Distance-to-Measure (DTM) as the filtration function. We extract 1-dimensional PDs, that are then transformed to PIs, PLs, or a simple signature consisting only of lifespans $l = d - b$ of the 10 most persisting cycles [3], and classified with an SVM. We consider a "PH simple" pipeline which relies on the 10 lifespans, and denote with "PH" the pipeline wherein grid search is employed to choose the best out of the three aforementioned persistent signatures and the values of their parameters. For details, see Appendix A and Appendix B.

[3] Although a sphere has no 1-dimensional holes, its PD might consist of many short intervals which correspond to the small holes on the surface. In addition, in the presence of noise, additional small holes might appear for any point cloud. Hence it is not a good idea to consider the cardinality |PD| of the PD as the signature.

Results We investigate the clean and robust test accuracy under four types of noisy transformations (gaussian noise, outliers). Details about the transformations are given in Appendix A. We train the classifier on 80% of the clean point clouds, and test on the remaining 20% of the clean data or transformed versions thereof. The results reported in Figure 3 show that PH obtains very good accuracy for this (robust) classification problem, even in the presence of affine transformations or noise, outperforming baseline machine- and deep-learning techniques.[4]

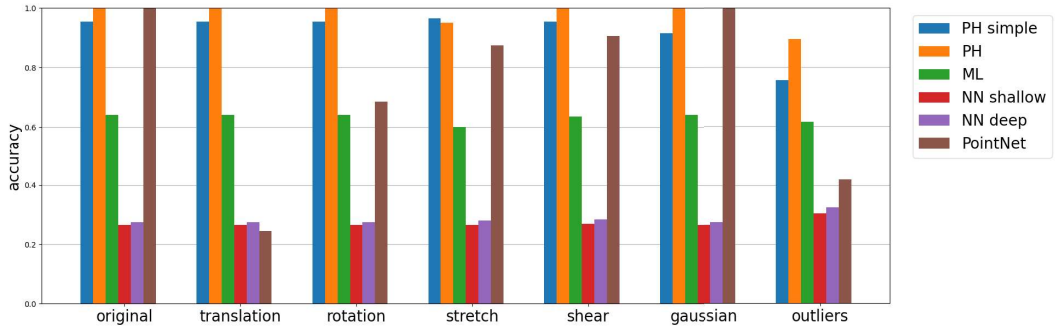


Figure 3: Persistent homology can detect the number of holes.

We reach a similar conclusion in case of limited training data and computational resources. The evolution of accuracy across training epochs, and different amounts of training data is reported in Appendix B. For both PH pipeline and neural networks we conduct hyperparameter tuning. Since we consider a wide range of parameters for the different persistence signatures, the tuning of the PH pipeline does take time (less than tuning only a few hyperparameters for PointNet). Figure 3 shows that even the simple PH pipeline, where SVM is used directly on the lifespans of the 10 most persisting cycles (without any tuning of PH-related parameters) performs well.

4 Curvature

This section considers a regression task to predict the curvature of an underlying shape based on a point cloud sample. Estimating curvature-related quantities is of prime importance in computer vision, computer graphics, computer-aided design or computational geometry, e.g., for surface segmentation, surface smoothing or denoising, surface reconstruction, and shape design [19]. For continuous surfaces, normals and curvature are fundamental geometric notions which uniquely characterize local geometry up to rigid transformations [41]. Recently, PH has been shown to be able to detect curvature [17]. We run a similar experiment, evaluating the PH pipeline against our baselines.

Data A balanced dataset is generated in the same way as in [17]: We consider unit disks D_K on surfaces of constant curvature K: (i) $K = 0$, Euclidean plane, (ii) $K > 0$, sphere with radius $1/\sqrt{K}$, and (iii) $K < 0$, Poincaré disk model of the hyperbolic plane (surface of a saddle, or a potato chip). Curvature K lies in the interval $[-2, 2]$ so that a disk with radius one can be embedded on the upper hemisphere of a sphere with constant curvature K (as it spherical cap). For each $K \in \{-2, -1.96, \ldots, -0.04, 0, 0.04, \ldots, 1.96\}$, we construct 10 point clouds by sampling $1\,000$ points from the unit disk D_K with the probability measure proportional to the surface area measure [15, Section 2.7, Section 4.1]. A few examples are illustrated in Figure 4. These $101 \times 10 = 1\,010$ point clouds are considered as the training data, whereas the test dataset is built in a similar way for 100 values of K chosen uniformly at random from $[-2, 2]$. The label of a point cloud is the curvature K of the underlying disk D_K. Note that all these disks are homoemorphic: they are contractible, so that their homology is trivial and homology is thus unable to distinguish between them [17].

[4] Interestingly, although PointNet was designed with the idea to be invariant to affine transformations, it performs poorly when the test data is translated or rotated (and this is consistent with some previous results [89, 88, 50, 87, 90, 85, 91]), or when it contains outliers. Traditional neural networks perform very poorly, which might not come as a big surprise, since it was recently demonstrated that they transform topologically complicated data into topologically simple one as it passes through the layers, vastly reducing the Betti numbers (nearly always even reducing to their lowest possible values: $\beta_k = 0$ for $k > 0$, and $\beta_0 = 1$) [56]. Of course, the choice of activation function and hyperparameters might have an important influence on performance [56].

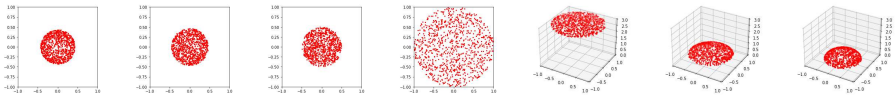
shapes							
curvature	-2.00	-1.00	-0.10	0.00	0.10	1.00	2.00

Figure 4: Curvature dataset.

PH pipeline For each point cloud X, we first calculate the suitable matrix of pairwise distances between the point-cloud points: hyperbolic, Euclidean or spherical, respectively for negative, zero and positive curvature [15, Section 2.7]. The input for PH is the filtered Vietoris-Rips simplicial complex.[5] We extract 0- and 1-dimensional PDs, which are then transformed into PIs, PLs or lifespans, to be fed to SVM. This PH pipeline is visualized in Figure 14.

Results Figure 5 shows the mean squared errors for the PH and other pipelines, together with their regression lines. The results show that PH indeed detects curvature, outperforming other methods.[6] Next to the PH pipeline discussed above, we also consider SVM on simple PH features (PH simple), which are lists of lifespans. In order to investigate if all persistence intervals contribute to prediction, we also consider SVM only on the 10 longest lifespans (PH simple 10). Similarly as in Section 3, the grid search across the different parameters for persistence signatures does take time (Figure 15), but Figure 5 shows that SVM on a simple signature of all lifespans performs well. However, we note that the performance drops if we only focus on the longest 10 intervals. Therefore, the many short intervals together capture the geometry of interest for this problem.

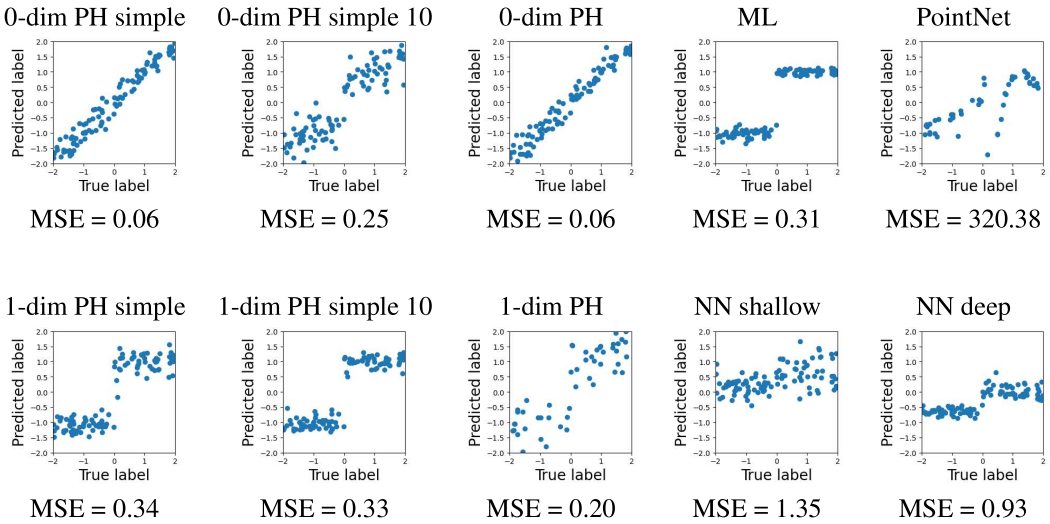


Figure 5: Persistent homology can detect curvature.

We highlight that the data used in Bubenik's paper [17], and thereby in this work, is sampled from surfaces with *constant* curvature. It would therefore be interesting to see similar experiments on shapes with non-constant curvature.

[5] Alpha complex is faster to compute, but requires information about the position, i.e., the coordinates of the points and therefore cannot be calculated on a distance matrix. To calculate PH, we rely on the Ripser software [78], which is at the time the most efficient library to compute PH with Vietoris-Rips complex [60].

[6] Simple machine and deep learning techniques are able to differentiate between positive and negative curvature, but perform poorly in predicting the actual value of the curvature of the underlying surface.

5 Convexity

In this section, we consider the binary classification task that consists in detecting whether a point cloud is sampled from a convex set. Convexity is a fundamental concept in geometry [32], which plays an important role in learning, optimization [9], numerical analysis, statistics, information theory, and economics [69]. Furthermore, points of convexities and concavities have been demonstrated as crucial for human perception of shapes across many experiments [71].

To the best of our knowledge, prior to our work PH has not been employed to analyze convexity, and it is a task for which we find PH's effectiveness to be surprising. In the first decade after the introduction of PH, it was seen primarily as the descriptor of global topology. Recently, there have been many discussions and greater understanding that PH also captures local geometry [3]. However, it is still suggested that the long persistence intervals capture topology (as was the case with the detection of holes in Section 3), and many—even too many for the human eye to count—short persistence intervals capture geometrical properties (as was the case with curvature prediction in Section 4). However, our experiments suggest that it is a single, and the second-longest persistence interval that enables us to detect concavity.

Data We construct a balanced dataset of $1\,000$ point clouds by sampling $5\,000$ points from convex and concave (nonconvex) shapes in $\mathbb{R}^2$. First, we consider the "regular" convex shapes of triangle, square, pentagon and circle, and their concave variants, sampling 60 point clouds of each of the eight shapes, 480 point clouds in total. Next, we build 480 "random" convex and concave shapes, in order to be able to investigate if an algorithm is actually detecting convexity, or only the different basic shapes. A few examples are shown in Figure 6. To construct a random convex shape, we generate 10 points at random, and then build their convex hull using the quickhull algorithm [84]. We construct random concave shapes in a similar way, but instead of the convex hull, we build the alpha shape [34, 36] with the optimized α parameter, which gives a finer approximation of a shape from a given set of points. If the alpha shape is convex (i.e., if the alpha shape and its convex hull are the same), we reconstruct the concave shape from scratch. A point cloud has label 1 if it is sampled from a convex shape, and 0 otherwise.

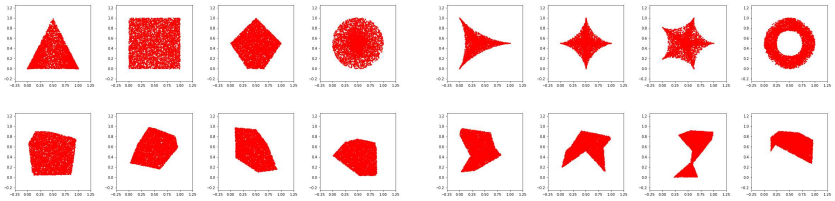
shapes		
convexity	1	0

Figure 6: Convexity dataset.

PH pipeline To build a filtration, we consider cubical complexes filtered by the height function $f_v(p) = \langle p, v \rangle$, where v is a unit vector. The height function is equal to the distance from the hyperplane orthogonal to v, so that, for a good choice of v, multiple connected components would be seen in the filtration of a point cloud sampled from a concave shape, at least for some values $r \in \mathbb{R}$. For this reason we consider the cubical complex, rather than the standard Vietoris-Rips simplicial complex wherein these separate components would commonly be connected with an edge. To build an image from the point cloud, we define a pixel as black if it contains any point-cloud points, and white otherwise.

Since sources of concavity can lie anywhere on the point cloud, we consider nine different directions for the height filtration function (for details see Figure 16). For each of the nine directions, we extract 0-dimensional PD, as it captures information about the connected components. If the point cloud is thus sampled from a convex shape, its PD will only see a single cycle from any direction, whereas there will be multiple cycles at least for some direction for points clouds sampled from concave shapes. For this reason, for each of the nine directions, we focus our attention only on the lifespan of the second most persisting cycle. We can consider this 9-dimensional vector as our PH signature, but

in our experiment choose an even simpler summary: the maximum of these lifespans, since we only care if there are multiple connected components *from at least one direction*. This scalar could even be used as some measure of a level of concavity of a shape.

Results As already indicated, to gain some insights into how well the different approaches discriminate convexity from concavity rather than differentiating between the different basic shapes, we look at the classification accuracies under different conditions (Figure 7). We start with the easiest case, where both the train and test data consist of the simple regular convex and concave shapes (Figure 6, first row), and then proceed to the scenario where both train and test data are random shapes (Figure 6, second row). Next we proceed to out-of-distribution test data, where we train on the regular and test on random shapes, or vice versa. In every case, we train on 400 and test on 80 point clouds. The results show that PH is able to detect convexity, surpassing other methods significantly in all scenarios except for PointNet in the scenario on the dataset of regular shapes which performs on par. Results reported in Appendix D show PH is also computationally efficient.

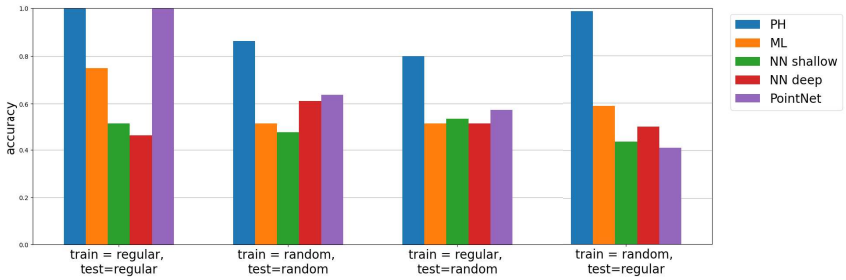


Figure 7: Persistent homology can detect convexity.

The PH pipeline above makes a wrong prediction when concavity is barely pronounced, or if it is missed by the selected height filtration directions (for details see Appendix D). This would also happen if we would include in the dataset some shapes that have small or non-central holes, e.g., a square with a hole in the top left corner. However, the accuracy of PH can easily be improved simply by considering a finer resolution for the cubical complexes and/or additional height filtration directions.[7] The pipeline is not limited to polygons, connected shapes, nor to point clouds in $\mathbb{R}^2$, and although more directions should be considered for high-dimensional point clouds, the curse of dimensionality can be avoided by considering random directions.

6 Implications for PH interpretation: topology vs. geometry

Here we discuss how our results contribute to the important and ongoing discussion about the interpretation of long versus short persistence intervals. When PH was first introduced in the literature, the long intervals were commonly considered as important or "signal", and short intervals as irrelevant or "noise" [30]. Subsequently the discussion has refined when it was shown that short and medium-length persistence intervals have the most distinguishing power for specific types of applications [8, 76]. The current understanding is roughly that long intervals reflect the topological signal, and (many) short intervals can help in detecting geometric features [3]. We believe that our work brings new insight into this discussion. We give a summary of the implications of our work in this section, and we provide a more detailed discussion in Appendix E.

[7] There is theoretical evidence that PH with respect to the height filtration from *every* direction (so-called persistent homology transform) is a sufficient shape descriptor [80, 33, 38], and there are also some results on the sufficient number of height filtration function directions [33]. In computational experiments in [80] on the MPEG-7 silhouette database of simulated shapes in $\mathbb{R}^2$ [73, 49] and the point clouds in $\mathbb{R}^3$ obtained from micro-CT scans of heel bones [13], the authors approximate the persistent homology transform by looking respectively at 64 evenly spaced directions, and 162 directions from a sphere based on a grid constructed by subdividing an icosohedron. 0- and/or 1-dimensional PH with respect to the height and/or related radial filtration has also been used as a 3D shape descriptor in [18], for analysis of brain artery trees [8], classification of MNIST images of handwritten digits [37], etc.

Topology and long persistence Stability results guarantee that a number of longest persistence intervals reflect the topological signal, i.e., the number of cycles [24]. These theorems give information about the threshold that differentiates between long and short persistence intervals. In Section 3, where we focus on the topology of underlying shapes, the experiments demonstrate that this threshold can be learned with simple machine learning techniques. However, it is important to highlight that the distinction between long and short persistence is vague in practice. Indeed, seemingly short persistence intervals capture the topology in Section 3, but the second-longest interval (which allows us to discriminate between convex and concave shapes) is topological noise in Section 5, since every shape in the dataset has a single connected component. These two problems also clearly indicate how the long intervals that encode topology might or might not be irrelevant, depending on the signal of the particular application domain.

Geometry and short persistence The current understanding is that (many) short persistence intervals detect geometry. Section 4 confirms that this indeed can be the case. However, we highlight that *all* cycles can encode geometric information, such as the information about their size (with respect to the Vietoris-Rips and related filtrations, as in Sections 3 and 4) or their position (with respect to the height filtration, as in Section 5). This further implies that, depending on the application, *any number* of short or intervals of *any persistence* can be important, which was clearly demonstrated in Section 5, where we show that a single interval detects convexity.

7 Conclusions

Main contribution The goal of this work is to gain a better understanding of the topological and geometric features that can be captured with persistent homology. We evaluate its performance for the detection of number of holes (Section 3), curvature (Section 4), and convexity (Section 5) of synthetic point clouds in $\mathbb{R}^2$ and $\mathbb{R}^3$, and demonstrate that is can solve these problems, outperforming a few baselines. This is true even when there is limited training data and computational resources, and for noisy or out-of-distribution test data.

Relevance Firstly, the findings point the way to further advances in utilizing the potential of PH in applications: we can expect PH to be successful for classification or regression problems where the data classes differ with respect to number of holes, curvature and/or convexity. Detailed guidelines are discussed in Appendix E. Due to the crucial role of shape classification in understanding and recognizing physical structures and objects, image processing and computer vision [55], our results demonstrate that PH can—to borrow the words from Wigner [86]—“remain valid in future research, and extend, to our pleasure", and lesser bafflement, to a variety of applications.

Secondly, the results advance the discussion about the importance of long and short persistence intervals, and their relationship to topology and geometry (Section 6). Topology is captured by the long intervals, geometry is encoded in all persistence intervals, and any interval can encode the signal in the particular application domain.

Limitations The results focus on the three selected problems and datasets, and it would therefore be interesting to consider other tasks. In addition, we do not have an extensive comparison of the state-of-the-art for the given problems. Our work seeks to understand if PH is successful for a selected set of tasks by benchmarking it against some well-performing methods.

Future research An in-depth analysis of the hypothetical applications discussed in this paper (Appendix E) and selected success stories of PH from the literature could further improve our understanding of the topological and geometric information encoded in PH, and the interpretation of persistence intervals of different lengths. Furthermore, even though our results imply that PH features are recommended over baseline models for the three selected classes of problems, they also provide inspiration on how to improve existing learning architectures. Further work could investigate deep learning models on PH (and standard) features or kernels [42, 14, 67], an additional network layer for topological signatures, or PH-based priors, regularization or loss function [14, 26, 29, 28].

Potential negative societal impact While we recognize the applications of shape analysis can take many different directions, we do not foresee a direct path of this research to negative societal impacts.

References

[1] Point cloud classification with PointNet. `https://keras.io/examples/vision/pointnet/`. Accessed: 2022-02-01.

[2] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *The Journal of Machine Learning Research*, 18(1):218–252, 2017.

[3] Henry Adams and Michael Moy. Topology applied to machine learning: From global to local. *Frontiers in Artificial Intelligence*, 4:54, 2021.

[4] Aaron Adcock, Erik Carlsson, and Gunnar Carlsson. The ring of algebraic functions on persistence bar codes. *arXiv preprint arXiv:1304.0530*, 2013.

[5] Hirokazu Anai, Frédéric Chazal, Marc Glisse, Yuichi Ike, Hiroya Inakoshi, Raphaël Tinarrage, and Yuhei Umeda. Dtm-based filtrations. In *Topological Data Analysis*, pages 33–66. Springer, 2020.

[6] Aras Asaad and Sabah Jassim. Topological data analysis for image tampering detection. In *International workshop on digital watermarking*, pages 136–146. Springer, 2017.

[7] Ulrich Bauer. Ripser: efficient computation of Vietoris-Rips persistence barcodes. *Journal of Applied and Computational Topology*, 2021.

[8] Paul Bendich, James S Marron, Ezra Miller, Alex Pieloch, and Sean Skwerer. Persistent homology analysis of brain artery trees. *Annals of Applied Statistics*, 10(1):198, 2016.

[9] Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. Testing convexity of figures under the uniform distribution. *Random Structures & Algorithms*, 54(3):413–443, 2019.

[10] Eric Berry, Yen-Chi Chen, Jessi Cisewski-Kehe, and Brittany Terese Fasy. Functional summaries of persistence diagrams. *arXiv preprint arXiv:1804.01618*, 2018.

[11] Omer Bobrowski and Sayan Mukherjee. The topology of probability distributions on manifolds. *Probability theory and related fields*, 161(3):651–686, 2015.

[12] Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. *Geometric and topological inference*, volume 57. Cambridge University Press, 2018.

[13] Doug M Boyer, Jesus Puente, Justin T Gladman, Chris Glynn, Sayan Mukherjee, Gabriel S Yapuncich, and Ingrid Daubechies. A new fully automated approach for aligning and comparing shapes. *The Anatomical Record*, 298(1):249–276, 2015.

[14] Rickard Brüel-Gabrielsson, Bradley J Nelson, Anjan Dwaraknath, Primoz Skraba, Leonidas J Guibas, and Gunnar Carlsson. A topology layer for machine learning. *arXiv preprint arXiv:1905.12200*, 2019.

[15] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research*, 16(1):77–102, 2015.

[16] Peter Bubenik and Paweł Dłotko. A persistence landscapes toolbox for topological statistics. *Journal of Symbolic Computation*, 78:91–114, 2017.

[17] Peter Bubenik, Michael Hull, Dhruv Patel, and Benjamin Whittle. Persistent homology detects curvature. *Inverse Problems*, 36(2):025008, 2020.

[18] Mathieu Carrière, Steve Y Oudot, and Maks Ovsjanikov. Stable topological signatures for points on 3D shapes. In *Computer Graphics Forum*, volume 34, pages 1–12. Wiley Online Library, 2015.

[19] Frédéric Cazals and Marc Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design*, 22(2):121–146, 2005.

[20] Frédéric Chazal and David Cohen-Steiner. Geometric inference. *submitted as a book*, 2013.

[21] Frédéric Chazal, David Cohen-Steiner, André Lieutier, and Boris Thibert. Stability of curvature measures. In *Computer Graphics Forum*, volume 28, pages 1485–1496. Wiley Online Library, 2009.

[22] Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric inference for probability measures. *Foundations of Computational Mathematics*, 11(6):733–751, 2011.

[23] Frédéric Chazal, Brittany Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, Alessandro Rinaldo, and Larry Wasserman. Robust topological inference: Distance to a measure and kernel distance. *The Journal of Machine Learning Research*, 18(1):5845–5884, 2017.

[24] Frédéric Chazal and Steve Yann Oudot. Towards persistence-based reconstruction in Euclidean spaces. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, pages 232–241, 2008.

[25] Chao Chen and Michael Kerber. Persistent homology computation with a twist. In *Proceedings 27th European workshop on computational geometry*, volume 11, pages 197–200, 2011.

[26] Chao Chen, Xiuyan Ni, Qinxun Bai, and Yusu Wang. A topological regularizer for classifiers via persistent homology. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2573–2582. PMLR, 2019.

[27] Harish Chintakunta, Thanos Gentimis, Rocio Gonzalez-Diaz, Maria-Jose Jimenez, and Hamid Krim. An entropy-based persistence barcode. *Pattern Recognition*, 48(2):391–401, 2015.

[28] James Clough, Nicholas Byrne, Ilkay Oksuz, Veronika A Zimmer, Julia A Schnabel, and Andrew King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[29] James R Clough, Ilkay Oksuz, Nicholas Byrne, Julia A Schnabel, and Andrew P King. Explicit topological priors for deep-learning based image segmentation using persistent homology. In *International Conference on Information Processing in Medical Imaging*, pages 16–28. Springer, 2019.

[30] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[31] Anne Collins, Afra Zomorodian, Gunnar Carlsson, and Leonidas J Guibas. A barcode shape descriptor for curve point cloud data. *Computers & Graphics*, 28(6):881–894, 2004.

[32] Loïc Crombez, Guilherme D da Fonseca, and Yan Gérard. Efficient algorithms to test digital convexity. In *International Conference on Discrete Geometry for Computer Imagery*, pages 409–419. Springer, 2019.

[33] Justin Curry, Sayan Mukherjee, and Katharine Turner. How many directions determine a shape and other sufficiency results for two topological transforms. *arXiv preprint arXiv:1805.09782*, 2018.

[34] Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. On the shape of a set of points in the plane. *IEEE Transactions on information theory*, 29(4):551–559, 1983.

[35] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 454–463. IEEE, 2000.

[36] Herbert Edelsbrunner and Ernst P Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG)*, 13(1):43–72, 1994.

[37] Adélie Garin and Guillaume Tauzin. A topological "reading" lesson: Classification of MNIST using TDA. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1551–1556. IEEE, 2019.

[38] Robert Ghrist, Rachel Levanger, and Huy Mai. Persistent homology and Euler integral transforms. *Journal of Applied and Computational Topology*, 2(1):55–60, 2018.

[39] Noah Giansiracusa, Robert Giansiracusa, and Chul Moon. Persistent homology machine learning for fingerprint classification. *arXiv preprint arXiv:1711.09158*, 2017.

[40] Barbara Giunti. *TDA-Applications*. zotero database, 2020.

[41] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J Mitra. PCPNET learning local shape properties from raw point clouds. In *Computer Graphics Forum*, volume 37, pages 75–85. Wiley Online Library, 2018.

[42] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. *arXiv preprint arXiv:1707.04041*, 2017.

[43] Umar Islambekov, Monisha Yuvaraj, and Yulia R Gel. Harnessing the power of topological data analysis to detect change points in time series. *arXiv preprint arXiv:1910.12939*, 2019.

[44] Maria-Jose Jimenez, Belen Medrano, David Monaghan, and Noel E O'Connor. Designing a topological algorithm for 3D activity recognition. In *International Workshop on Computational Topology in Image Context*, pages 193–203. Springer, 2016.

[45] Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek. *Computational homology*, volume 157. Springer Science & Business Media, 2004.

[46] Sara Kališnik. Tropical coordinates on the space of persistence barcodes. *Foundations of Computational Mathematics*, 19(1):101–129, 2019.

[47] Jisu Kim, Jaehyeok Shin, Frédéric Chazal, Alessandro Rinaldo, and Larry Wasserman. Homotopy reconstruction via the Cech complex and the Vietoris-Rips complex. *arXiv preprint arXiv:1903.06955*, 2019.

[48] Javier Lamar-León, Edel B Garcia-Reyes, and Rocio Gonzalez-Diaz. Human gait identification using persistent homology. In *Iberoamerican Congress on Pattern Recognition*, pages 244–251. Springer, 2012.

[49] Longin Jan Latecki, Rolf Lakamper, and T Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 424–429. IEEE, 2000.

[50] Hoanh Le. Geometric invariance of PointNet. *Science and Engineering*, 2021.

[51] Yongjin Lee, Senja D Barthel, Paweł Dłotko, S Mohamad Moosavi, Kathryn Hess, and Berend Smit. Quantifying similarity of pore-geometry in nanoporous materials. *Nature Communications*, 8(1):1–8, 2017.

[52] Javier Lamar Leon, Raúl Alonso, Edel Garcia Reyes, and Rocio Gonzalez Diaz. Topological features for monitoring human activities at distance. In *International Workshop on Activity Monitoring by Multiple Distributed Sensing*, pages 40–51. Springer, 2014.

[53] Mao Li, Hong An, Ruthie Angelovici, Clement Bagaza, Albert Batushansky, Lynn Clark, Viktoriya Coneva, Michael J Donoghue, Erika Edwards, Diego Fajardo, et al. Topological data analysis as a morphometric method: using persistent homology to demarcate a leaf morphospace. *Frontiers in plant science*, 9:553, 2018.

[54] Mao Li, Margaret H Frank, Viktoriya Coneva, Washington Mio, Daniel H Chitwood, and Christopher N Topp. The persistent homology mathematical framework provides enhanced genotype-to-phenotype associations for plant morphology. *Plant Physiology*, 177(4):1382–1395, 2018.

[55] Kart-Leong Lim and Hamed Kiani Galoogahi. Shape classification using local and global features. In *2010 Fourth Pacific-Rim Symposium on Image and Video Technology*, pages 115–120. IEEE, 2010.

[56] Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. *J. Mach. Learn. Res.*, 21(184):1–40, 2020.

[57] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3):419–441, 2008.

[58] Ippei Obayashi, Yasuaki Hiraoka, and Masao Kimura. Persistence diagrams with linear machine learning models. *Journal of Applied and Computational Topology*, 1(3-4):421–449, 2018.

[59] Nina Otter. Magnitude meets persistence. Homology theories for filtered simplicial sets. *arXiv preprint arXiv:1807.01540*, 2018. to appear in *Homology, Homotopy and Applications*.

[60] Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1):17, 2017.

[61] Rahul Paul and Stephan Chalup. Estimating Betti numbers using deep learning. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019.

[62] Jose A Perea, Anastasia Deckard, Steve B Haase, and John Harer. Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC bioinformatics*, 16(1):257, 2015.

[63] Giovanni Petri, Martina Scolamiero, Irene Donato, and Francesco Vaccarino. Topological strata of weighted complex networks. *PLOS ONE*, 8(6):e66506, 2013.

[64] James R Pomerantz. Wholes, holes, and basic features in vision. *Trends in Cognitive Sciences*, 7(11):471–473, 2003.

[65] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. `https://github.com/charlesq34/pointnet`. Accessed: 2022-02-01.

[66] 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*, pages 652–660, 2017.

[67] Archit Rathore, Sourabh Palande, Jeffrey S Anderson, Brandon A Zielinski, P Thomas Fletcher, and Bei Wang. Autism classification using topological features and deep learning: a cautionary tale. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 736–744. Springer, 2019.

[68] Vanessa Robins. Computational topology for point data: Betti numbers of α-shapes. In *Morphology of Condensed Matter*, pages 261–274. Springer, 2002.

[69] Christian Ronse. A bibliography on digital and computational convexity (1961-1988). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):181–190, 1989.

[70] Matteo Rucco, Filippo Castiglione, Emanuela Merelli, and Marco Pettini. Characterisation of the idiotypic immune network through persistent entropy. In *Proceedings of ECCS 2014*, pages 117–128. Springer, 2016.

[71] Gunnar Schmidtmann, Ben J Jennings, and Frederick AA Kingdom. Shape recognition: convexities, concavities and things in between. *Scientific reports*, 5(1):1–11, 2015.

[72] Benjamin Schweinhart. Fractal dimension and the persistent homology of random geometric complexes. *Advances in Mathematics*, 372:107291, 2020.

[73] Thomas Sikora. The mpeg-7 visual standard for content description-an overview. *IEEE Transactions on circuits and systems for video technology*, 11(6):696–702, 2001.

[74] Nikhil Singh, Heather D Couture, JS Marron, Charles Perou, and Marc Niethammer. Topological descriptors of histology images. In *International Workshop on Machine Learning in Medical Imaging*, pages 231–239. Springer, 2014.

[75] Primoz Skraba and Katharine Turner. Wasserstein stability for persistence diagrams. *arXiv preprint arXiv:2006.16824*, 2020.

[76] Bernadette J Stolz, Heather A Harrington, and Mason A Porter. Persistent homology of time-dependent functional networks constructed from coupled time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(4):047410, 2017.

[77] The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.3.0 edition, 2020.

[78] Christopher Tralie, Nathaniel Saul, and Rann Bar-On. Ripser.py: A lean persistent homology library for python. *The Journal of Open Source Software*, 3(29):925, Sep 2018.

[79] Renata Turkeš, Jannes Nys, Tim Verdonck, and Steven Latré. Noise robustness of persistent homology on greyscale images, across filtrations and signatures. *PLOS ONE*, 16(9):e0257215, 2021.

[80] Katharine Turner, Sayan Mukherjee, and Doug M Boyer. Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014.

[81] Katharine Turner and Gard Spreemann. Same but different: Distance correlations between topological summaries. In *Topological Data Analysis*, pages 459–490. Springer, 2020.

[82] Yuhei Umeda. Time series classification via topological data analysis. *Information and Media Technologies*, 12:228–239, 2017.

[83] Oliver Vipond. Multiparameter persistence landscapes. *J. Mach. Learn. Res.*, 21(61):1–38, 2020.

[84] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, 17(3):261–272, 2020.

[85] Yan Wang, Yining Zhao, Shihui Ying, Shaoyi Du, and Yue Gao. Rotation-invariant point cloud representation for 3-d model recognition. *IEEE Transactions on Cybernetics*, 2022.

[86] Eugene P Wigner. The unreasonable effectiveness of mathematics in the natural sciences. *Communications on Pure and Applied Mathematics*, 13:1–14, 1960.

[87] Chenxi Xiao and Juan Wachs. Triangle-net: Towards robustness in point cloud learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 826–835, 2021.

[88] Junming Zhang, Ming-Yuan Yu, Ram Vasudevan, and Matthew Johnson-Roberson. Learning rotation-invariant representations of point clouds using aligned edge convolutional neural networks. In *2020 International Conference on 3D Vision (3DV)*, pages 200–209. IEEE, 2020.

[89] Zhiyuan Zhang, Binh-Son Hua, David W Rosen, and Sai-Kit Yeung. Rotation invariant convolutions for 3D point clouds deep learning. In *2019 International Conference on 3D Vision (3DV)*, pages 204–213. IEEE, 2019.

[90] Ziwei Zhang, Xin Wang, Zeyang Zhang, Peng Cui, and Wenwu Zhu. Revisiting transformation invariant geometric deep learning: Are initial representations all you need? *arXiv preprint arXiv:2112.12345*, 2021.

[91] Chen Zhao, Jiaqi Yang, Xin Xiong, Angfan Zhu, Zhiguo Cao, and Xin Li. Rotation invariant point cloud analysis: Where local geometry meets global topology. *Pattern Recognition*, page 108626, 2022.

[92] Zhen Zhou, Yongzhen Huang, Liang Wang, and Tieniu Tan. Exploring generalized shape analysis by topological representations. *Pattern Recognition Letters*, 87:177–185, 2017.

[93] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] In both the introductory and concluding section, we clearly point to the relevant sections for each of the contributions.
   (b) Did you describe the limitations of your work? [Yes] The limitations are highlighted in the concluding Section 7, and in the end of curvature and convexity Sections 4 and 5 we provide more details about the interesting types of point clouds that could be considered.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] There is a paragraph in Section 7 addressing these impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 1 and Appendix A.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Due to the large number of computations, the error bars are only reported for some of the experiments (Figure 11 and Figure 12)
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] The PointNet article, and the model we use in our work, are both cited in Section 1. The image of the architecture is acknowledged in Figure 8.
   (b) Did you mention the license of the assets? [Yes] See the last part of the PointNet paragraph in Appendix A.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The data and code are referenced in Section 1 and Appendix A.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We do not use any personal data.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We construct datasets of point clouds that do not contain any personal information.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]